

The Best Thesis Anyone Has Ever Seen

by

GREGGORY H. ROTHMEIER

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Masters of Science
in the College of Arts and Sciences
Georgia State University
2012

Copyright by
Greggory Rothmeier
2012

The Best Thesis Anyone Has Ever Seen

by

GREGGORY H. ROTHMEIER

Committee Chair: A. G. Unil Perera

Committee: Mukesh Dhamala

Brian Thoms

D. Michael Crenshaw

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
May 2012

Dedication

Mama.

Acknowledgements

Perera, Dhamala, Brooke, Lab Mates, Dhamala's Lab

Contents

Acknowledgments	iv
List of Tables	vii
List of Figures	viii
1 Introduction	1
2 Calculating Temperature Changes using the fMRI BOLD Response	2
2.1 Background	3
2.1.1 Generation of the Blood Oxygen Level Dependent (BOLD) Response	3
2.1.2 Previously Proposed Temperature Models	3
Single-Voxel Approach	3
Multi-Voxel Approach	3
2.2 Modeling the BOLD Response	3
2.3 Modeling Temperature	3
2.3.1 The Approach	3
How the temperature is calculated	3
Calculating the equilibrium temperature	3
Calculating Metabolism and Blood Flow Changes	3
Calculating the change in temperature in the active brain	3
2.3.2 Results	3
Using Theoretical BOLD Data	3
Using Experimental BOLD Data	3
3 Detector Applications to measuring the active brain	4
3.1 Functional Near-Infrared fNIR Imaging	4
3.2 Temperature Measurements	4

4 Conclusion	6
References	7
Appendices	8

List of Tables

List of Figures

2.1	Generation of the fMRI BOLD Response	3
3.1	Absorption spectra of water, deoxyhemoglobin and oxyhemoblogin	5

AWESOME TITLE

A thesis

presented in Partial Fulfilment of Requirements for the Degree of Master of Science in the

College of Arts and Sciences

Georgia State University

2012

by

Greggory Rothmeier

Committee:

A. G. Unil Perera, Chair

Mukesh Dhamala, Member

Brian Thoms, Member

D. Michael Crenshaw, Member

April 1, 2012

Date

Dick Miller
Department Chair

Chapter 1

Introduction

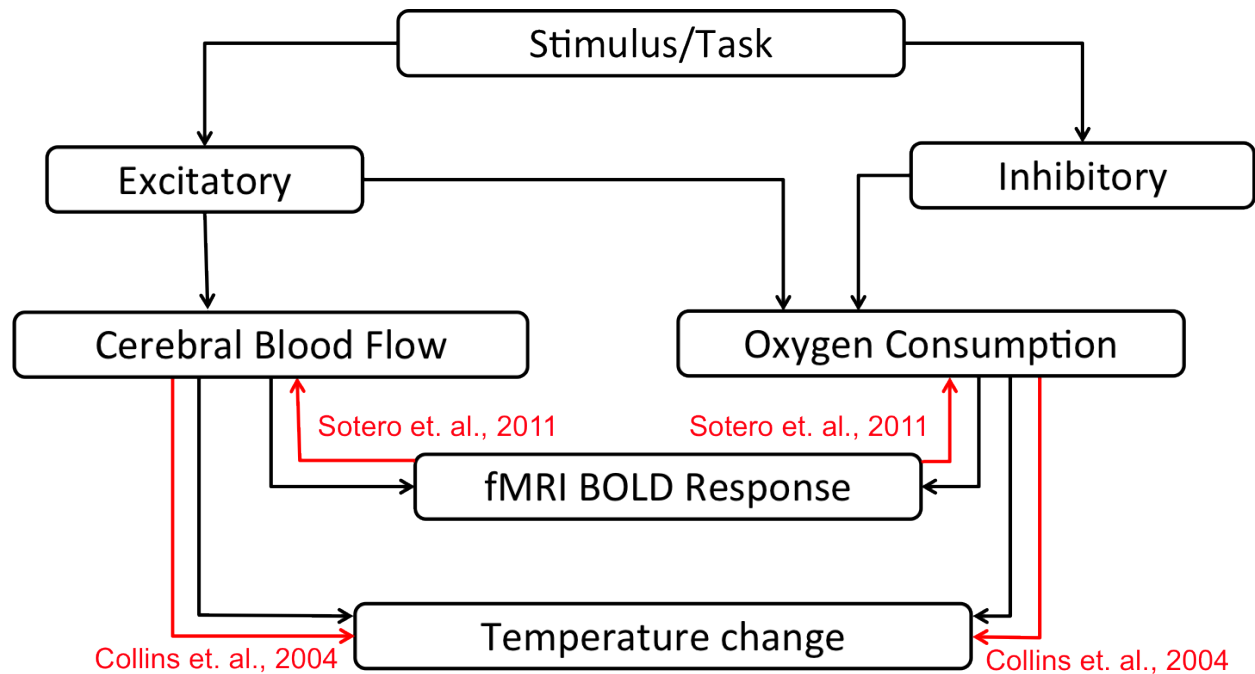


Figure 2.1: bla bla bla

Chapter 2

Calculating Temperature Changes using the fMRI BOLD Response

2.1 Background

2.1.1 Generation of the Blood Oxygen Level Dependent (BOLD) Response

2.1.2 Previously Proposed Temperature Models

Single-Voxel Approach

Multi-Voxel Approach

Chapter 3

Detector Applications to measuring the active brain

3.1 Functional Near-Infrared fNIR Imaging

$$I = I_0 e^{-\alpha x} \tag{3.1}$$

3.2 Temperature Measurements

From the Beer-Lambert law eq. (3.1), the penetration depth, δ_p can be expressed as

$$\delta = \frac{1}{\alpha} \tag{3.2}$$

where α is the absorption coefficient. At body temperature (37°) the peak wavelength in the blackbody spectrum is approximately BLA. For water at this wavelength, α is approximately HUGE, so δ is VERY SMALL.

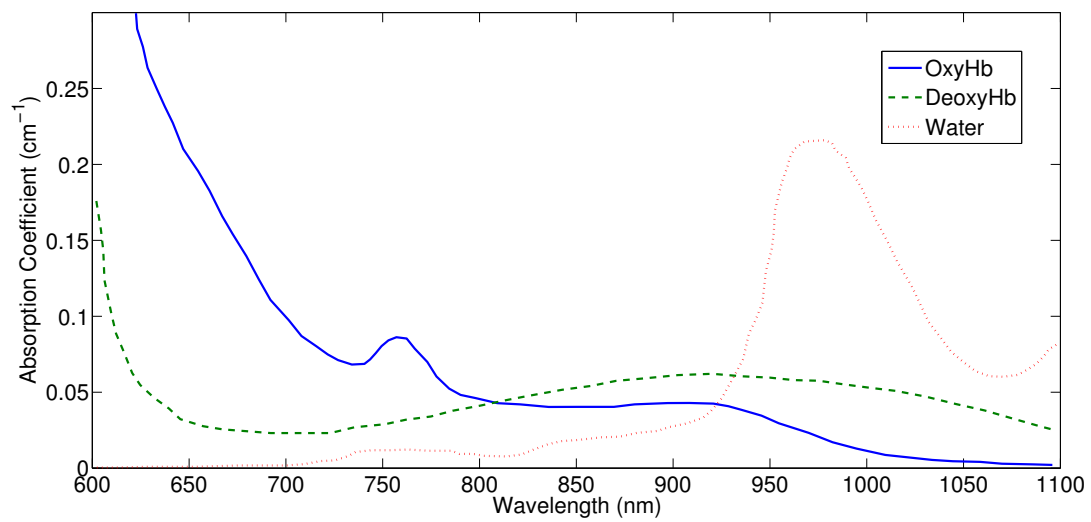


Figure 3.1: Absorption spectra of water, Hb and Dhb. From Cope [1] and HB stuff from Horecker [2]

Chapter 4

Conclusion

Bibliography

- [1] M Cope. The development of a near infrared spectroscopy system and its application for non invasive monitoring of cerebral blood and tissue oxygenation in the newborn infants. *London University*, 1991.
- [2] B L Horecker. The absorption spectra of hemoglobin and its derivatives in the visible and near infra-red regions. *The Journal of Biological Chemistry*, 1942.

