BrainShapeToolKit for Generating Synthetic Data of Cortical Thickness

Hyunsoo Kim
School of Computing
KAIST
Daejeon, South Korea 34141
khskhs@kaist.ac.kr

Jinah Park
School of Computing
KAIST
Daejeon, South Korea 34141
jinahpark@kaist.ac.kr

November 25, 2024

Abstract

In this technical report, we introduce BrainShapeToolKit [1], a software stack for shape-based brain synthetic data generation. Instead of directly synthesizing the table, BrainShapeToolKit first synthesizes the shape of the brain as an intermediate representation and then generates the targeting measurements from the synthesized shapes. This intermediate representation can be used to generate various synthetic measurements, and we believe that it can preserve intricate correlations among actual measurements for synthetic data. We demonstrate the use case of BrainShapeToolKit by synthesizing cortical thickness data of normal brains from the ADNI dataset [1].

1 Introduction

Recently, synthetic data has emerged as a critical component in AI training. One of the biggest beneficiaries of synthetic data is medical AI models, as medical training data is costly to acquire, often impartial, and difficult to release due to privacy. For a comprehensive review of the applications of synthetic medical data, refer to [2].

One common approach to synthesizing tabular data is using end-to-end generative models such as GAN or VAE. These models directly synthesize tabular data, which follows the learned distribution of input data. However, their generation process is purely statistical and **without anatomical knowledge**, impeding the explainability of the resulting data. In addition, these models are **not expandable**. Once they are trained, expanding them to synthesize additional fields is so challenging that training a new model is a more viable option. Thus, the resulting models are often **monolithic**. A model is trained for one specific dataset of interest and is rarely reused in its life cycle.

These issues can be mitigated with the synthetic data generation process that utilizes anatomical structures. This approach is especially efficient for brain data because brain measurements are usually accompanied by brain imaging, so why not use it? We choose the anatomical shape as an intermediate representation for measurement synthesis rather than the images, as 1) the far fewer dimensions of shapes compared to images allow them to be synthesized easily, and 2) from shapes, anatomical measurements can be directly computed, unlike images which need additional processing. Therefore, given the anatomical shape of a brain, lightweight regression models can synthesize the anatomical measurement from the shape with small data quantities, which is a common condition in medical image analysis. These synthetic measurement models can be trained individually for the targeting measurement and can be used in any combination to generate a synthetic data table.

Hence, we propose BrainShapeToolKit, a modular approach to synthetic brain data based on anatomical knowledge. Unlike end-to-end generative models, which directly synthesize fields from latent variables sampled from tractable distribution, BrainShapeToolKit uses synthetic brain shapes as an intermediate representation. Then, measurements are derived from the synthetic shape, replicating the anatomical procedure of the actual measurements.

[1] The source code is available in https://github.com/SSTDV-Project/BrainShapeToolKit.

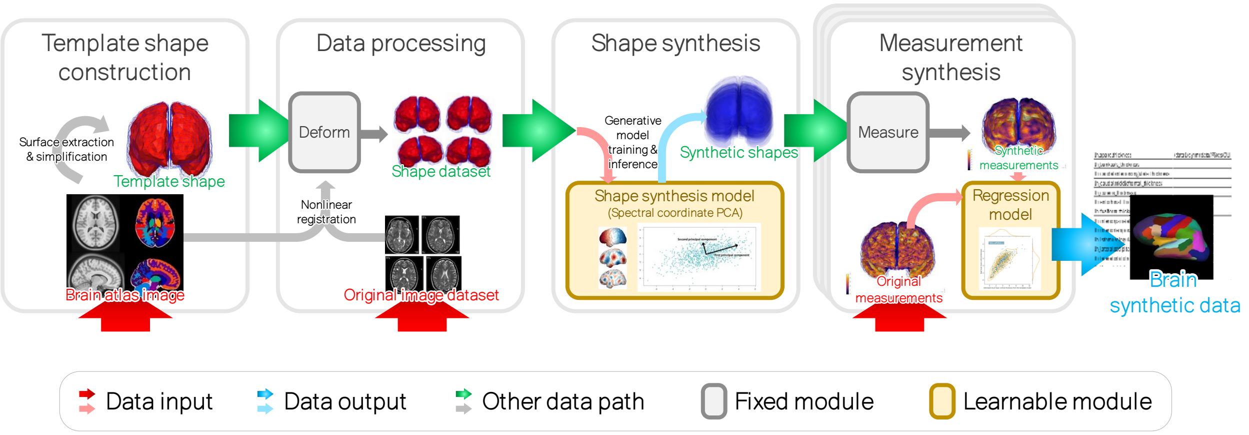

Figure 1: Overview of BrainShapeToolKit data synthesis pipeline.

We demonstrate BrainShapeToolKit using cortical thickness data from the brain. Cortical thickness measures the thickness of gray matter between the pial surface and the white matter surface of the brain. The data consists of average cortical thickness and standard deviation of 68 parcels defined in the Desikan-Killany atlas [3]., per subject. Comparing the distribution of 100 real and 100 synthetic data, we observe 0.95 on the KS complement metric and 0.0131 on the KL divergence metric.

2 Method

BrainShapeToolKit is designed to be extended to any use cases involving synthetic brain measurement data based on brain imaging. Figure 1 depicts the data synthesis pipeline of BrainShapeToolKit. It consists of four stages: template shape construction, data processing, shape synthesis, and measurement synthesis. They have modular designs so that each module can be improved and expanded according to the needs. In this section, we explain the roles of each module and how each works.

2.1 Template shape construction

Algorithm 1 illustrates the first two stages of the BrainShapeToolKit pipeline: template shape construction and data processing. For synthesizing cortical thickness data, the pial and white matter surfaces are extracted from the atlas image. FreeSurfer [4] is used to extract cortical surfaces from the atlas image as high-resolution surfaces.

The high-resolution surfaces, consisting of over 100k vertices, have too many degrees of freedom of their deformation for shape synthesis. Thus, the high-resolution surfaces are simplified to reduce the degrees of freedom in an appropriate range. For this purpose, a semi-convex hull method [5] is used to simplify the original high-resolution surfaces into low-resolution surfaces with uniform triangular meshes.

However, surfaces with too few degrees of freedom lack the expressive power for diverse deformation of cortical surfaces. Specifically, we want low-resolution surfaces that preserve deformations from datasets as much as possible. Thus, we choose one of the various degrees of simplified meshes that maintains the original deformation fields when the deformation is interpolated from the surface to the space with the cage deformation method [6]. The resulting template shape parameterizes brain deformation in downstream tasks.

Algorithm 1: Template construction and data processing

```
Input: Brain atlas image $I_0$, original image
       dataset $D_I = \{I, ...\}$
Output: Template shape $S_0 = (V_0, F_0)$, shape
        dataset $D_S = \{S = (V, F_0)\}$
begin
    /* Template shape construction */
    Segment ROI mask $\mathcal{R}$ from $I_0$;
    Construct high-res surface $S_H$ from $\mathcal{R}$;
    Simplify $S_H$ to get $S_0 = (V_0, F_0)$;
    /* Data processing */
    $D_S \leftarrow \emptyset$;
    for $I \in D$ do
        Find the mapping $\Psi$ from $I_0$ to $I$;
        Deform $V_0$ using $\Psi$ to get $V$;
        $D_S = D_S \cup \{S = (V, F_0)\}$;
    end
    Return $(S_0, D_S)$;
end
```

2.2 Data processing

BrainShapeToolKit uses the T1 brain MRI dataset to extract the original distribution of brain shapes, which the shape synthesis model will replicate. In this subsection, we describe the processing steps from T1 MRI to shape distribution.

