Fast and accurate voxel-by-voxel perfusion imaging using convolution models

A. Mikheev¹, and H. Rusinek¹

¹Radiology, NYU School of Medicine, New York, NY, United States

Modeling of the perfusion using dynamic contrast-enhanced (DCE) imaging based on convolution models is gaining increasing attention. The time activity curve of the voxel is sampled at predefined time intervals, yielding a contrast concentration $C(t)$ evaluated at discrete times $t=t_1, \ldots, t_{NT}$. Perfusion model is then specified to express $C(t)$ as the convolution of an impulse response function $IRF(t)$ and the arterial input function $A(t)$. For a given model, $IRF(t)$ is represented as a piecewise analytical function that is parametrized by tissue parameters such as flow rate, transit time, or volume distribution. These parameters $\{p_1, p_2, \ldots, p_n\}$ are then used to fit the measured data $C(t)$ by minimizing the residual:

$$\delta = \sum_{i=1}^{NT} (C(t_i) - A(t_i) \otimes IRF(t_i))^2$$

Typically the convolution operator is invoked on the order of $10^4$ times for each voxel’s minimization, and the imaging volume of interest contains between $10^3$ and $10^5$ voxels. Thus, an efficient implementation of the convolution represents a significant computational challenge. The naive approach consists of a uniform sampling $t=t_1, \ldots, t_{NT}$ of the temporal domain, extrapolation of $A(t)$ and $IRF(t)$ over these samples, and calculation of convolution either by discrete integration or by multiplying corresponding Fourier transforms. Computational complexity is $O(N^2)$ and $O(N \log(N))$ respectively. In practice direct integration is often used due to its simplicity. Additionally, due to discrete sampling of $IRF(t)$ result in an error that is proportional to $1/N$. To improve the computational speed and be able to control the convolution error due to discrete sampling of $IRF(t)$ we have developed and implemented the adaptive convolution algorithm. The method was tested for its speed and convolution accuracy on renal perfusion renography data using Gd-DTPA as the tracer.

METHODS

The method takes as an input parameter a user-specified tolerance $\tau$ measured in units of tracer concentration. The value

$$\epsilon = \tau \int A(t) \, dt$$

is then computed. For any given value of fitting parameters $\{p_1, p_2, \ldots, p_n\}$ we approximate the $IRF(t)$ with a (possibly discontinuous) piecewise linear (PWL) function $IRF^\#(t)$. To achieve this, we subdivide the domain $[0,T]$ into subintervals $[t_j, t_{j+1}]$ where: (a) $IRF$ is continuous, and (b) the modulus of its 2nd derivative is non-strictly monotonic. For each such subinterval we apply the following procedure to calculate $IRF^\#(t)$ with accuracy $|IRF(t)-IRF^\#(t)| < \epsilon$. Starting from the endpoint where the 2nd derivative module is larger, we compute the longest step $[u,v]$ towards the second endpoint, such that is guaranteed to satisfy $|IRF^\#(t)-IRF^\#(t)| < \epsilon$ for all $t$ in $[u,v]$. We are using the fact that module of the 2nd derivative can only decrease in the direction of advance according to the subinterval constraints. The process is iterated (i.e $u:=v$ or $v:=u$ depending on the starting endpoint) until the entire subinterval $[t_j, t_{j+1}]$ is covered. The second task, the convolution of two PWL functions, $A(t)$ and $IRF^\#$ is then calculated precisely in analytical form. The computational complexity of this step is $O(N_1+N_2)$ where $N_1$ is the number of samples for $A(t)$ and $N_2$ the number of samples for $IRF^\#$. The result is guaranteed to differ by no more than $\tau$ from the exact convolution. The adaptive convolution algorithm was implemented using C++ language.

RESULTS AND DISCUSSION

The algorithm was tested on the kidney perfusion model [1] and MR renography acquisition that consisted of 17 dynamic frames acquired in a nonuniform fashion over 75 seconds. Each 3D volume consisted of 300K voxels. Calculation of renal plasma flow (Fig. 1) on the dual-core Intel T7200 2.3GHz mobile processor took about 3 minutes for $\tau = 0.001$ mM. Figure 2 plots the execution time as a function of $\tau$. For comparison, the naive approach, when implemented using optimized C++ code, required sampling $dt = 0.1$ sec to attain a similar precision and took 400 times longer to execute. In conclusion, the adaptive convolution algorithm enables fast and accurate analysis of DCE MR datasets.

REFERENCES