Calculating the Temperature Change

```
1 function temperatureOut = tempCalcChangingMetabolismFlow(tissue,bloodT,
    airT,nt,tmax,pastCalc,start,stop,amplitudeMet,amplitudeFlow,region,
    savesteps)
2 % tempCalcChanging Metabolism How does changin metabolism affect things?
3 % tissue: holds all of the strucual information
4 % bloodT: Temperature of the blood
5 % airT: Temperature of the surrounding ait
6 % nt: Number of time steps
7 % tmax: Total amount of time the simulation should run over
8 %
9 % region: logical matrix same size as head
10 % start: units of steps
11 % stop: units of steps
12 %
13 % Writen by Gregory Rothmeier (greggroth@gmail.com)
14 % Georgia State University Dept. Physics and Astronomy
15 % May, 2011
16 %#codegen
17
18 %% Default Values
19 if nargin<2, bloodT = 37; end
20 if nargin<3, airT = 24; end
21 if nargin<4, nt = 3; end
22 if nargin<5, tmax = 1; end
23 if nargin<6, pastCalc = 0; end % Voxel size (m)
24 if nargin<7, start = 10; end % in steps
25 if nargin<8, stop = 20; end % in steps
26 if nargin<9, amplitudeMet = 1.2; end % normalized
27 if nargin<10, amplitudeFlow = 1.2; end % normalized
28 if nargin<11, savesteps = 1; end
29
```

```

30
31 dx = 2*10^-3;
32 if nt < (2*tmax),
33     warning('Time step size is not large enough. Results will be
        unreliable. Consider increasing the number of steps or reducing
        tmax.')
34 end
35
36
37 % Constants used that aren't already stored in tissue
38 [xmax ymax zmax t] = size(tissue);
39 clear t;
40 dt = tmax/(nt-1);
41 % rhoBlood = 1057;
42 % wBlood = 1000;
43 % cBlood = 3600;
44
45 statusbar = waitbar(0, 'Initializing');
46
47 %% Maps
48 % Creates a map that identifies where there is tissue
49 % the condition squeeze(tissue(:,:,:)~=airIndex picks out the elements
    that are
50 % tissue
51
52 temperatureOut = ones(ceil((nt-1)/savesteps),xmax,ymax,zmax,'single');
53 temperature = ones(2,xmax,ymax,zmax,'single')*airT;
54 if pastCalc == 0
55     temperature(1,squeeze(tissue(:,:,:,1))~=1) = bloodT;
56 else
57     temperature(1,:,:,:) = pastCalc(end,:,:,:);
58     % temperature(1,:,:,:) = pastCalc;

```

```

59 end
60 temperatureOut(1,:,:,:) = temperature(1,:,:,:);
61 metabMultiplier = ones([xmax ymax zmax],'single');
62 flowMultiplier = ones([xmax ymax zmax],'single');
63
64 %% Do Work.
65 % This is a vectorized version of the next section. For the love of
    god
66 % don't make any changes to this without first looking below to make
    sure
67 % you know what you're changing. This is [nearly] impossible to
68 % understand, so take your time and don't break it.
69 % data is stored in 'tissue' as such :
70 % [tissuetype 0 Qm c rho k w]; <-- second element is blank for all.
71 % [ 1 2 3 4 5 6 7
72
73 % This makes an array that has smoothed out variations in k by averaging
74 % all of the k's around each voxel (including itself). This is a
75 % hap-hazard solution to the problem that if you only take the value of
    k
76 % for the voxel without considering what surrounds it, it doesn't matter
77 % whether the head is surrounded by air, water or anything else. Since
78 % water is a better thermal conductor than air, we need a way of
79 % accounting for this. This is one way:
80 averagedk = (circshift(tissue(:,:,:,:),[1 0 0])+circshift(tissue(:,:,:,:),6)
    ,[-1 0 0])+circshift(tissue(:,:,:,:),[0 1 0])+circshift(tissue(:,:,:,:),6)
    ,[0 -1 0])+circshift(tissue(:,:,:,:),[0 0 1])+circshift(tissue(:,:,:,:),6)
    ,[0 0 -1])+tissue(:,:,:,:),6))/7;
81 rhoblood = 1057;
82 cblood = 3600;
83
84 %% Only saves every 4 steps

```

```

85 tic
86 for t2 = 1:nt-1
87     waitbar(t2/(nt-1),statusbar,sprintf('%d%',round(t2/(nt-1)*100)));
88     if (start<t2) && (t2<stop) % for __ steps
89         metabMultiplier(region) = amplitudeMet; % region is hardcoded
90         here
91         flowMultiplier(region) = amplitudeFlow;
92     elseif t2==stop % once the period is over, reset it back to ones
93         metabMultiplier(region) = 1;
94         flowMultiplier(region) = 1;
95     end
96
97 temperature(2,:,:,:) = squeeze(temperature(1,:,:,:)) + ...
98 dt/(tissue(:,:,:5).*tissue(:,:,:4)).* ...
99 ((averagedk/dx^2).*...
100 (circshift(squeeze(temperature(1,:,:,:)),[1 0 0])-2*squeeze(
101     temperature(1,:,:,:))+circshift(squeeze(temperature(1,:,:,:))
102     ,[-1 0 0])+... % shift along x
103     circshift(squeeze(temperature(1,:,:,:)),[0 1 0])-2*squeeze(
104     temperature(1,:,:,:))+circshift(squeeze(temperature(1,:,:,:))
105     ,[0 -1 0])+... % shift along y
106     circshift(squeeze(temperature(1,:,:,:)),[0 0 1])-2*squeeze(
107     temperature(1,:,:,:))+circshift(squeeze(temperature(1,:,:,:))
108     ,[0 0 -1]))... % shift along z
109     -(1/6000)*rhoblood*flowMultiplier.*tissue(:,:,:7)*cblood.*(
110     squeeze(temperature(1,:,:,:))-bloodT)+metabMultiplier.*
111     tissue(:,:,:3));
112
113 % resets the air temperature back since it's also modified above,
114 but
115 % it needs to be kept constant throughout the calculations
116 temperature(2,squeeze(tissue(:,:,:1))==1) = airT;
117 temperatureOut(ceil(t2/savesteps),,:,:,:) = temperature(2,:,:,:);

