

Received December 7, 2021, accepted January 2, 2022, date of publication January 6, 2022, date of current version January 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140810

Super-Resolution of 3D Brain MRI With Filter Learning Using Tensor Feature Clustering

SEONGSU PARK¹ AND JIN KYU GAHM²

¹Department of Information Convergence Engineering, Pusan National University, Busan 46241, Republic of Korea

²School of Computer Science and Engineering, Pusan National University, Busan 46241, Republic of Korea

Corresponding author: Jin Kyu Gahm (gahmj@pusan.ac.kr)

This work was supported in part by Institute of Information communications Technology Planning Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2020-0-01450, Artificial Intelligence Convergence Research Center [Pusan National University]); and in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2020R1C1C1008362).

ABSTRACT Surface-based analysis of magnetic resonance imaging (MRI) data of the brain plays an important role in clinical and research applications. To achieve accurate three-dimensional (3D) surface reconstruction, high-resolution (HR) MR image acquisition is needed. However, HR image acquisition is hindered by hardware limitations that result in long acquisition time and low spatial coverage. Single image super-resolution (SISR) can alleviate these problems by converting a low-resolution (LR) image to an HR image. However, unlike 2D SISR methods, conventional 3D methods incur a large computational cost and require abundant data. Further, 3D boundaries for surface reconstruction based on MR images have not been sufficiently investigated. We herein propose a cost-efficient novel regression-based framework for super-resolution of 3D brain MRI that directly analyzes 3D features by introducing a tensor using gradient information. We initially cluster features using tensors to create labels for both the training and testing stages. In the training stage, for each label, we collect LR patches and corresponding HR intensities to compute filters. In the testing stage, for each voxel, we construct a tensor to obtain a feature and predict the HR intensity using trained filters. We also propose a patch span reduction method by limiting patch orientation to reduce the orientation span and increase shape variety. Using only 30 masked T1-weighted brain MR volumes from the Human Connectome Project (HCP) 900 dataset, the proposed algorithm exhibited superior performance in terms of HR boundary recovery in the cerebral cortex as well as improved overall quality compared to conventional methods.

INDEX TERMS Super-resolution, surface-based analysis, tensor analysis, brain MRI, 3D image analysis.

I. INTRODUCTION

Magnetic resonance imaging (MRI) is a non-invasive medical imaging modality for characterizing tissue. The structural details that can be observed using this approach assist in the identification of not only tissue transformation but also malignant lesions based on abnormal tissue contrast. In particular, the structural details of the cortical and subcortical regions of the brain have been typically represented using surfaces that were reconstructed from three-dimensional (3D) MR volumes. Moreover, surface-based analysis plays an important role in the local tracking of disease based on surface registration in clinical and research settings [1], [2]. Therefore, high-resolution (HR) MRI is needed to achieve

accurate surface reconstruction to aid in diagnostic decisions [3]. However, high spatial resolution MR image acquisition is inhibited by hardware limitations that adversely affect the scanning time and signal-to-noise ratio (SNR), and introduce body motion artifacts [4], [5]. Single image super-resolution (SISR) aims to recover a high-resolution (HR) image from a given low-resolution (LR) image, and can potentially be utilized to address this problem. By exploiting SISR, an HR volume can be obtained by applying a super-resolution (SR) technique to an LR volume, thereby reducing the MR image acquisition time.

The SISR problem is an ill-posed inverse problem because the number of dimensions of an HR image is greater than that of an LR image, and the number of HR cases that can be estimated using LR data is infinite. The classical approach used in SISR involves estimating the values of pixels using

The associate editor coordinating the review of this manuscript and approving it for publication was Vishal Srivastava.

mathematical models used for processing LR images, such as interpolation [6], [7] or back-projection [8]. Although these methods are not time-consuming, they cannot represent high-frequency image structures (e.g., edge or textures); thus, they appear smoother than the original.

To solve this problem, an example-based SR using the information in images was proposed based on the high-frequency information of HR images. This approach reformulates the SISR problem as a restoration of the HR counterpart of the LR image by manifold learning of HR images. The initial example-based SR method aims to find the nearest neighbor in the LR image patch dictionary and to match it to the corresponding HR using the Markov Random Field [9]. Since this approach has high computational complexity, sparse coding and regression-based methods were proposed. Sparse coding methods attempt to find meaningful sparse representations of LR patches by compacting image patches to a low-dimensional vector at a low computational cost [10], [11]. Zeyde *et al.* [12] used a sparse dictionary with K-SVD [13] to improve image quality with a significant speed improvement. Regression-based methods commonly involve the building of regressors that approximate HR components given an LR patch, instead of using sparsity models for improved performance. Timofte *et al.* [14], [15] and Yang and Yang [16] used anchored neighbor regression and exemplars clustering to achieve better performance. To reduce the regressor searching time in these methods, hashing [17], [18] and forest-based searching structures [19], [20] have been proposed. Pérez-Pellitero *et al.* [21] proposed a patch span reduction method by collapsing variations of a patch transform (e.g., the symmetry and rotation) using a mirror symmetry transform-based distance to improve the performance of the regression-based method. However, unlike most studies that estimate HR images using an external database, dictionary, or regressors, Glasner *et al.* [22] attempted to downscale a single image into patches at different scales, and to subsequently estimate an HR image from the recurrence of the patches without a prior example.

More recently, deep learning-based approaches that use convolutional neural networks (CNN) have been proposed for the SR task in natural and MR images. Dong *et al.* [23] proposed the first approach that utilized the end-to-end deep learning model for SR (SRCNN), which directly learned the mapping of bicubic-interpolated LR to HR images. Subsequently, the accuracy was increased at a reduced computational cost using the LR image as an input, and learning LR upscaling methods such as transposed convolution [24], pixel shuffle [25] operator, and progressive upscaling [26]. In addition, networks with more layers and capacity [27] such as deep-residual [28], wide-residual [29], or couple-projection residual [30] networks were introduced for performance improvement. Dense networks [31], [32], attention modules [33]–[35], and generative adversarial networks [36] also contributed to the improvement of SR image quality.

Despite the rapid advancement of these SISR algorithms, most methods were originally designed only for two-dimensional (2D) images. However, MR images are acquired as 3D volumes. These 2D methods do not use the full 3D information of the MR volume to solve the 3D SR problem. Instead, most studies typically convert the 3D medical images into a stack of 2D slices projected along one specific axis (e.g., axial, sagittal, and coronal planes), apply the 2D algorithm on each slice separately, and merge the results into a stack of slices [37], [38]. In these approaches, 3D features that appear in a different dimension could be lost. To solve the 3D SR problem, therefore, a model that extracts 3D structural information is preferred. In 3D models, tensors have been used to represent 3D gradient features such as in the case of vessel segmentation in medical image analysis [39]–[41]. In deep learning, 3D CNN networks [42] have achieved state-of-the-art performance with residual densely connected generators and adversarial models [43]–[45]. However, these approaches are not only computationally intensive but also require an abundance of 3D datasets to optimize different parameters. However, obtaining large-scale datasets is expensive in MR applications. Moreover, most previous SISR approaches aimed to improve only the overall quality. For MR images, however, boundary regions such as gray-white matter boundaries and gyral and sulcal regions need to be more precisely restored for accurate surface reconstruction and surface-based analysis, but, to the best of our knowledge, there is limited research on the boundaries of 3D structures.

