Master’s Thesis

Intuitive Volume Deformation Authoring Framework Using Moving Least Squares with Density-Aware Weighting

Soonhyeon Kwon

Department of Electrical and Computer Engineering
The Graduate School
Sungkyunkwan University
Master’s Thesis

Intuitive Volume Deformation Authoring Framework Using Moving Least Squares with Density-Aware Weighting

Soonhyeon Kwon

Department of Electrical and Computer Engineering
The Graduate School
Sungkyunkwan University
Intuitive Volume Deformation Authoring Framework Using Moving Least Squares with Density-Aware Weighting

Soonhyeon Kwon

A Master’s Thesis Submitted to the Department of Electrical and Computer Engineering and the Graduate School of Sungkyunkwan University in partial fulfillment of the requirements for the degree of Master of Science in Engineering

October 2019

Approved by
Sungkil Lee
Major Advisor
This certifies that the Master’s Thesis of Soonhyeon Kwon is approved.

Thesis Supervisor:

Committee Chairman:

Committee Member 1:

The Graduate School
Sungkyunkwan University
December 2019
# Contents

List of Tables ........................................ ii
List of Figures ........................................ ii
Abstract ............................................... iii
1. Introduction ........................................... 1
2. Related Work ......................................... 4
  2.1 F+C Volume Visualization .......................... 4
  2.2 Volume Deformation ................................. 5
  2.3 Direct Volume Manipulation ......................... 7
3. Our Extensions ....................................... 9
  3.1 Non-spatial MLS ................................... 9
  3.2 MLS for Heterogeneous Volumes .................... 10
  3.3 Efficient Volume Deformation Synthesis .......... 14
4. Spatiotemporal Volume Editing ......................... 16
  4.1. Control Points and Visual Metaphors .............. 16
  4.2. Interaction Scheme and Spatiotemporal Extension .... 18
5. Results ............................................... 22
6. Discussion ........................................... 33
7. Conclusion ........................................... 35
References ............................................. 36
Korean Abstract ........................................... 39
Abstract

Intuitive Volume Deformation Authoring Framework Using Moving Least Squares with Density-Aware Weighting

Volume visualization and deformation techniques have been studied for improvement of visual interpretation. Number of studies with physically accurate computation and fine-tuned parameters has been proposed, while density-aware heterogeneous volume deformation with interactive and instant feedback have seldomly been studied.

In this paper, we present an intuitive volume deformation authoring framework. A novel density-aware weighting metric is introduced for importance control from data, by extending previous technique using moving least squares. The deformation can be applied to both heterogenous and homogeneous volume. Also, it can be applied for non-spatial attributes, as well as spatial one. We also present up- and downsampling for volume deformation synthesis for performance efficiency. Then we introduce a visual metaphor and planar scheme for an instant feedback of an user interaction to allow intuitive authoring of volume data sets. Finally, with supporting spatiotemporal editing scheme, our framework is implemented fully on graphics processors, allowing interactive prototyping volumes.
Keywords: Moving least squares, Heterogenous volumetric data, Volume deformation, Volume animation, Volume Visualization
I. Introduction

Scientific volumetric data sets (volumes) are used for handling scientific values in three-dimensional space. Typical examples can be a group of planar images, however, their visual appearance can be varied to interpret them effectively. Previous visualization techniques are focused on mapping data fields into visual properties such as an opacity and a color. Recent development of hardwares presents a solution for interactive volume visualization, and direct volume rendering [1] is a typical technique using current graphics processing units (GPUs).

Recent interactive techniques allow instant feedback to users, in direct volume rendering, transfer function must be well-designed to visualize data appropriately. Also, intuitive and efficient authoring of volume deformation remains challenging. This led to previous studies focused on physics such as finite element methods [2] and fluid dynamics [3][4]. Instead of fine-tuning of transfer function, focus-context (F+C) distinction in information visualization can inspire to relax spatial structure constraints [5].

For more intuitive method, non-physically-based deformation has been proposed. It is based on straightforward controls such as proxy structures (boxes or points) and keyframing [6]. In particular, moving least squares (MLS) [7] showed volume deformation based on the scattered data interpolation,
proving its effectiveness.

In volume applications, assuming a consistent density is useful to optimize computational cost. However, natural objects with complicated materials cannot be expressed in such simple way [21]. In contrast to homogeneous volumes, heterogeneous volumes are able to express fine structures and materials by varying distribution of a medium. However, density should be calculated on the fly, making calculation more complex. Due to its high complexity, techniques such as numerically coarsening structures [22] have been studied for scalable deformation of heterogeneous volumes, while number of techniques [23] are focusing on elasticity of materials, instead of density itself.

In this paper, we present a framework for intuitive volume animation authoring for heterogeneous volumes with density-aware weighting scheme. Fig. 1. shows a result overview of our framework. Our contributions include:

- Extension of MLS approach [7] with density-aware weighting metric for heterogeneous volume deformation
- An upsampling scheme for efficient volume synthesizing
- Visual metaphors for intuitive volume editing.
Fig. 1. Result overview of our framework.
II. Related Work

In this section, we briefly summarize previous studies related with our work: F+C volume visualization, volume deformation and direct volume manipulation.

1. F+C Volume Visualization

Cohen et al. [5] proposed F+C volume visualization for better data readability of large-scale data, by spatially distorting parts of volume and emphasizing details.

Krüger et al. [8] introduced ClearView, a GPU-based implementation of F+C volume visualization, by changing non-deformative property such as opacity to control visibility of context.

Wang et al. [9] expanded F+C volume visualization by constraining distortion bound and introducing an energy optimization, with considering screen displaying.
2. Volume Deformation

Direct control of volumetric fields is challenging, hence common approaches focus on using proxy geometries to manipulate data easily. While earlier example takes volumes as a geometric objects [10], a technique using bounding boxes [11] allows easier controllability with sacrificing detailed options. Later techniques are improving local controllability [12].

Applying skeletal animation of volumes is common solution for animations of human or mammals. Endpoint–based animation can be solved by inverse kinematics [13]. Although controls of endpoint can be easy, rendering and simulating skeletal animation can be very costly.

Other common research fields contain scattered data interpolation to provide free–form deformation. While using octree [14] and slicing to 2D [15] are proposed earlier, MLS–based approaches have been studied recently. MLS–based deformation uses finite control points instead of traditional grad–like structures, with providing similar quality. By using affine transformation, it can be applied to both 2D images [16] and 3D volumes [7]. While earlier MLS for 2D image deformation exploits polygonal rasterization to deal with forward approach, which cannot be applied to 3D volume with GPU rendering. Hence, backward mapping is used in the previous approach.
In spatial MLS interpolation, a set of control points \( \{x\} \) and their target points \( \{y\} \) are controlling the spatial deformations. While typical examples hold \( \{x\} \) as fixed points, our framework allows them to move. Let an arbitrary position as \( u \) and its deformation as \( v \). Typical local weights are based on radial distance function: \( w_i = 1/d_i \), where

\[
d_i = |v - y_i|^n ,
\]

where \( n \) is the power (typical choice is \( 2–4 \)). The backward mapping is used to compute the weight on the deformed point. Now, weighted centroids of \( \{x\} \) and \( \{y\} \) can be defined as

\[
\hat{x} = \frac{\sum_i w_i x_i}{\sum_i w_i} \quad \text{and} \quad \hat{y} = \frac{\sum_i w_i y_i}{\sum_i w_i} .
\]

If affine transformations are associated with \( \{x\} \) and \( \{y\} \), target of MLS can be changed to the following minimization problem:

\[
\sum_i w_i |p_i - Mq_i|^2 ,
\]

where \( p_i = x_i - \hat{x} \), \( q_i = y_i - \hat{y} \), and \( M \) is a transformation matrix that linearly deforms to the source points.
Now, equation (3) can be minimized by following solution $M$:

$$M = \left( \sum w_i q_i q_i^T \right)^{-1} \sum w_j q_j p_j.$$  \hspace{1cm} (4)

By least squares objective function and normal-equation solution, the spatial interpolation $f_s$ at point $v$ can be written as follows:

$$f_s(v) = \sum T_j(v)(x_j - \hat{x}) + \hat{x}.$$  \hspace{1cm} (5)

Previous studies [7][16] are providing details of mathematical analysis.

Applying general affine transformations including shearing and non-uniform scaling, is an implication for above derivations.

### 3. Direct Volume Manipulation

In direct volume manipulation, existing voxels are manipulated directly. Sculpting interface [19] and Editing by painting [20] are examples of direct volume manipulation.

