

Point-based digitally reconstructed radiograph

Aodong Shen¹, Limin Luo^{1,2}

1. Laboratory of Image Science and Technology, Southeast University, Nanjing, China

2. Center de Recherche en Information Biomédicale Sino-français (CRIBs), France

shenaodong@gmail.com, luo.list@seu.edu.cn

Abstract

In this paper, we present a novel point-based digitally reconstructed radiography (PBDRR) method for large CT data sets. Three steps are mainly included in the proposed method, namely point sampling, point rendering and quantization. Finite samples are determined on the nature or user-specified transfer function by a Monte Carlo technique. These samples are projected onto an image plane and the intensity value of each pixel records the count of samples projected onto the corresponding pixel area. The transfer function is related with the X-ray photon energy and the CT value. From the view of Monte Carlo integration, sufficient samples make the estimated pixel intensities believable theoretically. Moreover, PBDRR has $O(N^2)$ time complexity depending on the number of samples and is suitable for parallel processing and hardware acceleration. Some experimental results demonstrate the performance of the present method without redundant adjustments of the transfer function.

1 Introduction

Digitally reconstructed radiograph (DRR) plays a more and more important role for clinical applications, such as 2D-3D image registration [8, 12], and portal image for radiotherapy [8]. Several simulation toolkits have been developed for radiotherapy treatment planning and verification [8]. DRR simulates X-ray imaging on the virtual human phantom reconstructed from medical tomography data sets, *e.g.* CT, MRI. Virtual X-ray imaging toolkits divide into two categories, namely deterministic and stochastic methods in general. Deterministic methods generate the X-ray like-image directly based on the law of attenuation absorption. It is dominant for stochastic methods that the Monte Carlo technique is applied to simulate the interactions between photons and materials based on physical process.

However, most algorithms generally have $O(N^3)$ time complexity for clinical application, such as ray casting [7], splatting [1], shear-warp factorization [2]. Although some rapid algorithms improve the speed of ray casting [16], its computation complexity is still in same order as before. Fourier volume rendering (FVR) method enables real time preview of DRR reducing the relative computational complexity to $O(N^2 \log N)$ but not supports perspective projection [5]. Attenuation fields DRR (AF-DRR) generates the projections rapidly but required large memory for pre-computation results [12]. Stochastic methods simulate photon transport during traversal of all CT data voxels by various Monte Carlo methods. However, to refine results with scatter effect, all of those techniques need huge number of photons up to millions or even billions and are too time-consuming [11].

Recently, volume rendering technique pays attention to use points as rendering primitives for the sake of its efficiency [13, 14, 15]. Monte Carlo volume rendering (MCVR), proposed similarly by Csebfalvi [4, 3], is a very suitable technique for generating DRRs since it is dependent on the number of the samples of point cloud rather than the size of medical data sets and supports perspective projection also. The proposed point-based DRR (PBDRR) technique considers the medical data set as a set of points. The point density is derived from a user-specified or nature transfer function that converts the point's intensity (Hounsfield unit values in a CT data set) into the attenuation coefficients. Contrast to the MCVR technique, PBDRR considers the number of the points except for the intensities of the points onto projection area. To reduce the variance, Russian roulette sample technique is moreover utilized. PBDRR obtains the good-like X-ray image further without adjusting the transfer function repetitiously. Compared with the MCVR, the present PBDRR achieves better DRR quality with less number of sample with the nature transfer function.

2 Materials and methods

2.1 Point-based DRR

Point-based DRR is very similar to point-based volume rendering consisted by three aspects: point sampling, point rendering and quantization according to Monte Carlo integral [4, 14]. In detail, the ideal of Monte Carlo integral is to evaluate numerically the integral $I = \int_{\Omega} f(x)dx$ using random samples, where I represents the pixel intensity of image plane and $f(x)$ describes the attenuation quantity for each path step dx on the volume domain Ω . The solution, in its basic form, resorts independently sampling N points X_1, \dots, X_N according to given probability density function (PDF) $p(x)$, and then computing the expected value

$$\hat{I} = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)} \quad (1)$$

This estimator is known as the Horvitz-Thompson estimator [9]. It is obviously to show the estimator \hat{I} gives the correct result on average. Specifically, we have

$$\begin{aligned} E[\hat{I}] &= E\left[\frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{p(X_i)}\right] \\ &= \frac{1}{N} \sum \int_{\Omega} \frac{f(x)}{p(x)} \cdot p(x) dx \\ &= \int_{\Omega} f(x) dx \\ &= I \end{aligned} \quad (2)$$

provided that $f(x)/p(x)$ is finite whenever $f(x) \neq 0$. Therefore, the corresponding variance has the following form[4]:

$$var(\hat{I}) = E[\hat{I}^2] - E[\hat{I}]^2 = \frac{p(x)(1-p(x))}{N} \quad (3)$$

Monte Carlo integral is very suitable for graphics processing because it converges at a rate of $O(N^{-1/2})$ in any dimension, regardless of the smoothness of the integrand[6]. In addition, importance sampling can be applied to handle integrands with singularities effectively such as Metropolis method [10].

2.2 Point sampling

2.2.1. Hybrid sampling method Given a continuous PDF, the rejection method is often used to generate the point cloud due to von Neumann [6]. The idea is to

sample from some convenient density $g(x)$ such that $p(x) \leq c \cdot g(x)$ for some constant c .

In our method, both $p(x)$ and $g(x)$ have the equivalent domain, and for simplification, the assumption is considered as $g(x) \equiv 1$ and $c \equiv 1$. Russian roulette is also used to avoid randomly losing of most of the evaluations associated with small contributions as follows.

Algorithm 1 Hybrid sampling method

```

1:  $i \leftarrow 0$ 
2: while  $i < N$  do
3:    $l \leftarrow random(1, MaxLevel)$ 
4:   if  $random() \leq p(l)$  then
5:      $size \leftarrow hist[l].size()$ 
6:     if  $size < random() \cdot MeanSize$  then
7:       if  $random() > q_i$  then
8:         continue
9:       end if
10:    end if
11:     $index \leftarrow random(0, size)$ 
12:     $pointIndex \leftarrow hist[l][index]$ 
13:     $point[i][l] \leftarrow filter(pointIndex)$ 
14:     $i \leftarrow i + 1$ 
15:  end if
16: end while

```

First of all, the histogram of the data sets, namely $hist[l][i]$, is generated and corresponding voxel indices have been stored to the origin, which is the left-bottom corner of image in general. The first dimension l of the histogram denotes the density level within the range [0, 4096]. The second dimension i represents the voxel index associated with the same value. The number N determines the number of sample points to acquire better image quality. The procedure $random(a, b)$ returns one random variant between the input a and b , in which the default is 0 and 1 respectively. The variant $MeanSize$ describes the mean size of all valid levels. In addition, one tiny Russian roulette threshold q_i is used to balance the high density with small size. Therefore, the new estimator \hat{I}'_i replaces the original form as

$$\hat{I}'_i = \begin{cases} \frac{1}{q_i} \hat{I}_i & \text{with probability } q_i \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Unlike adaptive sampling, Russian roulette sampling do not introduce any bias[6].

$$\begin{aligned} E[\hat{I}'_i] &= q_i \cdot \frac{1}{q_i} E[\hat{I}_i] + (1 - q_i) \cdot 0 \\ &= E[\hat{I}_i] \end{aligned} \quad (5)$$

To smooth the holes on the final image, the space filter is utilized including various types, tent, ellipsoid, sphere,

and so on. The time complexity of this hybrid sampling depends on N , the number of sample points instead of the number of the data sets. In general, N is be proportional to the product of the dimension of the final image, or equal specially. Both the time and storage complexity of the present algorithm is $O(N^2)$ in common against the MCVR.

2.2.2. Metropolis method Metropolis technique is very useful for sampling arbitrary densities on high-dimensional spaces originally used to predict the material properties of liquids, and has the advantage that the density function does not need to be normalized. Thus, it is most useful when to generate a long sequence of samples from the given density $p(x)$ [6, 10].