In this paper, we propose a cost-efficient novel 3D regression-based SR algorithm by introducing a tensor using 3D gradient information for feature extraction and filter training. The advantage of this algorithm is that it not only maintains high SR performance with a small dataset and a short training time but also recovers clear 3D boundaries for accurate surface reconstruction. In the proposed algorithm, we extract 3D features from a patch volume by constructing 3 by 3 tensors for patch label selection during the training and testing stages. To select a label, we create labels by clustering extracted features from tensors using defined distance measurement metrics. In the training stage, we compute tensors from the LR patches extracted from volumes of interest to find the label, and collect the patches and corresponding HR intensity values for each label. Filters are computed that approximate HR intensities from LR patches. In the testing stage, for each voxel, a tensor is constructed to obtain a label and to predict the HR intensity using corresponding filters. Furthermore, we suggest a patch span reduction method by limiting the orientation that can be augmented by flipping or transposing to increase the variety of shapes using a small dataset. In the experiments based on the Human Connectome Project (HCP) 900 dataset [46], visual and quantitative results are presented to show that the proposed algorithm outperforms conventional methods in terms of the recovery of MR boundary regions and the overall MR volumes.

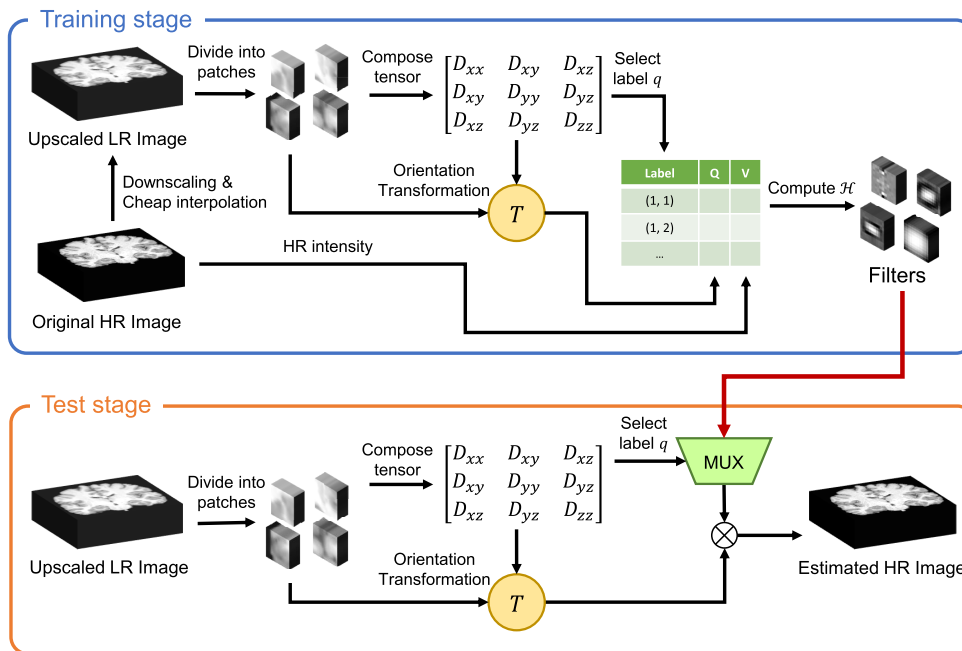


FIGURE 1. Overall architecture of the proposed algorithm. In the training stage, transformed LR patches and corresponding HR image intensities were acquired by accumulating matrices Q and V for each label to compute least-square filters. In the testing stage, for each voxel, a tensor was constructed to determine the label, and the corresponding filter was applied to the transformed patch.

Our contributions in this paper are listed below:

- We propose a novel SR method using 3D gradient information as tensor and span reduction.
- The method uses MRI tensor shape and orientation features to create 3D filters.
- The proposed algorithm delivers high SR performance with a small dataset and in a short training time.
- We also provide a design to evaluate SR performance in MRI cortical boundary region.
- We showed that the method outperforms conventional methods, particularly for HR boundary recovery.

The remainder of the paper is organized as follows.

In Section II, we propose a framework for 3D SR, including tensor calculation based on the patch volume, feature clustering, and a patch span reducing method. Experimental results are presented in Section III, wherein we compare the proposed algorithm and conventional methods using masked T1-weighted MR volumes. Finally, the main conclusions are summarized in Section IV.

II. METHOD

In this section, we first define the 3D SISR problem and present the details of our regression-based approach that was developed to address this issue. We also describe the construction of the tensors from 3D patches, the extraction of the tensor shape and orientation features, and the creation of labels by clustering using the extracted features. Finally, a patch span reduction method that limits the tensor orientation and determines the relationship between the elements of the tensor and the patch orientation is introduced.

A. PROBLEM FORMULATION AND PROPOSED ARCHITECTURE

In the 3D SISR problem, the relationship between LR and HR volumes can be defined as a linear model as follows:

$$\mathbf{y} = \mathbf{D}_s \mathbf{x}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{H_s \times W_s \times D_s}$ and $\mathbf{y} \in \mathbb{R}^{H \times W \times D}$ denote HR and the corresponding LR image, respectively. The operator $\mathbf{D}_s : \mathbb{R}^{H_s \times W_s \times D_s} \rightarrow \mathbb{R}^{H \times W \times D}$ downscales HR to an LR image using a factor of s along each axis. The objective of the SR task is to determine the inverse operator $\mathbf{F}_s : \mathbb{R}^{H \times W \times D} \rightarrow \mathbb{R}^{H_s \times W_s \times D_s}$ that minimizes the following errors:

$$\min_{\mathbf{F}_s} \|\mathbf{F}_s \mathbf{y} - \mathbf{x}\|_2^2. \tag{2}$$

In this paper, we address the 3D SISR problem for brain MRI using a regression-based method that involves approximating the HR intensity by applying a filter to an extracted LR patch. The overall proposed framework is illustrated in Fig. 1. In the training stage, for a given LR volume, we first cheap-interpolate the LR volume using factor s to match the size of \mathbf{y} . A tensor wherein each cubic patch $\mathbf{p} \in \mathbb{R}^n$ is sampled from the upscaled LR volume is then constructed, where n is the patch volume size. We extract the patch label q from the tensor, and classify similar patches for more accurate filter learning and evaluation. To obtain the label, we compute the shape and orientation features based on the tensor and predict the clusters from the pre-fitted two cluster models. Further details on the cluster models and features that were extracted from the tensors are given in

