A Splatting Method to Generate DRRs for Deformed CT Volume

Purpose: To develop an efficient algorithm for generating high-quality digitally reconstructed radiographs (DRRs) for regularly and irregularly sampled volumes based on a splatting method with dynamic elliptical Gaussian kernels, and to evaluate this method against ray tracing.

Method: Methodologies for simulating x-ray images or computing digitally reconstructed radiographs (DRRs) are an essential component of 3D/2D image registration. DRR generation is to simulate an x-ray image derived from a computed tomography (CT) volume dataset by computing the attenuation coefficient of virtual x-rays. This perspective volume rendering can be divided into two categories: 1) forward projection methods, such as the splatting algorithm, mapping the volume elements (voxels) onto the image plane, which is voxel-driven approach, and 2) back-projection methods, such as ray tracing algorithm. For ray tracing algorithms, rays pass through the CT volume to form an image using the trilinear interpolation among voxels, and the contribution of every intersected voxel is taken into account to the DRR.

In splatting, a voxel’s contribution is mapped directly onto the pixels without any interpolation. Figure 1 illustrates the projection of each voxel onto the image plane, in which the voxel’s energy is spread over image plane by a reconstruction kernel centered on the voxel. This reconstruction kernel is called a “splat” or “footprint”. Theoretically, the splat is regarded as a spherically symmetric 3D reconstruction kernel, such as a 3D Gaussian kernel. In our application, the splat is reconstructed into a 2D image plane so it can be implemented by a 2D reconstruction kernel, called a “footprint function”, which contains the integration of the 3D kernel along one dimension. Equation (1) shows how the projection image is calculated, where f(x, y) is the final pixel intensity of the projection image, Voxel(i, j, k) is the intensity of voxel at (i, j, k) position; the kernel(x, y) is the “footprint function” at kernel center (s, t), x and y are the Gaussian kernel area within a radius of 3σ on the projection image.

For perspective projection with anti-aliasing, the Gaussian kernel radius σ should have dynamic sizes, which can be calculated from similar triangles shown in Figure 2 and Equation (2), where d1 is the distance from the source to the voxel, d2 is the distance from the source to kernel center on the image plane going through a specific voxel point, r is the size of splat, and VoxelScale is the scale of the voxel size. The optimum coefficient value between σ and r is 0.57 based on our series of experimental results. If the volume is a regular grid, VoxelScale is fixed for all the voxels. If the volume is irregularly-sampled, VoxelScale varies individually, depending on the neighbors’ distribution of one voxel. We can use the original Gaussian kernel in regular grid to compute the deformed elliptical Gaussian kernel in irregular grid under the M affine transformation as shown in Figure 3. The Gaussian kernel is a circle with equal size at all directions; otherwise, it is an ellipse.

Results: We implement our programs using Microsoft Visual C++ 2010 and Matlab R2010a on an Intel Xeon X5160 3.00GHz with 8GB memory. The regularly-sampled volume is obtained from the 4D extended cardiac-torso (XCAT) digital phantom at respiratory Phase one (end...
inhale). It has a dimension of 256 x 256 x 130 with regular voxel size of 0.3125 mm and the projection image has a dimension of 256 x 256 with a pixel size of 0.4 mm. Figure 4(a) shows ray tracing with trilinear interpolation results; the calculation time is 9.1829 seconds. Figure 4(b) shows the splatting results, obtained in 3.0970 seconds. Figure 4(c) illustrates intensity difference between ray tracing and splatting; the normalized cross correlation (NCC) between two projections is 0.9980.

The irregularly-sampled volume is obtained at respiratory phase four (end exhale) and is presented as a dataset of Phase one plus the deformation vector of every voxel. For the splatting method, the Gaussian kernel of every voxel is modified according to the deformation vectors of its neighbors. The DRR is illustrated in Figure 4(e). In order to generate a DRR using the ray tracing method, the irregularly-sampled dataset must be re-sampled to a regularly-sampled dataset; the result is shown in Figure 4(d). The NCC between the splatting results without re-sampling (Figure 4(e)), and ray tracing result with re-sampling (Figure 4(d)) is 0.9977. Figure 4(f) shows the intensity difference between these two methods. The differences in Figure 4(f) are greater than those in Figure 4(c).

One main advantage of splatting over ray tracing is that splatting has a faster calculation speed. In splatting, it is very easy to ignore empty voxels which do not contribute to the final image; however, this is difficult to realize with ray tracing method. With similar results, the speed of splatting is 3 times faster than ray tracing. Another significant advantage is the ability to handle deformed volumes in an irregular grid. It is convenient to calculate the projection image depending on each voxel’s current position and the deformation vectors. In contrast, the ray tracing method requires the computation of rays going through all voxels following deformation, which is very complicated, or re-sampling of the deformed volume to a regularly sampled grid prior to applying the ray tracing. This re-sampling process results in additional computation cost and leads to inaccuracy.

Conclusions: Our splatting approach can generate high-quality DRRs efficiently and is a good alternative for current DRR generation techniques for deformed volume images.

References: