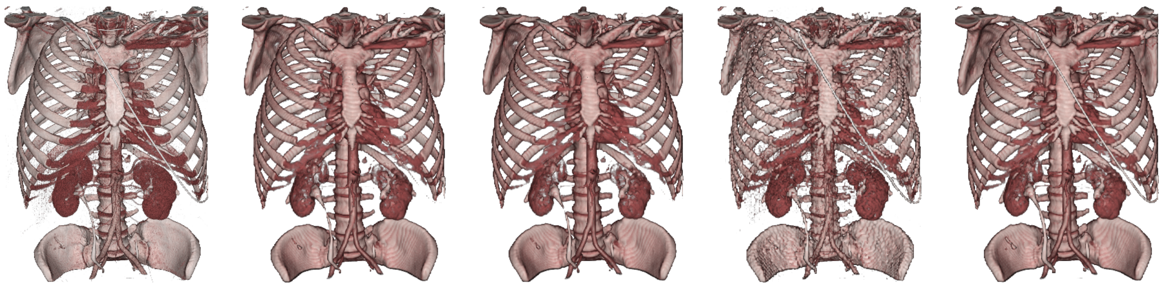
Feature-Preserving Downsampling for Medical Images

J. Díaz-García[†1] , P. Brunet[1] , I. Navazo[1] , F. Pérez[2] and P. Vázquez[1]

[1]Universitat Politècnica de Catalunya, [2]Alma IT Systems



(a) Full resolution (b) Average (c) Gaussian (d) Topology-Guided (e) Our proposal

Figure 1: *Comparison of downsampling filters for volumetric scalar fields. (a) shows the full resolution (512^3) CT dataset. Models on the right (128^3) are produced using different filters. While (b) and (c) lose some structures (ureter, ribs, catheter...), (d) produces excessively noisy results. Our proposal (e) preserves most of these features and produces quality results.*

Abstract

In the medical imaging field, interactive direct volume rendering of large volume datasets is a challenging task. Multi-resolution techniques deal with this problem by downsampling the original dataset to produce coarser representations. We present an evaluation of different downsampling filters with respect to their effectiveness at preserving details of the original dataset. Moreover, we propose a new Gaussian-based filter that produces quality lower-resolution representations and preserves small features that are prone to disappear.

Categories and Subject Descriptors (according to ACM CCS): I.4.10 [IMAGE PROCESSING AND COMPUTER VISION]: Image Representation—Volumetric

1. Introduction

With the improvement in capture devices, medical image datasets have grown continuously. The amount of memory of modern GPUs is also growing, but unfortunately the increasing rate of the size of volumetric datasets is even much higher. Different rendering techniques based on bricking, LOD, multiresolution or data compression have been proposed to obtain interactive visualizations [BRGIG*14, BHP14, HKRs*06].

† Projects TIN2013-47137-C2-1-P and TIN2014-52211-C2-1-R of the Spanish Ministerio de Economía y Competitividad and Alma IT Systems, for supporting J. Díaz-García's Ph.D. Thesis

In this paper we focus our attention on building LOD resolution models of scalar data fields [LHJ99, GWGS02]. Although more consistent coarser representations of a volume dataset are generally produced by means of downsampling color data [KB08], it is more convenient to store medical data as scalar fields, where density values can be mapped to opacity-weighted colors during post-classification by means of a transfer function that can be modified interactively. To deal with the memory size problem, commercial solutions usually reduce the dataset by a factor of two in each dimension until it fits the available memory. Alternatively, a well-known approach consists in pre-filtering and subsampling the original volume at discrete resolution levels. However, the visualization of coarser representations leads to inconsistencies and the loss of information in relation to the original data. Some authors strive to prevent this issue by storing more information in coarser models or performing costly pre-processes [YMC06, SHKM14]. Nevertheless, efficient approaches are desirable for the software/hardware used in the clinical practice. Following this line, downsampling by means of averaging voxels, or possibly filtering and subsampling the original dataset, are the most common approaches [BNS01, LW90]. Also, a method to downsample scalar data preserving topology is described in [KE01].

We have studied several downsampling techniques to analyze both their effect in preserving the main structures of the original datasets and the quality of the obtained models. Based on these results, we propose a new fully automatic feature preserving filter that locally adapts to structures that are prone to disappear during simplification.

2. Feature-Preserving Downsampling

Given a volumetric scalar field V_n of resolution 2^n, to compute a coarser representation V_{n-k} ($k > 0$) our downsampling technique proceeds in three steps:

- First, a temporary coarser volume S_{n-k} is computed by means of Gaussian-filtering and subsampling.
- Next, the distance between V_n and S_{n-k} provides hints about the loss of features in the first step. Using this information we generate a filtered volume F_n using *Local Feature Kernels*, which better preserve original volume details that would otherwise disappear with standard Gaussian filtering and subsampling.
- Finally, F_n is subsampled to obtain V_{n-k}.