BrainShapeToolKit utilizes the brain atlas image to extract correspondences and deformation fields from the original dataset via registration. The ICBM152 nonlinear symmetric atlas [7, 8] is used as an atlas image. The registration is performed using EasyReg [9, 10]. Resulting deformation field $\Psi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ from the atlas image $I_0 : \mathbb{R}^3 \rightarrow \mathbb{R}$ to a input image $I : \mathbb{R}^3 \rightarrow \mathbb{R}$ satisfies the following:

$$\forall x : I_0(x) \simeq I(\Psi(x))$$

Deformation fields are used to acquire the deformed template shape that matches the target surfaces in the image data. Algorithm 1 shows the process of deforming template shapes from image registration. By deforming the vertices of the template shape along with the deformation field, the resulting shape will match the same features on the image data as the template shape on the brain atlas. Specifically, for a given template shape $S_0 = (V_0, F_0)$, where the vertices $V_0 \in \mathbb{R}^{V\times 3}$ and face indices $F_0 \in \mathbb{N}^{F\times 3}$ form a mesh, the deformed shape $S = (V, F_0)$ is computed as $v^i = \Psi(v_0^i)$, where v_0^i denotes the i-th vertex in V_0, and v^i denotes the same for V

2.3 Shape synthesis

Using the dataset of the deformed template shapes extracted from the input images, a shape synthesis model is trained. This model generates a synthetic shape that follows the distribution of the input shape dataset. Specifically, the model will generate the displacements of the vertices in the template shape and then apply the generated displacements to the template shape to create a synthetic shape. Algorithm 2 illustrates the training of the shape synthesis model and the sampling of synthetic shapes from the model.

Algorithm 2: Shape and measurement synthesis

```
Input: Template shape $S_0 = (V_0, F_0)$, shape
       dataset $D_S = \{S = (V, F_0)\}$, original
       measurements $M \in \mathbb{R}^{|D_S|\times K}$, the
       number of synthetic samples $N$
Output: Shape synthesis model $\mathcal{M}_S$,
        measurement synthesis model $\mathcal{M}_M$,
        synthetic shapes $D'_S$, synthetic data
        $M' \in \mathbb{R}^{N\times K}$
begin
    /* Train $\mathcal{M}_S$                              */
    $\Phi \leftarrow$ Eigenbases of $\Delta_{S_0}$;
    $\Delta Y \leftarrow \{\Phi^T(V - V_0) \,|(V, F_0) \in D_S\}$;
    $\mathcal{M}_S \leftarrow$ PCA.fit($\Delta Y$);

    /* Train $\mathcal{M}_M$                              */
    $\Delta Y_R =$ PCA.transform($\mathcal{M}_S, \Delta Y$);
    $D_R \leftarrow V_0 + \Phi \Delta Y_R$;
    $M_S \leftarrow$ Measure($D_R$);
    $\mathcal{M}_M \leftarrow$ Regressor.fit($M_S, M$);

    /* Sample synthetic data                    */
    Sample $\Delta Y'$ from $\mathcal{M}_S$;
    $D'_S \leftarrow V_0 + \Phi \Delta Y'$;
    $M'_S \leftarrow$ Measure($D'_S$);
    $M' \leftarrow$ Regressor.predict($\mathcal{M}_M, M'_S$);
    Return ($\mathcal{M}_S, \mathcal{M}_M, D'_S, M'$);
end
```

BrainShapeToolKit uses a principal component-based generative model in a spectral deformation space for shape synthesis. Here, spectral coordinates are chosen over Cartesian coordinates as an intermediate representation of the deformation. While Cartesian coordinates describe the deformation of the shape by displacement vectors of each vertex, *i.e.*, $\Delta X = V' - V_0 \in \mathbb{R}^{V\times 3}$, spectral coordinates describe it by the linear combination of orthogonal bases, which are eigenfunctions of the Laplace-Beltrami operator L, *i.e.*, $\Delta Y = \Phi^T \Delta X \in \mathbb{R}^{V\times 3}$. Here, $\Phi \in \mathbb{R}^{V\times V}$ is a matrix with an eigenfunction ϕ_i as a column vector, where $L\phi_i = \lambda_i \phi_i$. Mathematically, the conversion from Cartesian to spectral coordinates maintains the distance L_2, since this transformation merely involves the rotation of points within $\mathbb{R}^V$, thus minimally affecting the results of the principal component analysis. However, because spectral coordinates can encode coarse-scale deformations more compactly, they are easier to explain the trends. In addition, spectral coordinates can easily truncate high-frequency noises if needed.