Although this approach is direct and has fast performance, to get a plausible result, precise controls and professional skills are required. Also, temporal
interpolation cannot be easily achieved; in contrast to proxy structure based approaches, entire volume data must be stored into memory devices every keyframe, making memory requirements too costly.
III. Our Extensions

In this section, we introduce our extensions: non-spatial MLS, MLS for density-aware distance weighting and volume upsampling.

1. Non-spatial MLS

From we reviewed previous MLS deformation [7][16] in section II, our work extends spatial attributes straightforwardly to non-spatial attributes \( \{ \mathbf{z} \} \). Centroids of them can be summarized as similar as equation (2) as follows:

\[
\hat{\mathbf{z}} = \frac{\sum_{i} w_i \mathbf{z}_i}{\sum_{i} w_i}.
\]  

(6)

Also, interpolation at \( \mathbf{v} \) can be described as similar as equation (5):

\[
f_s(\mathbf{v}) = \sum_{j} T_j(\mathbf{v}) (\mathbf{z}_j - \hat{\mathbf{z}}) + \hat{\mathbf{x}}.
\]  

(7)
2. MLS for Heterogeneous Volumes

Previous approaches incorporate heterogeneous volumes rarely found. We extends the MLS model with density–aware weighting, allowing temporal control further.

Typical examples of heterogenous volumes contain medical imaging data sets, such as a skull and skin tissues. They have different properties, deformations may need to change. To support this problem, our work extends weighting scheme, by introducing density–aware distance metric. As equation (1) shows local weights, we redefine it as follows:

\[ d_i = \frac{1}{\rho(v, y_i)} |v - y_i|^n, \]  

where \( \rho(v, y_i) \) is a density accumulation from \( v \) to \( y_i \). Then we use ray casting to get it numerically, rather than analytical measure that is hard to get:

\[ \rho(v, y_i) = \frac{1}{\phi(v, y_i)} \int_v^{y_i} I(t) dt, \]  

where \( I(t) \) is accumulated density (or intensity) by a ray at point \( t \). Then we apply normalization factor \( \phi(v, y_i) \) as minimum density value.
Iterative synthesis to find $\rho(v, y_i)$ from ray casting requires multi-pass rendering, which is very costly. We introduce an analogical solution for finding $\rho(v, y_i)$, which is using $\rho(u, y_i)$ instead. Linearity preservation in affine deformation and using the source volume for casting ray, now we may find $\rho(u, y_i)$ in a single-pass synthesis. This can be summarized as follows:

$$\hat{d}_i = \frac{1}{\rho(u, y_i)} |v - y_i|^n.$$  \hspace{1cm} (10)

Now, dealing with an equation (10) has an important problem: after the deformation, we cannot find true $u$ properly. Instead, we extend 2D image warping solution [16], fixed-point iteration, to 3D volume. Displacement from a voxel $v$ to the source point $u$ can be written as:

$$\delta(u) = v - u = v - f_s(v).$$ \hspace{1cm} (11)

Then we estimate $u$ as an iterative way, and $(k+1)$-th updating can be written as:

$$u_{k+1} = u \leftarrow \hat{u}_{k} := v - \delta(u_k)$$ \hspace{1cm} (12)

where initial $u_0$ is a initial seed and estimation is performed at convergent fixed point.
Now, in practically, the initial value of equation (12) can be set by MLS without density-aware weighting. To make the equation to converge, discontinuities in scalar fields must be removed; by MLS, volumes are interpolated smoothly enough to flatten them. By speed of the convergence in practice, iterations can be set to 2–3 times for balanced quality.

We realized that we can introduce faster ray casting for interpolation purpose by coarser accumulation. Still, coarser accumulation cannot cover a thin structure (see Fig. 2), instead, we propose conic ray casting, similar to the voxel cone tracing [18]. Implementation can be easily done by pre-computing 3D mipmap textures and sampling with level of details in lookup.
Fig. 2. Comparison between (a) coarser linear ray casting and (b) conic ray casting.
3. Efficient Volume Deformation Synthesis

Significant processing overhead occurs when the least square is solved for every voxel, we propose effective optimization. We focused on costly cases of MLS interpolation. While previous approach using piecewise linear patches [12] are proposed, limitations of early GPUs make it hard to implement. However, recent GPUs can realize it, hence we revisit the idea and separating controls and details volume to reduce cost.

Typically the number of control points is significantly smaller than the number of voxels in MLS deformation. Hence, we separate volume into control volume (containing texture coordinates) and detail volume (source volume). Then, we can reduce a control volume to a lower resolution by downsampling. Then, deformation are estimated in coarser control volume, which makes performance significantly faster. Then, the deformation is synthesized with upsampling to detail volume. Fig. 3. shows an example of this stage. In our experiments, typically peak signal-to-noise ratio (PSNR) is roughly 26 - 29 dB.