Section II.B. In addition, we transform the LR patch with extracted transformation Φ from the tensor and the details of the method are described in Section II.C. For each label q , the HR intensity $\mathbf{Y}_q \in \mathbb{R}^{1 \times l_q}$ is regressed on the corresponding transformed LR patches $\mathbf{P}_q \in \mathbb{R}^{n \times l_q}$, where l_q is the total number of collected similar patches. The computation of a filter $\mathbf{h}_q \in \mathbb{R}^{1 \times (n+1)}$ can be formulated as:

$$\mathbf{h}_q = \arg \min_{\mathbf{h}} \left| \mathbf{Y}_q - \mathbf{h} \begin{pmatrix} \mathbf{P}_q \\ \mathbf{1} \end{pmatrix} \right|^2, \quad (3)$$

where $\mathbf{1} \in \mathbb{R}^{1 \times l_q}$ is a vector with all the values set as 1.

For filter convergence in the right term of Eq. (3), \mathbf{P}_q stores 10^5 raw patches and requires approximately 500 MB of memory for each q . This causes memory shortage considering a large number of labels. Instead, we accumulate fixed-sized $\mathbf{Q}_q \in \mathbb{R}^{(n+1) \times (n+1)}$ and $\mathbf{V}_q \in \mathbb{R}^{1 \times (n+1)}$ matrices using an accumulation step j as follows:

$$\mathbf{Q}_q = \sum_j \begin{pmatrix} \mathbf{P}_{q,j} \\ \mathbf{1} \end{pmatrix} \begin{pmatrix} \mathbf{P}_{q,j} \\ \mathbf{1} \end{pmatrix}^T, \quad (4)$$

$$\mathbf{V}_q = \sum_j \mathbf{Y}_{q,j} \begin{pmatrix} \mathbf{P}_{q,j} \\ \mathbf{1} \end{pmatrix}^T, \quad (5)$$

and initialize $\mathbf{P}_{q,j}$ and $\mathbf{Y}_{q,j}$ to avoid memory constraint challenges [17]. Finally, we compute the least-squares solution of the filters $\mathbf{h}_q = \arg \min_{\mathbf{h}} |\mathbf{Q}_q - \mathbf{h}\mathbf{V}_q|^2$ directly using an efficient conjugate gradient method for a positive-definite matrix.

In the testing stage, the k -th HR voxel $\hat{\mathbf{x}}_k$ is estimated by computing a tensor with a patch centered on the voxel, predicting a label, and applying a corresponding trained filter to the LR patch $\mathbf{p}_k \in \mathbb{R}^{m \times 1}$ as follows:

$$\hat{\mathbf{x}}_k = \mathbf{h}_q \left(T \begin{pmatrix} \mathbf{p}_k, \Phi_k \\ \mathbf{1} \end{pmatrix} \right), \quad (6)$$

where $T(\mathbf{p}, \Phi)$ is the operator that transforms the patch \mathbf{p} using an affine transformation Φ . The estimated image $\hat{\mathbf{x}}$ is then constructed by simply estimating each voxel and combining them.

B. TENSOR ANALYSIS FOR LABEL CREATION AND SELECTION

Tensors are generally used to obtain quantitative features such as stress tensors in mechanics and the Hessian matrix or structure tensor in image analysis [39], [41]. A tensor can be independently decomposed into shape (structure) and orientation via eigen-decomposition. In addition, the tensor shape can be decomposed into isotropic and anisotropic components expressed as eigenvalues or tensor invariants. We approached the SR problem using tensor analysis to define a tensor that represented the gradient features of a 3D image patch, decomposed it into the tensor shape and the orientation features, and classified the features by clustering them.

To create labels to classify the patches, prior to the training stage, we classified a large set of tensor features that were randomly sampled patches ranging from training volumes to K types. We extracted shape and orientation features based on the tensors constructed from the patches and classified them, considering that the tensor shape and orientation are independent. We used centroid-based clustering to group features because each feature difference is indicative of the degree of similarity and the tensor shape and orientation are not independent [47]. Specifically, K-means clustering was used to change the distance metric for each shape and orientation to cluster the tensor shape and orientation. In the training and testing stages, each patch was labeled as $q = (c_{shape}, c_{ori})$, where c_{shape} and c_{ori} were the predicted cluster indexes of the tensor shape and the orientation cluster model, respectively.

To create the clustering models, we first defined the quantitative feature (e.g., gradient angle or strength) of the LR image as a tensor. Considering that the interpolated LR image mostly excluded the high frequencies of the HR image, we focused on the edge region where the voxel value changes rapidly, to use these statistics as a geometry measure. The image-gradient tensor $\mathbf{D} \in \mathbb{R}^{3 \times 3}$ was computed as follows:

$$\mathbf{D} = \mathbf{G}^T \mathbf{W} \mathbf{G}, \quad (7)$$

where $\mathbf{G} \in \mathbb{R}^{n \times 3}$ is a gradient matrix for which the column vectors are gradients along each axis for a 3D patch volume, and $\mathbf{W} \in \mathbb{R}^{n \times n}$ is a diagonal Gaussian weighting matrix that assigns a large weight to the gradient around the center of the patch [17], [48]. The second-order tensor \mathbf{D} is a symmetric positive-definite matrix that has six degrees of freedom. Each diagonal component represents the coefficient of each axis, and the other components represent the relationship between the different axes. A tensor can be typically eigen-decomposed into three orthogonal eigenvectors $\mathbf{R} = [\mathbf{e}_1 \ \mathbf{e}_2 \ \mathbf{e}_3]$ and the corresponding non-negative eigenvalues $\Lambda = [\lambda_1^2 \ \lambda_2^2 \ \lambda_3^2]$, wherein the eigenvalues are sorted in the descending order (i.e., $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$). The eigenvalues correspond to the square root of each element of the scaling matrix Λ because the tensor \mathbf{D} is computed by multiplying the gradient matrix \mathbf{G} twice. We represent the decomposed Λ , \mathbf{R} as the independent tensor shape and orientation, respectively, to build cluster models to create labels based on the distance $\mathbf{R} = [\mathbf{e}_1 \ \mathbf{e}_2 \ \mathbf{e}_3]$ and the corresponding non-negative eigenvalues $\Lambda = [\lambda_1^2 \ \lambda_2^2 \ \lambda_3^2]$. Each eigenvalue of the tensor represents the coefficients of the corresponding eigenvector. However, each eigenvalue fails to extract shape components such as the strength or cohesion of the patch gradient. Instead, tensor invariants computed as simple arithmetic combinations of the three eigenvalues are preferred to represent these features. Given that tensor shape invariants reflect local features, we selected trace, fractional anisotropy (FA), and a mode defined as

