Eurographics Conference on Visualization (EuroVis) (2015)
R. Maciejewski and F. Marton (Editors)

Poster

Interactive Free-Form Volume Editing

Kihyuk Kim[†] and Sungkil Lee[‡]

Sungkyunkwan University, South Korea

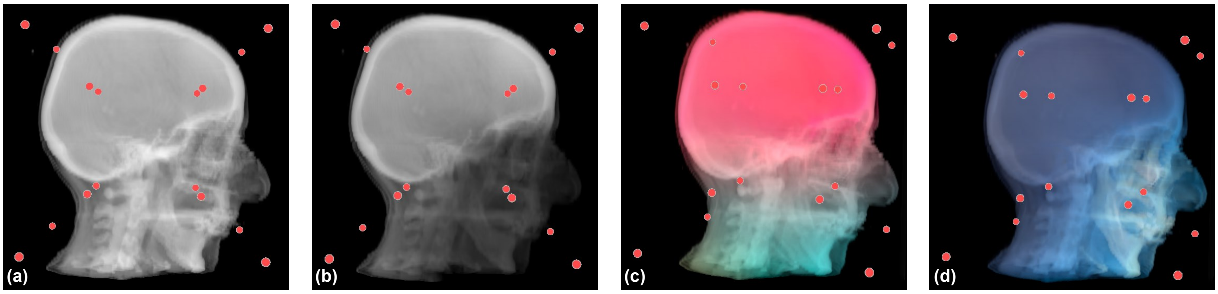

Figure 1: *Manipulation of volume attributes by our free-form volume editing: (a) input volume (Skull head), (b) feature highlighting by emphasizing density around brain, (c) hue manipulation, (d) composition of global (blue) color and local colors.*

Abstract

Volume editing with moving least squares is one of the effective schemes to achieve continuous and smooth deformation of existing volumes, but its interactive authoring has not been explored extensively. We present a framework for interactive editing of volume data with free-form deformation, which provides intuitive and interactive feedback on the fly. Given control points, we extend moving least squares with their visual metaphor to further encompass non-spatial attributes including lightness, density, and hues. Furthermore, a full GPU implementation of our framework achieves with instant real-time feedback with quick-and-easy volume editing metaphor.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques

---

1. Introduction and Related Work

Effective editing of volume data is of interest of volume designers, who strive to create volumetric scenes beyond realistic capture. Intuitive and interactive feedback is crucial in creating and modifying appearance of volumes. Many editing schemes have been suggested in the past, including volume sculpting [CMMM00], direct surface painting and annotation [BKW08], and scattered data interpolation [CSC07, ZLWK12]. Among them, ones relying on moving least squares (MLS) [ZLWK12] are particularly well suited for continuous and smooth deformation of spatial attributes (e.g., displacements) of existing volumes. While they have great potentials in interactive volume authoring, their utility has been rather limited to some extents (in particular, deformation of spatial-only attributes).

This poster presents a framework for interactive editing of volume data with MLS. While the previous work [ZLWK12] focused on texture-guided weighting for plausible deformation of voxels, our framework is distinguished for its visual interaction metaphors and utility of editing non-spatial attributes including lightness, hue, and density (Figures 1 and 2). Our framework allows us to attain quick-and-easy prototyping of volume deformation with instant feedback, which has a potential for authoring of volume animation/streaming.

[†] kihyuk@skku.edu
[‡] sungkil@skku.edu. Corresponding author.

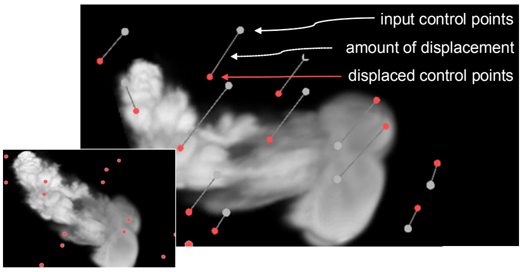

Figure 2: *Visual interaction metaphor of our volume editing of given an input volume (lower left).*

2. Interactive Volume Editing

Similarly to [ZLWK12], we use the 3D extension of 2D MLS deformation [SMW06]. Given a set of control points $\{p\}$ and their altered attributes $\{t\}$, MLS interpolation f_t of arbitrary points v can be defined as:

$$f_t(v) = \sum_j A_j (t_j - t^*) + t^*, \tag{1}$$

where A_j is a single scalar defined as:

$$A_j = (v - p^*) \left(\sum_i \hat{p}_i^{\mathsf{T}} w_i \hat{p}_i \right)^{-1} w_j \hat{p}_j^{\mathsf{T}}. \tag{2}$$

Here, p^* and t^* are centroids weighted with weight w (the product of inverse-distance weights and user importance), and derive $\hat{p}_i = p_i - p^*$.