In control volume downsampling, we select relatively low factor (e.g. × 0.5 to all axis). For general cases, usual Gaussian upsampling can be applied, although details such as sharp edges can be smoothed by it. Inspired by Gouraud and Phong shading, upsampling of spatial positions can be a better solution.
Fig. 3. Comparison by synthesizing scheme.
IV. Spatiotemporal Volume Editing

In this section, our volume deformation framework with visual metaphors for user interaction and spatiotemporal deformation support.

1. Control Point Definition and Visual Metaphor

As we already described in section 3.2, our volume deformation is extended to non-spatial attributes (e.g., color, and density) as well as spatial attributes. Here, we add temporal duration for each keyframe, to support spatiotemporal deformation (i.e., animation) without duplicating volumes.

In our framework, intuitive visual metaphors are provided for every control points (see fig. 4.). A number of control points is fixed for different keyframes. We used inner offsets of bounding volumes for point initialization.

For spatial attributes, source positions (gray dots) and displaced positions (green dots) are shown as dots with different colors, while connected gray lines between them show direction and amount of displacement.

In contrast, non-spatial attributes are directly reflected in rendering, thus, no additional visualization is required.
Fig. 4. Our visual metaphors for control points.
2. Interaction Scheme and Spatiotemporal Extension

In typical manipulation of 3D models, translation axis are limited for precise translation. Because axes visualization is mandatory and axis must be chosen, it is not very intuitive way. In contrast, our intuitive approaches employ a translation plane like panning a camera in 3D navigation (see fig. 5.). We limit movements of control points in the viewport plane, and users can rotate the plane when they need to.

![Planar interaction scheme](image)

Fig. 5. Planar interaction scheme.

Also, this scheme can be easily applied to non-spatial attributes. Fig. 6. shows alternation of hue values, deploying a* and b* from CIE L*a*b* color space, in planar axes.
Finally, we extend our framework eligible to manipulating spatiotemporal editing with keyframe animation. Fig. 7. shows extended visual metaphors for spatiotemporal editing. Instead of applying transition overlay directly, we choose not very intuitive way, because visualization can be too complex by many involving variables. Final frame–by–frame editing examples are shown in fig. 8.
Fig. 7. Visual metaphors for spatiotemporal editing.
Fig. 8. Frame-by-frame editing example along five key frames.
V. Results

We selected OpenGL Application Programming Interface for our implementation. Both volume rendering and authoring is fully operated on GPU. Fig. 9. shows performance measurements of five volume data sets (namely the Skull, Abdomen, Foot, Smoke, and Lobster), on Intel i7 3.0 GHz with NVIDIA GeForce GTX 980 Ti. We used stepwise ray casting (step size of 1024) for volume rendering. For control volumes resolution in our measurement, we chose a half resolution for each axis. Fig. 9 shows overall summary of our experiments.

![Graph showing measured frame time (ms) by combinations of our solution.](image)

Fig. 9. Measured frame time (ms) by combinations of our solution.

The dimensions and appearances of the five volume data sets are summarized in fig. 10.

In homogeneous volume deformation, using control volume causes texture
lookup too many time, making the performance 2 times slower. Thus, MLS without using control volume is faster.

However, in heterogenous volume deformation, introducing control volume reduces computational costs significantly, roughly speedups about up to 350%. Also, conic ray tracing reduces up to 30% of computational costs. Combining two techniques, the frame time can be reduced to interactive frame rates. But still it is slower than homogeneous volume deformation, because ray casting of each control points is required in animation.

Table 1 shows performance evaluation for different numbers of control points. The number of control points is a major factor of computational costs. In our evaluation, we chose 0, 8, 16, 24, and 32 control points for the test. Tests show that performances of our algorithm follows the number of control points. As we described our framework does not require huge amount of control points; in our experiments, we chose 16 points that is even enough for complex heterogenous deformation.
Table 1. Performance evaluation for different numbers of control points.

Note that 0 control points indicate we measured only rendering time without deformation.
Fig. 10. Five volume data sets.
We measured both homogeneous and heterogeneous deformation for performance evaluation. Example images of both deformations are shown in fig. 11. Fig. 12-16 show authoring examples of the five volume data sets with animated sequence.