Considering volume domain Ω and a non-negative function $f : \Omega \rightarrow \mathbb{R}^+$, Metropolis method generates a random sequence $X_0, X_1, \dots, X_i \in \Omega$, i.e., Markov chain. Each point X_i is determined by making a random walk to X_{i-1} according to an acceptance probability $a(X \rightarrow Y)$ and the initial point X_0 is selected randomly. The detailed sampling procedure is consisted by two steps as follows in general. Firstly, a candidate point X'_i is generated with one random walk and corresponding probability density $p(X'_i)$ is calculated. Secondly, the candidate point is either accepted or rejected, according to the acceptance probability $a(X_{i-1} \rightarrow X'_i)$ as follows:

$$X_i = \begin{cases} X'_i & \text{if } r \leq a(X_{i-1} \rightarrow X'_i) \\ X_{i-1} & \text{otherwise} \end{cases} \quad (6)$$

where the uniform random number $r \in [0, 1]$, and the acceptance probability $a(X_{i-1} \rightarrow X'_i) \equiv p(X'_i)/p(X_{i-1})$.

To improve the sampling efficiency in Metropolis method, we take the density level as the sampling domain rather than the space domain. As the hybrid sampling method, the Russian roulette technique is also utilized to decrease the image variance.

2.3 Point rendering and Quantization

Based on the aforementioned algorithms, the point cloud can be obtained according to specified transfer function. The pixel of image-plane is generated by projecting the point cloud and assigned the count of fallen into the corresponding area simply. The spatial impression can be improved by depth cueing and shading in contrast of the classical X-ray like volume rendering [10]. The other technique can be utilized to incorporate the effects of point collision with the Z-buffer algorithm during the projection stage [14].

A simulated X-ray image of the volume is rendered by quantizing the estimated normalized intensities onto L gray levels provided by the available display device [10]. For simplification and more wider dynamic range, we show the projection image directly with adjusting the corresponding window level and window width.

3 Results and discussion

Our algorithms were implemented in C++. The hardware configuration was single 3.0 GHz Pentium 4 PC with 512 MB of RAM, and it was executed purely on the single thread software without exploiting any supporting 3D hardware. We utilized various areas CT volume data sets, which came from the Visible Human Project (VHP) with 1 mm resolution (<https://mri.radiology.uiowa.edu/VHDicom/index.html>). One of data sets is a $512 \times 512 \times 300$ CT scan of female head; and the other is a $512 \times 512 \times 400$ CT scan of hip.

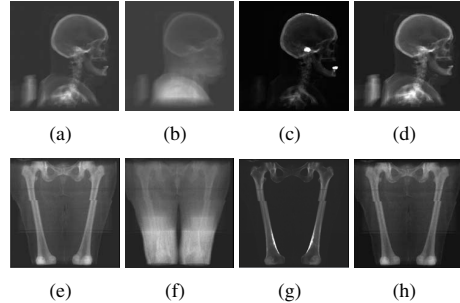


Figure 1. DRR images generated from CT data sets head ((a)-(d)) and hip ((e)-(h)) using the PBDRR ((a) and (e)), the MCVR ((b) and (f)) PBDRR without Russian roulette ((c) and (g)) with 16M samples, and the PBDRR with 8M samples (d) and (h)

Compared to MCVR and AMCVR, PBDRR introduces the Russian roulette technique to decrease the ratio of singularity samples as where the density is very large with less quantity. Figure 1 shows the DRR images generated using the PBDRR, the MCVR and PBDRR without Russian roulette sampling. For the same transfer function and number of sample points, PBDRR gives better X-ray like image than the MCVR for the sake of Metropolis sampling method. To obtain the similar results, the MCVR must adjust the transfer function to reduce the fraction of large numbers of low density voxels by try-error methods but time-consuming. Under the same conditions, the PBDRR produces more ac-

curate and refine images than PBDRR related only the density.

Figure 2 gives the root-mean-square (RMS) error to describe the difference between PBDRR and MCVR. The reference images for the RMS error calculation were generated by analytically integrating the intensity function assuming trilinear reconstruction [3]. From the view of RMS, the PBDRR evaluates nearly better performance than MCVR by one times.