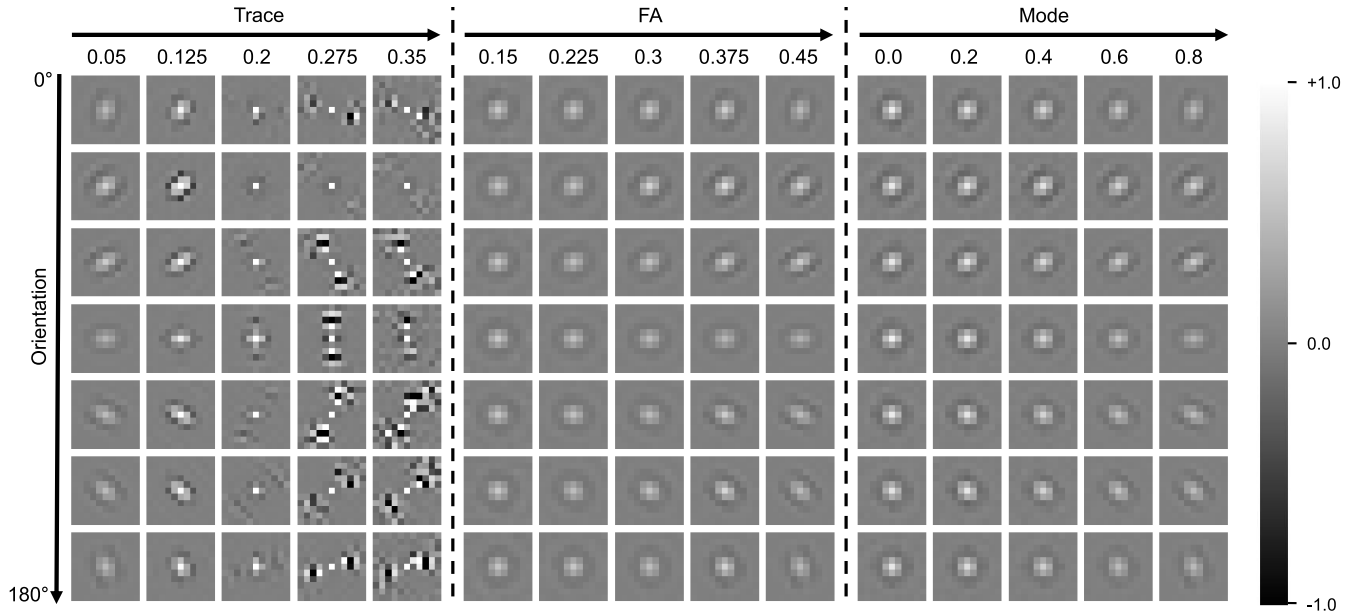


FIGURE 2. Visualization of the learned $2 \times$ filters according to the tensor shape and orientation. Each column represents a shape feature based on $J = (0.05, 0.45, 0.8)$ whereby each invariant is replaced by the number indicated in the figure. For the orientation of each row, the centroids were sampled at uniform angles between \mathbf{e}_{11} and \mathbf{e}_{13} with $\mathbf{e}_{12} = 0$.

follows:

$$\text{trace}(\mathbf{D}) = \lambda_1 + \lambda_2 + \lambda_3, \quad (8)$$

$$\text{FA}(\mathbf{D}) = \frac{1}{\sqrt{2}} \frac{\sqrt{(\lambda_1 - \bar{\lambda})^2 + (\lambda_2 - \bar{\lambda})^2 + (\lambda_3 - \bar{\lambda})^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}, \quad (9)$$

$$\text{mode}(\mathbf{D}) = \frac{27(\lambda_1 - \bar{\lambda})(\lambda_2 - \bar{\lambda})(\lambda_3 - \bar{\lambda})}{2(\lambda_1^2 + \lambda_2^2 + \lambda_3^2 - \lambda_1\lambda_2 - \lambda_1\lambda_3 - \lambda_2\lambda_3)^{3/2}} \quad (10)$$

where $\bar{\lambda} = (\lambda_1 + \lambda_2 + \lambda_3) / 3$. The range of trace, FA, and the modes are $[0, \infty)$, $[0, 1]$, and $[-1, 1]$, respectively. Trace was considered as the strength and the other two as the spread. These three are also used in diffusion tensor MRI [47]. Next, we extracted tensor orientation features from the rotation matrix \mathbf{R} , which consisted of three eigenvectors and represented the rotation from the principal frame. We considered the orientation feature as only the primary eigenvector $\mathbf{e}_1 = [e_{11} \ e_{12} \ e_{13}]^T$ to focus on the major gradient of the patch, wherein $e_{11} \geq 0$ was used to resolve the sign ambiguity that \mathbf{e}_1 and $-\mathbf{e}_1$ represented the same direction. Then, we carefully defined a tensor shape and orientation distance between the tensors \mathbf{A} and \mathbf{B} as follows:

$$d_{\text{shape}}(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^3 |J_i(\mathbf{A}) - J_i(\mathbf{B})| / \sigma_i, \quad (11)$$

$$d_{\text{ori}}(\mathbf{A}, \mathbf{B}) = 1 - \mathbf{e}_{A1} \cdot \mathbf{e}_{B1}, \quad (12)$$

where $J = (\text{trace}, \text{FA}, \text{mode})$, σ_i is the standard deviation of each invariant, and \mathbf{e}_{A1} and \mathbf{e}_{B1} are the primary vectors of \mathbf{A} and \mathbf{B} , respectively. In the case of shape distance, we used

each invariant weight equally by normalizing dividing the standard deviation and adding the Manhattan distance (i.e., L1 distance). We used the cosine distance to compare the angle between two unit vectors in the orientation distance. To show the effect of filters according to different tensor shapes and orientations, we illustrate the learned $2 \times$ wide variety of filters using our clustering model wherein the orientation features are predefined to be evenly distributed as shown in Fig. 2. For each column, it is evident that the direction of the filter reflects the primary eigenvector. The trace indicates the relative intensity of the directional structure and reflects the shape and strength of the filter. The FA indicates the directionality of the patch, such that the filter with the higher FA is more directional to the patch orientation. The mode represents the anisotropy associated with the gradient. Therefore, the anisotropic shape of the filter changes as the mode changes (e.g., orthotropic or linear anisotropic). Note that the shape centroids are manually chosen for visualization. As such, the tensor shape and orientation features are well-defined to generate a variety of 3D filters for SR of MR volumes.