Fig. 11. Example of homogeneous and heterogeneous deformation.
Fig. 12. Homogeneous deformation applied to Skull data set.
Fig. 13. Homogeneous deformation applied to *Abdomen* data set, with complex control points.
Fig. 14. Heterogeneous deformation applied to Foot data set: the flesh deforms more than bones.
Fig. 15. Deformation of non-spatial attributes applied to *Smoke* data set: lightness, hues, and density changed.
Fig. 16. Overall deformation applied to *Lobster* data set; both spatial and non-spatial attributes changed.
Examples show that authoring volume deformation and animation can be easily performed for various scenarios, including both spatial and non-spatial, homo- and heterogenous deformations.

Also, since our framework use intuitive visual metaphors, designers can save time and efforts: trained designer in our experiments, less than a half hour required for a single shot.
VI. Discussion

Our framework is aimed to edit volumes by deformation-based technique. Volume space expansion cannot be performed by our framework, because amounts of expansion cannot be easily solved.

MLS-based deformation technique limits a result smoother, and sharp details can be lost. To solve this problem, more sophisticated deformation model would be required.

In our performance evaluation for heterogeneous deformation with density-aware weighting, we optimized the performance to fit interactive frame rates, by introducing conic ray tracing and control volume. Since our technique is based on ray casting based scheme, computational cost can be reduced without relying it.

Also, our deformation is performed by backward mapping, however, to deal with it, complicated formulations and weighting schemes are required. Forward mapping can solve this problem. However, applying it to volume deformation is not an easy problem in contrast to 2D image deformation.

Another notable limitation is caused from point-based deformation control. At first time, control points must be initialized; however, automatic control point initialization is not a simple problem. Also, control point based deformation is
performed by atomic level. Integrating MLS and direct manipulation can be a possible solution for flexible volume editing.
VII. Conclusions

In this paper, we presented a volume authoring framework with heterogenous volume deformation. Our techniques extend 3D MLS with density-aware weighting scheme and speed-up solutions. For our framework, we demonstrated our visual metaphors and prototyping scheme. More than the visualization, our framework can be utilized for artistic ways. Editing and authoring animated volume data can be done more interactive and intuitive way by our framework.

With 3D MLS extended by a data-driven density-aware weighting scheme, our heterogeneous volume deformation technique is can be applied both spatial and non-spatial attribute and optimized by coarser control volumes. Our framework can be extended furthermore in terms of performance, flexibility and usability.
References


논문요약

밀도 가중형 이동최소자승법 기반 직관적 볼륨 변형
저작 프레임워크

권순현
전자전기컴퓨터공학과
성균관대학교

볼륨 데이터의 시각화와 변형 기법의 연구는 시각적 해석의 향상을 위하여 진행되어 왔다. 물리적으로 정확한 계산과 파라미터의 세밀한 조정을 포함하는 많은 기법들이 제안되어 왔으나, 사용자 반응에 대한 인터랙티브하며 즉각적인 피드백을 포함하는 밀도 기반의 불균등 볼륨에 대한 변형 기법은 아직 연구된 바가 많지 않다.

본 논문에서는 직관적인 볼륨 변형을 위한 저작 프레임워크를 제안한다. 이를 위해 이동최소자승법을 확장하여, 데이터의 중요도 조절을 위한 밀도에 기반한 가중법을 제안한다. 이를 통해 볼륨의 변형은 균등 볼륨 데이터와 비균등 볼륨 데이터 모두에 적용 가능하게 하며, 공간적인 요소와 함께 비공간적인 요소에도 변형을 적용할 수 있도록 한다. 또한 성능 향상을 위해 업샘플링과 다운샘플링을 통한 볼륨의 합성 기법을 제안한다. 나아가, 본 프레임워크에서 사용되는 시각화 표현법과 평면상에서의 상호작용 규칙을 제안함으로써, 사용자 반응에 대한 즉각적인 피드백이 가능하게 볼륨 데이터 편집을 위한 직관적인 사용자 상호작용이 가능하도록 한다. 마
지막으로, 본 프레임워크는 시공간적 편집이 가능하도록 확장되며, 그래픽스 프로세서 상에서 동작함으로써, 볼륨 데이터의 인터렉티브한 프로토타이핑이 가능하도록 한다.

주제어: 이동최소자승법, 불균등 볼륨 데이터, 볼륨 변형, 볼륨 애니메이션, 볼륨 시각화