The shape synthesis model is trained as follows. First, spectral bases Φ are acquired from the template shape by solving for eigenfunctions of the Laplace-Beltrami operator L defined on the template shape. Then, the shape deformations of the input dataset, resulting from Section 2.2 and represented in Cartesian coordinates ΔX, are converted to spectral coordinates $\Delta Y = \Phi^T \Delta X$. In this step, no information is lost, as the full spectral bases are used, and each deformation in the dataset is processed independently. Next, a principal component analysis model is fitted using the spectral coordinates of the deformations in the dataset. This model captures the distributions within the dataset and reduces the degrees of freedom to $K = 32$ principal components. The resulting model encodes the shape distributions with the principal components $A \in \mathbb{R}^{3V\times K}$, the mean deformation $\mu \in \mathbb{R}^{3V}$, and the variances $\Sigma \in \mathbb{R}^K$ for each of the principal components.

Given the trained model, the new shape S' can be synthesized by sampling from the normal distribution in principal components space, using the mean and variances of the trained model:

$$\begin{aligned} z &\sim N(0, \Sigma) && \text{(Sampling from the normal distribution)} \\ \Delta Y &= \mu + Az && \text{(Mapping from reduced PCA space to spectral space)} \\ \Delta X &= \Phi \Delta Y && \text{(Mapping from spectral space to Cartesian space)} \\ S' &= (V_0 + \Delta X, F_0) && \text{(Applying the deformation to the template shape)} \end{aligned}$$

2.4 Measurement synthesis

BrainShapeToolKit synthesizes brain measurements, such as cortical thickness statistics, by measuring them on synthetic shapes. Here, each measurement model synthesizes a specific variable about brain anatomy. The measurement synthesis models can be trained independently and used in any combination. Because the synthetic measurements are correlated based on the synthetic shape from which they are measured, BrainShapeToolKit, while modular, can preserve correlations among the synthetic measurements. Algorithm 2 depicts the training of the measurement synthesis model and how it uses the regression model to map the measurements done in synthetic shapes to the distributions of real measurements.

Because the template shape is a simplified version of the original shape, the measurements may have a different distribution than the original. Thus, BrainShapeToolKit has a mapping layer between raw measurements on the synthetic shape and the resulting synthetic brain measurements, so the synthetic measurements follow the original distributions. This regression task is performed by XGBoost [11]. To address the different degrees of distortion by regions in the template shape, the mapping model also takes positional encoding in addition to the raw measurements on the synthetic shape.

3 Experiments: cortical thickness data

We applied BrainShapeToolKit to brain cortical thickness data for demonstration purposes. We used cortical thickness statistics (mean and standard deviation) from 68 parcels of 200 normal brain T1 MRIs, obtained from FreeSurfer [12]. For the cortical thickness synthesis task, we used a template shape consisting of pial and white matter surfaces of each brain hemisphere.

3.1 Evaluation

We evaluated the quality of the synthetic data using two metrics: KS complement and KL divergence. Both metrics evaluate the similarity of two probabilistic distributions. KS complement is based on KS statistics D_{KS}, which measures the largest absolute difference between two cumulative distributions. KS complement is defined by $1 - D_{KS}$, 1 if two distributions are identical, and 0 if their ranges do not overlap at all. KL divergence is widely used as a distribution-matching loss in deep learning. Conceptually, it represents how unlikely a distribution is to be from another distribution.

We measured KS complement and KL divergence between the kernel density estimations of both real and synthetic measurements. Because the data already describe the mean and standard deviation of the cortical thickness at each parcel, we sampled 10 values from the normal distribution of the same mean and standard deviation to simulate the distribution of the real measurements. Then, the Gaussian mixture kernel is fitted using the kernel density estimation from those sampled values. From the two Gaussian mixtures, KS statistics can be analytically derived, and KL divergence can be acquired using a Monte Carlo estimator

$$D_{KL} = \frac{1}{N} \sum_i \log \frac{p(x_i)}{q(x_i)},$$

where $p(x)$ and $q(x)$ denote the probabilistic density functions of two Gaussian mixtures and x_i is a sample from the distribution of $p(x)$.

We used 200 ADNI normal brain T1 images [1] as input data and synthesized 200 synthetic cortical thickness statistics. Then, we split the real and synthetic data into two equal parts and measured KL complement and KL divergence within and between them. Table 1 shows the results.