C. TENSOR ORIENTATION TRANSFORMATION FOR PATCH SPAN REDUCTION

In the example-based method, a model has a trade-off between the computational time and the accuracy for the exemplar parameter K (e.g., dictionary size, several centroids). Therefore, a moderately large K ($2^{10} \sim 2^{11}$) was used for the 2D SR [14], [15], [18]. However, since the patch dimension increases when expanded to 3D space in this problem, it is more difficult to learn fine features using K for

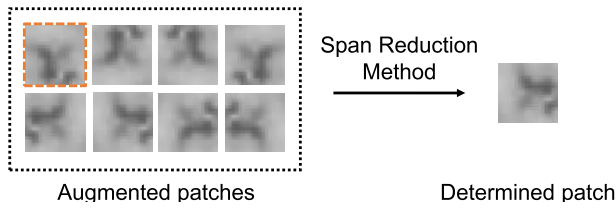


FIGURE 3. Example of the proposed patch span reduction method. An original patch (highlighted among the augmented patches) can be augmented using multiple patches by flipping or rotating. The proposed span reduction method determines these patches to only one patch for training or inference.

2D. In addition, a higher K can increase the computational time or results in memory shortage problems. To maintain K , it is necessary to learn fine features, which is achieved by reducing the patch span. We propose a patch span reduction method based on the transformation of the tensor orientation. This approach limits the orientation of the patch as shown in Fig. 3. The transformation of the patch was determined by directly associating the tensor orientation with the patch, thereby reducing the span of the tensor orientation.

The original span of the primary eigenvector e_1 satisfies $e_{11}^2 + e_{12}^2 + e_{13}^2 = 1, e_{11} \geq 0$. To reduce the span by constraining e_1 , we compute the transformation Φ that collapses the variations of e_1 into e_1^* :

$$e_1^* = \Phi e_1 = \begin{bmatrix} e_{11}^* \\ e_{12}^* \\ e_{13}^* \end{bmatrix}, \quad \Phi = \Phi_P \Phi_F \quad (13)$$

where $e_{11}^* \geq e_{12}^* \geq e_{13}^* \geq 0$. Φ is achieved by the composite of the two transformations. The flip transformation Φ_F changes only the sign of each element of the input vector so that it is nonnegative. The permutation matrix Φ_P then sorts each element of the input vector in descending order. For visualization, we show scatter plots of the tensor orientation (primary eigenvector) before and after span reduction in Fig. 4. Compared to the case of a half unit sphere with a normalization of 1 without any transformations, as shown in Fig. 4(a), only the positive space of the unit sphere is taken after flipping as shown in Fig. 4(b). Figure 4(c) illustrates the final reduced span, which shows that the patch density increased as the orientation span decreased. Considering the number of these orientation limitation cases, the flip and permutation matrices reduce the orientation span by 1/4 and 1/6, respectively. As such, if 72 orientational filters are created for each shape feature to cover a half unit sphere, only 3 orientation filters are required for the span reduction method.

To apply the proposed span reduction method to the overall algorithm, the primary vector and patch transformed with Φ were used for orientation clustering, training, and evaluation as follows:

- In orientation clustering, e_1^* was used instead of e_1 . The nearest neighbor of e_1^* was also used for label selection during the training and testing stages.

- In the training stage, we inserted the remapped patch with Φ into \mathbf{P} instead of the original patch.
- In the testing stage, we applied a filter to the LR patch that was remapped using the extracted Φ .

III. EXPERIMENTS

A. DATASETS

We obtained high-resolution T1-weighted MR volumes (brain-masked) from the Human Connectome Project (HCP) 900 dataset, which is publicly accessible [46]. Of the 897 volumes (3 scans were not available), 30 were used for training and the remaining 867 were used for evaluation. The volume size of the dataset was $260 \times 311 \times 260$ with an isotropic spatial resolution of 0.7 mm.

B. IMAGE PREPROCESSING

For the original HR volume, we clipped the maximum intensity using the upper 0.1% value for each volume to remove voxels with high-intensity values that darken the overall image. The data range of the volume was then normalized from 0 to 1. To match the size of the HR image and the interpolated LR image volume, the HR image volume was cropped to the maximum size of a multiple of the factor (2 to 4), and then downsampled using the tricubic kernel to generate the LR image volume. The LR volume was then upsampled using tricubic interpolation and the mask extracted from the HR image was overlapped on it.

C. PARAMETERS

The total number of labels K was set to 1023 (≈ 1024), which is a combination of 3 (reflecting span reduction method) and 341 centroids in the tensor orientation and shape cluster model, respectively. We set the patch volume to 93 to construct the tensor, and 113 for filter calculation and patch collection. The sigma of the Gaussian kernel of \mathbf{W} used for tensor calculation in Eq. (7) was set to 0.85. We experimented with the downscaling/upscaling factors of 2, 3, and 4 using the same HR training images and test sets for quantitative comparison of the overall recovery quality between different SR methods. Other experiments were performed using a factor of 2.

D. IMPLEMENTATION DETAILS

The cluster models were implemented by overriding the scikit-learn source code [49], the centroids were initialized with k-means++, and the patches were clustered using 30 iterations. We used approximately 7.5M patches to build the cluster model, wherein all the patches in the voxel with centers in brain region of the HR volume were used for training. We updated the \mathbf{Q} and \mathbf{V} matrices in Eqs. (4) and (5) 5 times per volume considering the trace-off between memory capacity and computational efficiency. The proposed model was trained within 3 hours without GPU acceleration. The source code is available in GitHub: https://github.com/Snailpong/SR_Tensor.

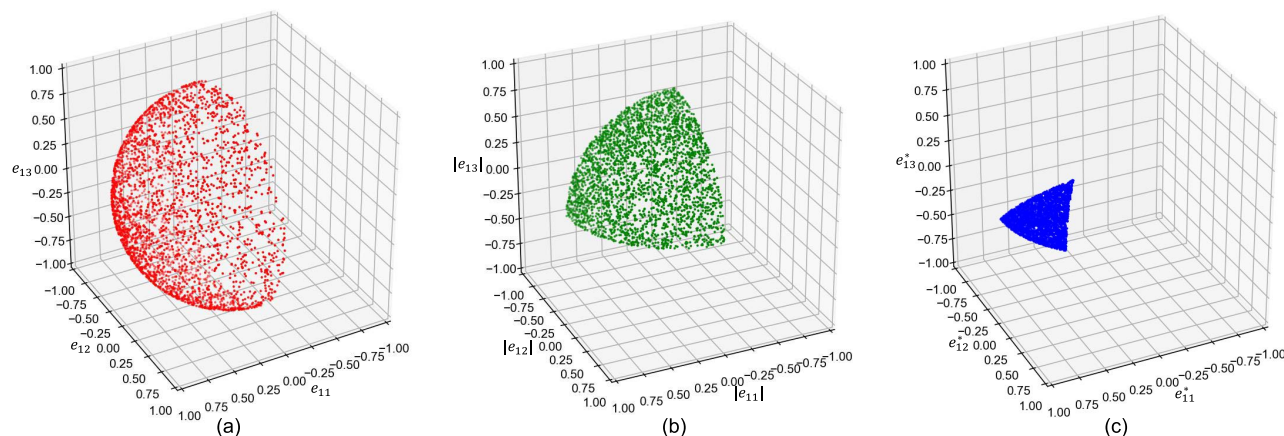


FIGURE 4. Scatter plots for tensor orientation (primary eigenvector) with/without the span reduction method: (a) without transformation, (b) with only flip transformation, and (c) with both transformations. As the span (searching space) decreases, the number of filters required to learn the patch direction decreases. Note the same patches were used for each plot.

