GPU-accelerated SAR computation with arbitrary averaging shapes

Andre Kuehne[1], Frank Seifert[1], and Bernd Ittermann[1]
[1]*Physikalisch-Technische Bundesanstalt, Braunschweig & Berlin, Germany*

Introduction: Specific absorption rate (SAR) evaluation of an MR transmit coil is mandatory to assess patient safety. Commercial EM-simulation software packages provide SAR computation routines; however these often lack the speed and flexibility required for more complex usage scenarios like multi-channel excitation. Additionally, the computation standards usually employed (IEEE 1528.1, C95.3) exhibit averaging artifacts at different tissue- and tissue-air interfaces [1, 3]. Both speed and artifact issues have been previously approached by regridding to high resolutions [2] and region growing algorithms [3]. As an alternative, we present a fast, inherently parallelized algorithm to calculate the SAR using modern multi-core CPUs and GPUs that can be easily implemented without advanced programming skills.

The SAR averaging process can be regarded as a repeated convolution of the three-dimensional mass and power density distributions using kernels of varying sizes representing the differently sized averaging volumes. Calculating the convolution in Fourier space is much faster than in real space. The FFT can be very efficiently calculated using modern graphics processing units (GPUs); and freely available implementations for MATLAB (e.g. GPUMat, gp-you.org) or Python (PyFFT, pypi.python.org/pypi/pyfft) allow code execution without knowledge of GPU-specific programming.

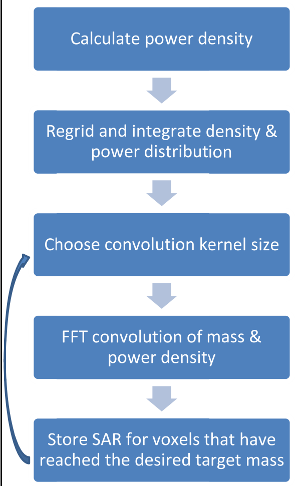

Fig. 1: Averaging algorithm

Methods: A flowchart of the algorithm is depicted in Fig. 1. For the purpose of this work, it was implemented using MATLAB and GPUMat. After calculating the local power density [3] the data needs to be regridded to a uniform mesh. Previous approaches required high resolutions of less than 1 mm to achieve the required precision [2]. This is circumvented by using sub-voxel growing of the convolution kernel, which is illustrated in Fig. 2. The shape of the averaging volume, e.g. cubical or spherical, is simply chosen by generating the appropriately shaped kernel. Minimum and maximum averaging kernel sizes are chosen based on the desired averaging mass, known resolution, minimum density and a volume fill factor, i.e. how much free space may be contained inside a volume. The mass and power arrays need to be zero padded with half the maximum kernel size to prevent aliasing artifacts. Stepping through the different kernels is not done linearly, but by sampling the region between minimum (0) and maximum (1) volume in increasingly smaller steps: [0, 1, ½, ¼, ¾,...]. No loop structures are required besides the kernel iteration loop, resulting in clear, fully vectorized and thus fast code. To test the algorithm, SAR distributions for a 3T, 16-rung birdcage head resonator loaded with the "Duke" model [5] (simulation resolution 1 mm) were calculated in XFdtd 7.2 (Remcom, State College, PA, USA) for both cubical and spherical averaging volumes and various regridding resolutions.

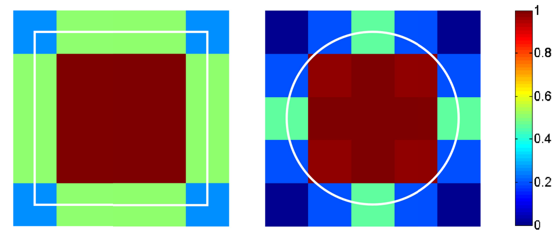

Fig. 2: Illustration of the sub-voxel volume growing scheme for a target volume diameter of 4 (only center slice is shown). The white outline shows the desired target volume size (left: cubical, right: spherical). Every kernel tap is filled with the value corresponding to the fraction of its volume being contained inside the target outline (color coded).

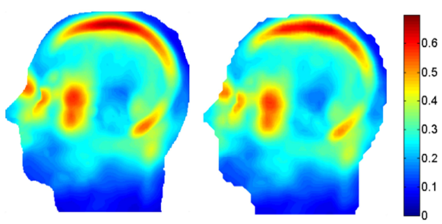

Fig. 3: Sagittal maximum intensity projections of the 10g averaged SAR / absorbed power (spherical volume) for 1mm (left) and 5 mm (right) resolution

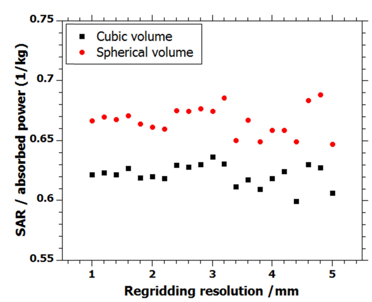

Fig. 4: Dependence of the 10g peak SAR / absorbed power (k-factor) on the chosen regridding resolution.

Results: The calculated 10g SAR distributions (Fig.3) show a very small variation of less than 5% for the peak value w.r.t. the XFdtd result when changing the gridding resolution (Fig. 4). Calculation takes less than 2 seconds at 5 mm resolution on an NVIDIA Tesla C2070. This shows that the sub-voxel growing scheme works as expected. SAR values for spherical averaging shapes were consistently higher by 7% on average. While the algorithm does not enforce connectedness of the voxels inside the averaging volume, it still effectively grows the averaging volume into the body at the outer surface and thus renders superficial hotspots correctly.

Conclusion: We have introduced an algorithm to very quickly calculate the SAR for arbitrary averaging mass and shape using modern GPUs. It is very robust even at low resolutions and can be straightforwardly extended to related problems such as the calculation of Q-matrices [6] for local SAR management in transmit arrays.

[1] Kuehne et al, ESMRMB, 2011, 91
[2] Kozlov et.al. (2009) Proc.ISMRM 17 4779
[3] Oh et.al. (2011) Proc.ISMRM 19 3868
[4] de Moerloose, J., & De Zutter, D. (1995). Microwave and Optical Technology Letters, 8(5), 257-260.
[5] Christ et.al. (2010) Phys.Med.Biol. 55(2); DOI: 10.1088/0031-9155/55/2/N01
[6] Homann et al. (2011). Magma 09/2011; DOI: 10.1007/s10334-011-0281-8