```

```

107     temperature(1,:,:,:) = temperature(2,:,:,:); % moves 2 back to 1
108 end
109 close(statusbar);
110 toc
111 %% Keeps all of the data
112 % Note: Make sure that temperature() has enough preallocated space
113 %{
114 tic
115 for t2 = 1:nt-1
116     %tic
117     waitbar(t2/(nt-1),statusbar,sprintf('%d%%',round(t2/(nt-1)*100)));
118     if (start<t2) && (t2<stop) % for 10 steps
119         metabMultiplier(region) = amplitudeMet; % region is hardcoded
120         here
121         flowMultiplier(region) = amplitudeFlow;
122     elseif t2==stop % once the period is over, reset it back to ones
123         metabMultiplier(region) = 1;
124         flowMultiplier(region) = 1;
125     end
126     temperature(t2+1,:,:,:) = squeeze(temperature(t2,:,:,:)) + ...
127         dt/(tissue(:,:,:5).*tissue(:,:,:4)).* ...
128         ((averagedk/dx^2).*...
129         (circshift(squeeze(temperature(t2,:,:,:)),[1 0 0])-2*squeeze(
130             temperature(t2,:,:,:))+circshift(squeeze(temperature(t2,:,:,:))
131             ,[-1 0 0])+... % shift along x
132             circshift(squeeze(temperature(t2,:,:,:)),[0 1 0])-2*squeeze(
133                 temperature(t2,:,:,:))+circshift(squeeze(temperature(t2,:,:,:))
134                 ,[0 -1 0])+... % shift along y
135                 circshift(squeeze(temperature(t2,:,:,:)),[0 0 1])-2*squeeze(
136                     temperature(t2,:,:,:))+circshift(squeeze(temperature(t2,:,:,:))
137                     ,[0 0 -1])))... % shift along z

```

```

131         -(1/6000)*rhoblood*flowMultiplier.*tissue(:,:,:,7)*cblood.*(
            squeeze(temperature(t2,:,:,:))-bloodT)+metabMultiplier.*
            tissue(:,:,:,3));
132     %   resets the air temperature back since it's also modified above,
        but
133     %   it needs to be kept constant throughout the calculations
134     temperature(t2+1,squeeze(tissue(:,:,:,1))==1) = airT;
135     %dispTimeLeft(1,1,nt-1,t2)
136 end
137 close(statusbar);
138 toc
139 %}
140
141 %% Old Code
142 %   This is what used to be used.   It's much slower (~60 times slower),
        but
143 %   it's much easier to understand compared to the above code.   If any
144 %   changes need to be made above, first look through this code to ensure
145 %   you understand what's happening before making changes.   It's really
        easy
146 %   to mess up the code above and nearly impossible to figure out where.
147
148 %   good luck.
149
150
151 %{
152 for t2 = 1:nt-1
153     for x2 = 2:xmax-1
154         for y2 = 2:ymax-1
155             for z2 = 2:zmax-1
156                 if tissue(x2,y2,z2,1) ~= 1,

```

```

157         temperature(t2+1,x2,y2,z2) = temperature(t2,x2,y2,z2)
           + (dt/(tissue(x2,y2,z2,5)*tissue(x2,y2,z2,4)))*((
           tissue(x2,y2,z2,6)/dx^2)*...
158         (temperature(t2,x2+1,y2,z2)-2*temperature(t2,x2,y2,
           z2)+temperature(t2,x2-1,y2,z2)+...
159         temperature(t2,x2,y2+1,z2)-2*temperature(t2,x2,y2,
           z2)+temperature(t2,x2,y2-1,z2)+...
160         temperature(t2,x2,y2,z2+1)-2*temperature(t2,x2,y2,
           z2)+temperature(t2,x2,y2,z2-1))...
161         -(1/6000)*rhoBlood*wBlood*cBlood*(temperature(t2,x2
           ,y2,z2)-bloodT)+tissue(x2,y2,z2,3));
162     end
163 end
164 end
165 end
166 end
167 %}
168
169
170 end

```

Calculating The Equilibrium Temperature