E. COMPARISONS

We compared the performance of the proposed method to the conventional nearest neighbor, tricubic interpolation, and the 3D extended SRCNN [23], which is a deep learning model. For the latter, we used a typical 9-1-5 structure model wherein only the 2D convolution operation was replaced in the 3D operation. This was implemented using Tensorflow2 [50] by minimizing the mean square error using the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) for a learning rate of 3×10^{-4} . Convergence was achieved using a batch size of 16 for each 100 K step using 30 training subjects. The input and output size of the 3D SRCNN was $60 \times 60 \times 60$ for the tricubic-interpolated LR and HR patches, similar to previously reported 3D-based studies, owing to insufficient GPU memory [44], [45].

F. OVERALL QUALITY METRICS

To quantitatively measure the overall quality of the recovered volumes using the SR methods, we used two standard metrics: peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM), which consider the reference-based image similarity [51]. Specifically, PSNR directly measures the voxel value difference between two images, and SSIM evaluates image similarity in terms of luminance, contrast, and structure. These metrics are calculated per voxel and are averaged for the entire volume.

G. BOUNDARY QUALITY METRICS

For the evaluation of SR performance in the boundary areas, especially around the cortex, we used surface models to effectively represent the complicated cortical folding. The same procedures of reconstruction based on the original and recovered HR T1w volumes were performed using FreeSurfer7 to generate triangular meshes (lh.pial and lh.white) and parcellation (lh.aparc.annot) of the left gray matter and white matter [52]. We first compared the surfaces

generated from the original and recovered HR volumes because the cerebral cortex and gray-white matter boundaries represented by the surface models significantly affected surface atlas registration and gyral labeling. To facilitate a quantitative comparison between the surface and the originals, we calculated the mesh-to-mesh distance between the original and recovered surfaces. We projected each vertex of the reconstructed surface onto the nearest point of the original surface, measured the distance for every vertex, and averaged all the distances. Moreover, we evaluated the recovery performance of gyral labeling by measuring the overlap of the gyral labels (34 labels selected from FreeSurfer's surface parcellation) generated from the original HR images and recovered volumes. The recovered surface was projected onto the original HR surface using the nearest point map to establish their one-to-one correspondences. For each of the 34 cortical regions, we then computed the dice coefficients (DCs) between the original and recovered labeled regions for the 867 test sets. Student's t-tests were then performed to statistically compare the DCs for each small cortical region generated from the recovered volumes using the proposed algorithm and SRCNN, and the number of labels with statistically significant DC differences was determined.

H. ENVIRONMENT

Training and inference of the proposed algorithm were performed using a CPU (Intel Xeon W-2265 3.50 GHz) with 64 GB RAM without a GPU. The NVIDIA GeForce RTX 2080 TI GPU was also used to train the SRCNN model.

IV. RESULTS

A. ABLATION STUDY

1) PERFORMANCE IMPACT OF THE NUMBER OF TRAINED SUBJECTS

We quantitatively investigated the impact on performance depending on the number of trained subjects (T) in the

TABLE 1. PSNR [dB] and SSIM performances (higher values are better) of the proposed algorithm compared to the nearest neighbor, tricubic, and SRCNN algorithms for factors of 2×, 3×, and 4×. Each result is represented with the mean and standard deviation of the PSNR and SSIM values for the 867 test subjects.

Scale	Metric	Nearest Neighbor	Tricubic	SRCNN	Proposed
2×	PSNR	30.32±0.52	32.17±0.58	35.80±0.84	35.97±0.88
	SSIM	0.9633±0.0049	0.9664±0.0054	0.9833±0.0039	0.9827±0.0041
3×	PSNR	28.11±0.46	29.96±0.52	32.92±0.63	32.63±0.62
	SSIM	0.9391±0.0069	0.9474±0.0068	0.9686±0.0055	0.9671±0.0056
4×	PSNR	26.46±0.49	28.37±0.58	30.85±0.53	30.64±0.53
	SSIM	0.9171±0.0089	0.9289±0.0082	0.9533±0.0063	0.9519±0.0064

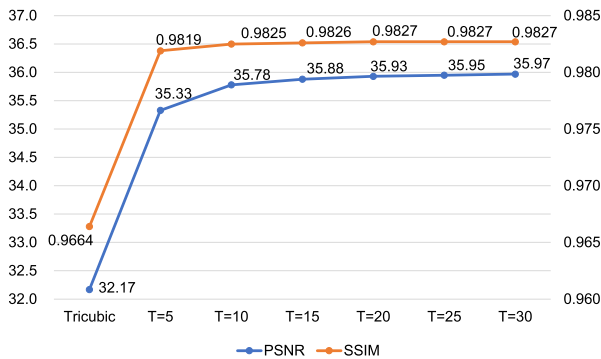


FIGURE 5. Quantitative comparison between the PSNR [dB] and SSIM performance vs. the number of trained subjects (T). Each result for each T is represented using the mean of the 867 test subjects. Based on the T for which each graph is flattened, the test result is rarely affected.

proposed algorithm by changing only the number of trained subjects. Note that as T increases during training, the trained subjects were added to the training dataset. Figure 5 shows the PSNR and SSIM performance according to the number of trained subjects. It was determined that 30 trained subjects were sufficient to train the proposed algorithm, resulting in a small gain in the PSNR of 0.04 dB and no gain in the SSIM compared to the case of 20 trained subjects.

2) PATCH SPAN REDUCTION EFFECT

To show the improvements achieved using the patch span reduction method presented in Section 2.3, the performance of the proposed method was quantitatively compared with and without span reduction. The original primary eigenvectors and patches without transformation were used for orientation clustering and training, and the number of centroids for each tensor orientation and shape cluster model was set to 32 to train the proposed algorithm without using the span reduction method. For an upscaling factor of 2, the PSNR was 35.82 ± 0.86 , and the SSIM was 0.9827 ± 0.0041 . It was observed that the span reduction method improved the PSNR by 0.15 dB, compared to the case when the method was not used.