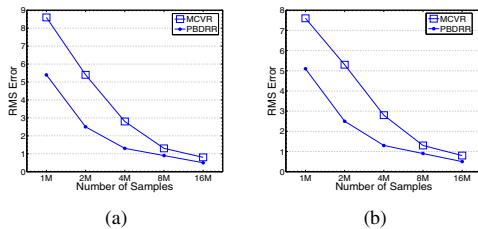


Figure 2. RMS error of DRR images generated by MCVR and PBDRR for CT head (a) and CT hip (b)

4 Conclusion and future work

In this paper, we have proposed a novel point-based DRR based on various sampling technique. The time and storage complexity is the same, namely $O(N^2)$ depending on the number of samples instead of the number of voxels. The present method is mainly adapt for large data sets, which can reduce the computation time and obtain the X-ray simulation image. It is very useful for clinical application such as 2D-3D registration and radiotherapy treatment planning. Based on the nature transfer function, the PBDRR produces the better DRR quality than the MCVR without redundant transfer function control. In the future, the PBDRR can be improved by multi-thread processing and low-discrepancy sequence to reduce the variance further.

Acknowledgment

This work was supported by National Basic Research Program of China under Grant no. 2003CB716102.

References

[1] W. Birkfellner, R. Seemann, M. Figl, J. Hummel, C. Ede, P. Homolka, X. Yang, P. Niederer, and

H. Bergmann. Wobbled splatting—a fast perspective volume rendering method for simulation of x-ray images from ct. *Physics in Medicine and Biology*, 50(9):N73–N84, 2005.

[2] W. Cai and G. Sakas. Drr volume rendering using splatting in shear-warp context. *Nuclear Science Symposium Conference Record, 2000 IEEE*, 3:19/12–19/17 vol.3, 2000.

[3] B. Csebfalvi. Interactive transfer function control for monte carlo volume rendering. *Volume Visualization and Graphics, 2004 IEEE Symposium on*, pages 33–38, 11–12 Oct. 2004.

[4] B. Csebfalvi and S.-K. Szirmay-Kalos. Monte carlo volume rendering. *vis*, 00:59, 2003.

[5] N. E, M. TA, and N. KS. Fourier volume rendering for real time preview of digital reconstructed radiographs: a web-based implementation. *Comput Med Imaging Graph*, 26(1):1–8, 2002.

[6] V. Eric. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, Dec. 1997.

[7] C. Fox, H. E. Romeijn, and J. F. Dempsey. Fast voxel and polygon ray-tracing algorithms in intensity modulated radiation therapy treatment planning. *Medical Physics*, 33(5):1364–1371, 2006.

[8] N. Freud, J.-M. Letang, and D. Babot. A hybrid approach to simulate x-ray imaging techniques, combining monte carlo and deterministic algorithms. *Nuclear Science Symposium Conference Record, 2004 IEEE*, 5:3075–3079, 16–22 Oct. 2004.

[9] D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, Dec. 1952.

[10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[11] D. W. O. Rogers. Fifty years of monte carlo simulations for medical physics. *Physics in Medicine and Biology*, 51(13):R287–R301, 2006.

[12] T. Rohlfing, D. B. Russakoff, J. Denzler, K. Mori, and J. Calvin R. Maurer. Progressive attenuation fields: Fast 2d-3d image registration without precomputation. *Medical Physics*, 32(9):2870–2880, 2005.

[13] M. Sainz and R. Pajarola. Point-based rendering techniques. *Computers & Graphics*, 28(6):869–879, Dec. 2004.

[14] N. Sakamoto, J. Nonaka, K. Koyamada, and S. Tanaka. Particle-based volume rendering. *Visualization, 2007. APVIS '07. 2007 6th International Asia-Pacific Symposium on*, pages 129–132, 5–7 Feb. 2007.

[15] I. Wald and H.-P. Seidel. Interactive ray tracing of point-based models. *Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, pages 9–16, 20–21 June 2005.

[16] H. Zhao and A. Reader. Fast ray-tracing technique to calculate line integral paths in voxel arrays. *Nuclear Science Symposium Conference Record, 2003 IEEE*, 4:2808–2812, 19–25 Oct. 2003.