Comparing *Real vs Real* and *Real vs Synth.* results, we found that although the KS complement is nearly identical, the KL divergence is significantly lower in *Real vs Synth.* result. This result can be interpreted as that the synthetic

Table 1: Distribution similarity metrics between 100 cortical thickness real and synthetic statistics. All metrics show the average value of all 68 columns.

Comparison	KS Complement ↑	KL Divergence ↓
Real vs Real	0.9574	0.0281
Real vs Synth.	0.9537	0.0131
Synth. vs Synth.	0.9691	0.0124

data were able to capture the common cases, hence similar KS complement score, but were not able to capture the extreme or exceptional cases that the real data have, which are shown in the high KL divergence in *Real vs Real* results. *Synth. vs Synth.* result also supports this claim by showing that the statistics are more similar within the synthetic data compared to the real data with higher KS complement and lower KL divergence.

4 Conclusion

We propose BrainShapeToolKit, a modular approach for generating synthetic brain data. First, synthetic brain shapes are created, and then measurements are derived from these shapes, mimicking real anatomical procedures. By introducing the synthetic shape as an intermediate representation, additional fields can be synthesized by attaching measurement synthesis models, allowing the modular expansion of the synthetic data generative model. We tested BrainShapeToolKit with cortical thickness data and showed that a modular approach is viable for synthesizing local statistics data such as cortical thickness. For future work, generating images from synthetic shapes and modeling longitudinal changes of synthetic shapes would be considered.

References

[1] Ronald Carl Petersen, Paul S Aisen, Laurel A Beckett, Michael C Donohue, Anthony Collins Gamst, Danielle J Harvey, CR Jack Jr, William J Jagust, Leslie M Shaw, Arthur W Toga, et al. Alzheimer's disease neuroimaging initiative (adni) clinical characterization. *Neurology*, 74(3):201–209, 2010.

[2] Mauro Giuffrè and Dennis L Shung. Harnessing the power of synthetic data in healthcare: innovation, application, and privacy. *NPJ digital medicine*, 6(1):186, 2023.

[3] Rahul S Desikan, Florent Ségonne, Bruce Fischl, Brian T Quinn, Bradford C Dickerson, Deborah Blacker, Randy L Buckner, Anders M Dale, R Paul Maguire, Bradley T Hyman, et al. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, 31(3):968–980, 2006.

[4] Anders M Dale, Bruce Fischl, and Martin I Sereno. Cortical surface-based analysis: I. segmentation and surface reconstruction. *Neuroimage*, 9(2):179–194, 1999.

[5] Fatma Guney and Andreas Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4165–4175, 2015.

[6] Tao Ju, Scott Schaefer, and Joe Warren. Mean value coordinates for closed triangular meshes. In *ACM SIGGRAPH 2005 Papers*, pages 561–566. 2005.

[7] Vladimir S Fonov, Alan C Evans, Robert C McKinstry, C Robert Almli, and DL Collins. Unbiased nonlinear average age-appropriate brain templates from birth to adulthood. *NeuroImage*, 47:S102, 2009.

[8] Vladimir Fonov, Alan C Evans, Kelly Botteron, C Robert Almli, Robert C McKinstry, D Louis Collins, Brain Development Cooperative Group, et al. Unbiased average age-appropriate atlases for pediatric studies. *Neuroimage*, 54(1):313–327, 2011.

[9] Malte Hoffmann, Benjamin Billot, Douglas N Greve, Juan Eugenio Iglesias, Bruce Fischl, and Adrian V Dalca. Synthmorph: learning contrast-invariant registration without acquired images. *IEEE transactions on medical imaging*, 41(3):543–558, 2021.

[10] Juan Eugenio Iglesias. A ready-to-use machine learning tool for symmetric multi-modality registration of brain mri. *Scientific Reports*, 13(1):6657, 2023.

[11] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[12] Bruce Fischl and Anders M Dale. Measuring the thickness of the human cerebral cortex from magnetic resonance images. *Proceedings of the National Academy of Sciences*, 97(20):11050–11055, 2000.