B. COMPARISONS

1) THE OVERALL QUALITY OF VOLUMES

First, we evaluated the performance of the proposed algorithm and the conventional algorithms in terms of the

prediction of the volume intensities. Figure 6 shows one of the 2× test results from the original HR volume ((a) subject id: 996782) using the tricubic (b), SRCNN (c), and the proposed algorithm (d), where axial (top-left), coronal (top-right), and sagittal (bottom) slices at the same location are shown in each subfigure. It is evident that the tricubic algorithm failed to recover the overall volume intensities, especially around the gray-white matter boundary and the sulcus region. However, the proposed method and SRCNN were better at recovering the HR structural details, wherein these regions are highlighted in red and blue dotted boxes. Table 1 shows the quantitative PSNR and SSIM performance of the proposed algorithm, interpolation, and SRCNN using the 867 test subjects with 2×, 3×, and 4× upscaling factors. It is evident that the proposed method and SRCNN significantly outperformed the interpolation methods for all the upscaling factors. For the factors of 2, 3, and 4, the proposed method significantly improved the PSNR by 3.80 dB (11.81%), 2.67 dB (8.91%), and 2.27 dB (8.00%), respectively, compared to tricubic; and the difference in the PSNR between the proposed approach and SRCNN is much smaller (less than 0.9%). For SSIM, the proposed method achieved 0.0163 (1.69%), 0.0197 (2.08%), and 0.0230 (2.48%) higher values than the tricubic algorithm for 2×, 3×, and 4×, respectively. However, the SSIM difference between the proposed algorithm and SRCNN was negligible (less than or equal to 0.15%).

2) BOUNDARY RECOVERY

Next, we evaluated the recovery performance of the proposed method in the cortical boundary regions by measuring the distance from the recovered cortical surface to the original HR-based surface. Figure 7 represents a subject’s (subject id: 849971) pial (left) and white matter (right) surface sections that were reconstructed using tricubic (a), SRCNN (b), and the proposed algorithm (c) with yellow border overlapped with the original surface with a red border, showing the sagittal (left), coronal (center), and axial (right) view of each surface. It is evident that the tricubic algorithm failed to recover the boundary region, significantly mismatching both the original pial and white matter surfaces, especially for the frontal and parietal lobes shown in the sagittal view. By contrast, the surfaces obtained for the SRCNN and the proposed algorithm exhibited well-recovered boundaries, but the proposed algorithm was slightly better than SRCNN

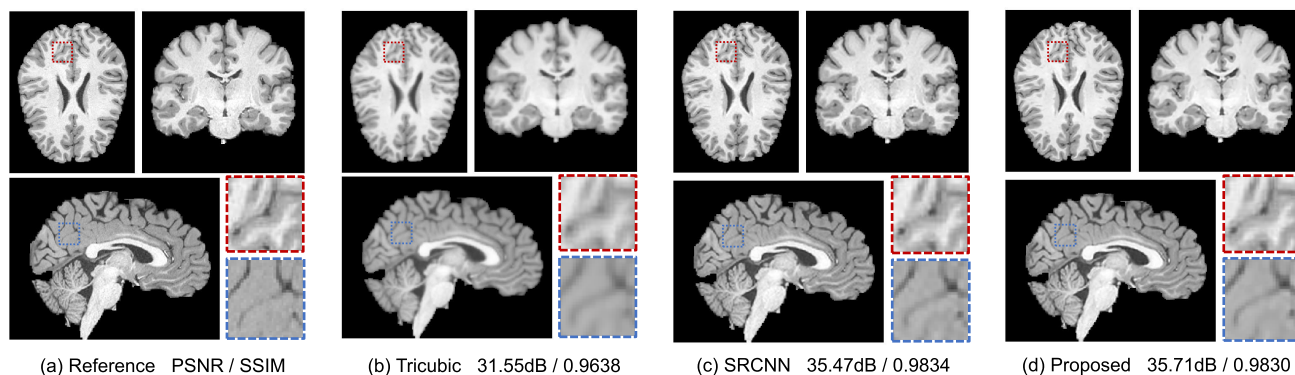


FIGURE 6. Visual comparison between (a) HR volume and the reconstructed volumes using (b) tricubic, (c) SRCNN, and (d) the proposed algorithm for 2x. For each volume, we present the axial (top-left), coronal (top-right), and sagittal (bottom-left) views, and the highlighted boundary regions (red and blue dotted boxes at the bottom-right). The PSNR and SSIM of the corresponding subject are shown at the bottom.