The type of attribute t determines the type of output $f_t(v)$. For instance, $t(v)$ of voxel density yields altered voxel densities. Likewise, t of spatial position finds deformed positions.

2.1. Dynamic Volume Attributes

We generate fixed number of control points (e.g., 16 points), aligned with the bounding volume yet with inner offsets. Each control point tracks values of position, deformation matrix, and user importance, which is necessary to evaluate (2); spatial displacement is defined by a transformation matrix. Non-spatial attributes include lightness, voxel density, and hues (a^*b^* in CIE $L^*a^*b^*$ color space).

Unlike spatial attributes, batch editing of non-spatial attributes is convenient, altering all the attributes simultaneously and controlling per-point details later. For this scheme, we additionally define editable global offsets for non-spatial attributes; the global editing mode is activated when user selects background area. As a result, the final values of non-spatial attributes are the sum of local and global alterations.

2.2. Visual Interaction Metaphor

Our framework supports free-form editing with visualization of input and deformed positions (gray and red points shown in Figure 2), which is particularly useful in editing spatial attributes. Also, the directions and amounts of deformation vector are shown with lines connecting source and output positions. When a user clicks on the screen (when using mice, left/right click is for spatial/non-spatial attributes, respectively), selection mode is enabled. When a nearby control point is found, local editing mode is selected; otherwise, global mode is selected for batch editing.

With respect to simple 1-D attributes (density and lightness), vertical movement of mouse controls the amount of alteration. Similarly, 2D hues (a and b in Lab space; equivalent to RG and BY chromaticities) can be naturally mapped to horizontal and vertical movements.

As for spatial deformation, it may not be straightforward without fixing one axis. We propose an easy-to-control interaction scheme. We only define planar spatial deformation, which is perpendicular to a view vector (or parallel along the image plane). Hence, we first rotate a view to find the translation plane, and a control point is moved only within the plane.

3. Implementation and Results

Our framework was implemented fully on GPU (OpenGL) on an Intel i7 3.6 GHz machine with NVIDIA GeForce GTX 780Ti. When editing mode is activated, the attributes are tracked and their resulting volumes are updated on the fly. Input/output volumes are represented as 3D textures, and editing of output volumes are updated by slice-wise rendering.

Figure 1 shows example volumes edited with our framework, which demonstrates feature highlighting, hue manipulation, local/global attributes. Figure 2 shows how to apply spatial deformation with visual interaction metaphor.

Performance of volume editing and their rendering (using 16 control points and ray casting) reaches real-time frame rates. For the cloud data (a resolution of $100^2 \times 40$), the average frame rates reach 330.1 Hz. Those of the skull volume data (a resolution of $256^2 \times 106$) reach 44.0 Hz.

4. Discussion

While our framework already proved its utility in rapid prototyping of volume deformation, it is still limited in terms of definition of control points. More sophisticated definition of control points considering local density distribution and animated sequences would be great future work. While our framework allows us to deform attributes freely, care has to be taken with density values to avoid potential misinterpretations when analyzing medical or other scientific datasets.

Acknowledgments

This work was supported by the Mid-career and Global Frontier (on *Human-centered Interaction for Coexistence*) R&D programs through the NRF grants funded by the Korea Government (MSIP) (Nos. 2012R1A2A2A01045719, 2012M3A6A3055695).

References

[BKW08] BURGER K., KRUGER J., WESTERMANN R.: Direct volume editing. *IEEE Trans. Visualization and Computer Graphics 14*, 6 (2008), 1388–1395. 1

[CMMM00] CHANDRU V., MAHESH N., MANIVANNAN M., MANOHAR S.: Volume sculpting and keyframe animation system. In *Proc. Computer Animation* (2000), pp. 134–139. 1

[CSC07] CORREA C. D., SILVER D., CHEN M.: Volume deformation via scattered data interpolation. In *Proc. Volume Graphics* (2007), pp. 91–98. 1

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. In *ACM Tran. Graphics* (2006), vol. 25, ACM, pp. 533–540. 2

[ZLWK12] ZHAO X., LI B., WANG L., KAUFMAN A.: Texture-guided volumetric deformation and visualization using 3D moving least squares. *The Visual Computer 28*, 2 (2012), 193–204. 1, 2