To compute the filtered volume dataset F_n, we perform the following convolution:

$$F_n(x) = \sum_{i \in B_r} V_n(x+i) \cdot f_x(i)$$

where f_x is a normalized *Local Feature Kernel* with support B_r, a ball of radius r centered at the origin. f_x is in turn a product of a normalized global Gaussian kernel g and a distance kernel d_x:

$$f_x(i) = \frac{1}{\alpha} \cdot g(i) \cdot d_x(i), \quad \forall i \in B_r$$

The denominator $\alpha = \sum_{j \in B_r} g(j) \cdot d_x(j)$ ensures the sum of weights in f_x equals one. The distance kernel d_x is defined as the normalized absolute distance of values in the neighborhood of x between the original scalar field V_n and the temporary Gaussian-downsampled scalar field S_{n-k}:

$$d_x(i) = \frac{1}{\beta} |V_n(x+i) - S_{n-k}(x+i)|, \quad \forall i \in B_r$$

Again, the denominator β ensures the normalization of weights in d_x. Note that V_n and S_{n-k} have different resolutions; sample positions in the kernel domain happen to be aligned with the center of V_n's voxels, but density values from S_{n-k} must be computed by tri-linear interpolation.

The distance kernel assigns larger weights to those samples in V_n that are prone to disappear (those which most differ with S_{n-k}). As both g and d_x are combined into f_x, smoothing or sharpening is done depending on their weights, which are local to the filtered sample position x; homogeneous regions will provoke homogeneous distance kernels, thus giving g_x greater influence, whereas feature regions will provide more characteristic distance kernels for the feature selection task.

3. Results and Discussion

We have compared our results with several other downsampling techniques: averaging 2^3 voxels into 1, flat subsampling, subsampling after filtering (Gaussian and bilateral) and Topology-Guided. Average downsampling usually achieves acceptable results at the expense of little computational effort. Gaussian filtering provides smoother results but excessive smoothing leads to information loss, as the scalar field changes excessively and thus the original transfer function mapping is no longer valid for the downsampled volume. Bilateral filters are not especially well suited for downsampling; while they perform edge-preserving smoothing, they still produce aliasing similarly to flat subsampling. Topology-Guided downsampling succeeds in preserving features at the expense of excessively noisy results. Figure 1 shows a comparison of some techniques; where other approaches tend to make features disappear (ureter, ribs, catheter, etc.), our method preserves them, yet in zones with less fine structures our results are similar to using a Gaussian kernel. The poster shows additional examples.

Without any optimization, running on a commodity PC (Intel Core i7 CPU, 8GB RAM) our algorithm takes up to 4 minutes in order to compute the downsampling for the thorax model from 512^3 to 128^3 voxels. Although it is a few times more costly than standard Gaussian-downsampling, it is affordable for a preprocess and scales linearly with the size of the input dataset, plus it clearly achieves higher quality results.

We plan to extent our tests studying the effects of this new filter with higher resolution datasets and increasing the number of downsampling iterations to see how far it can preserve structures. Moreover, we will also optimize the algorithm by taking advantage of the property of separability of the first step.

References

[BHP14] BEYER J., HADWIGER M., , PFISTER H.: A survey of GPU-based large-scale volume visualization. *Proceedings EuroVis 2014* (2014). 1

[BNS01] BOADA I., NAVAZO I., SCOPIGNO R.: Multiresolution volume visualization with a texture-based octree. *The Visual Computer 17*, 3 (2001), 185–197. 2

[BRGIG*14] BALSA RODRÍGUEZ M., GOBBETTI E., IGLESIAS GUITIÁN J. A., MAKHINYA M., MARTON F., PAJAROLA R., SUTER S.: State-of-the-art in compressed GPU-based direct volume rendering. *Computer Graphics Forum 33*, 6 (2014), 77–100. 1

[GWGS02] GUTHE S., WAND M., GONSER J., STRASSER W.: Interactive rendering of large volume data sets. In *Proceedings of the Conference on Visualization '02* (Washington, DC, USA, 2002), VIS '02, IEEE Computer Society, pp. 53–60. 1

[HKRs*06] HADWIGER M., KNISS J. M., REZK-SALAMA C., WEISKOPF D., ENGEL K.: *Real-time Volume Graphics*. A. K. Peters, Ltd., Natick, MA, USA, 2006. 1

[KB08] KRAUS M., BÃIJRGER K.: Interpolating and downsampling RGBA volume data, 2008. 1

[KE01] KRAUS M., ERTL T.: Topology-Guided Downsampling. In *Volume Graphics* (2001), Mueller K., Kaufman A., (Eds.), The Eurographics Association. 2

[LHJ99] LAMAR E., HAMANN B., JOY K. I.: Multiresolution techniques for interactive texture-based volume visualization. In *Proceedings of the Conference on Visualization '99: Celebrating Ten Years* (Los Alamitos, CA, USA, 1999), VIS '99, IEEE Computer Society Press, pp. 355–361. 1

[LW90] LEVOY M., WHITAKER R.: Gaze-directed volume rendering. *SIGGRAPH Comput. Graph. 24*, 2 (Feb. 1990), 217–223. 2

[SHKM14] SICAT R., HADWIGER M., KRUGER J., MOLLER T.: Sparse PDF volumes for consistent multi-resolution volume rendering. *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Visualization) 20*, 12 (2014), 2417–2426. 2

[YMC06] YOUNESY H., MÖLLER T., CARR H.: Improving the quality of multi-resolution volume rendering. In *Proceedings of the Eighth Joint Eurographics/IEEE VGTC conference on Visualization* (2006), Eurographics Association, pp. 251–258. 2