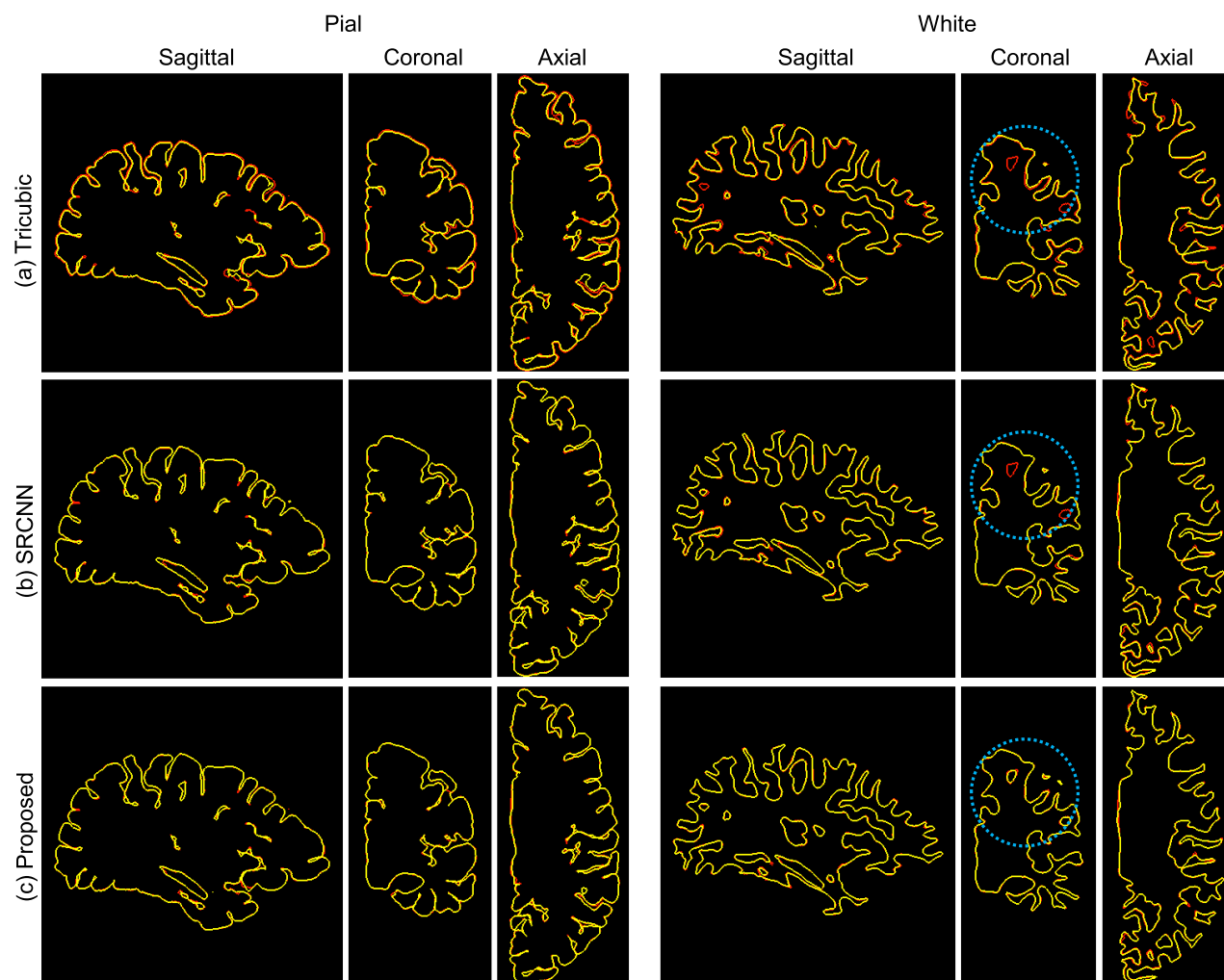


FIGURE 7. Visual comparison of surface reconstruction between the SR algorithms. Sagittal, coronal, and axial sections of pial (left) and white matter (right) surfaces are shown. The section that was reconstructed using each method is represented by a yellow line. To compare the quality of the boundary to the original surface, the original section with the red line was overlapped for each section. Regions with significant mismatch from the original are highlighted using a blue dotted circle.

for the regions on the white surface highlighted by blue dotted circles. Table 2 shows the quantitative mesh-to-mesh

measurement results of the proposed algorithm, SRCNN, and tricubic interpolation for pial and white surfaces using the

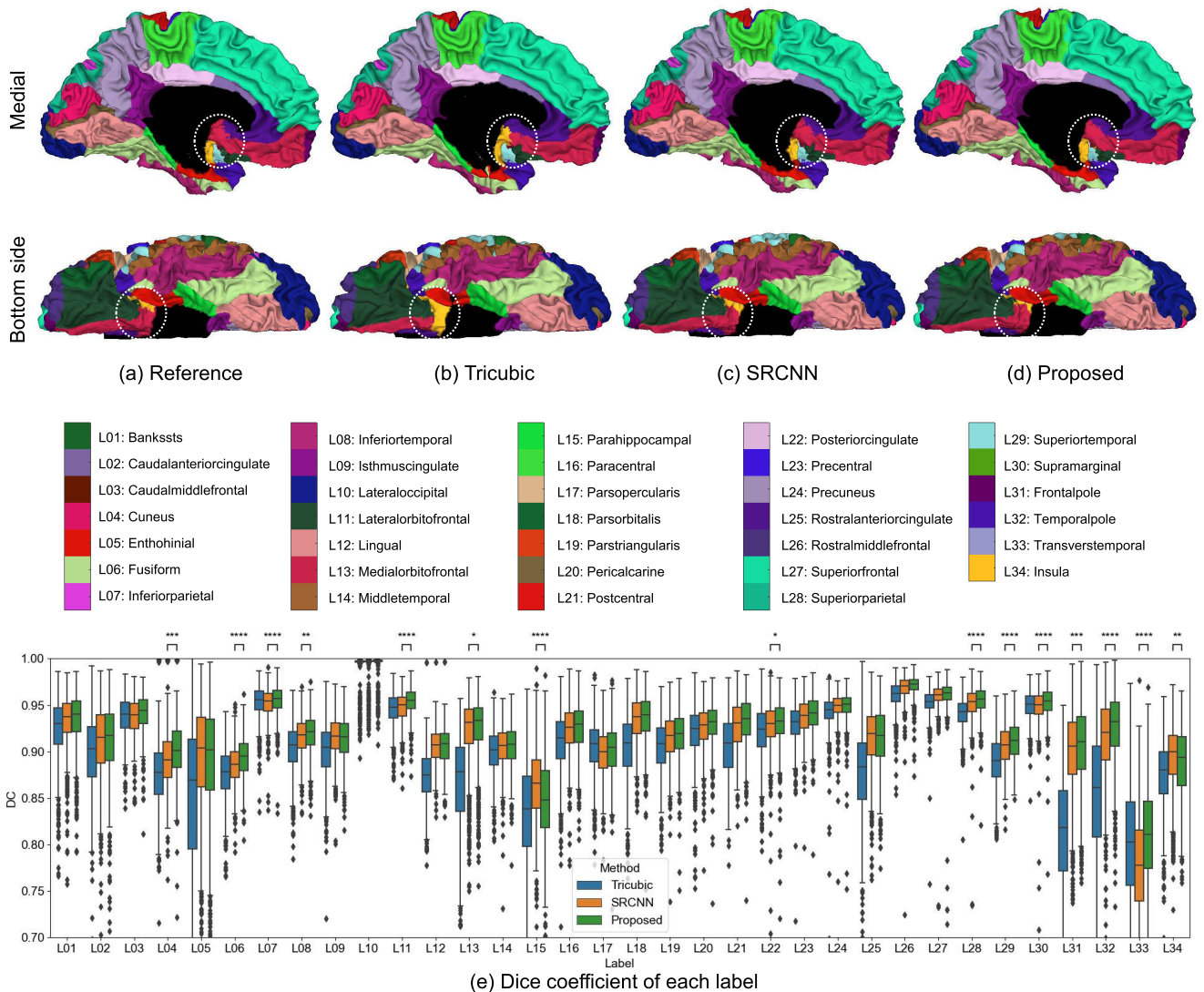


FIGURE 8. Validation of the SR algorithms for recovery of cortical parcellation using the large-scale dataset (867 subjects). White matter surfaces with parcellation of 34 cortical regions for the left hemisphere of a subject are shown with the medial and bottom side views. The color bar below shows the list of 34 gyral labels produced using Freesurfer. (e) Box plots of the dice coefficients (DC), which measure the overlap of labels between the truth and the reconstructed surface for each region. The asterisks on each plot indicate a statistically significant difference between the proposed method and SRCNN. 1 to 4 asterisks are used to indicate p-value less than 0.05, 0.01, 0.001, and 0.0001, respectively.

TABLE 2. Mesh-to-mesh distance [mm] (smaller is better) between original and reconstructed pial and white matter surfaces for each method. Each result is represented as the mean and standard deviation of the test subjects.

Surface	Tricubic	SRCNN	Proposed
Pial	0.2571±0.0229	0.1419±0.0123	0.1302±0.0132
White matter	0.2855±0.0275	0.1516±0.0148	0.1327±0.0159

867 test subjects with an upscaling factor of 2x. The results show that the proposed method was superior to the tricubic and SRCNN approaches in terms of the distance on the pial surface by 0.1269 mm (49.4%) and 0.0117 mm (8.2%), respectively, and by 0.1528 mm (53.5%) and 0.0189 mm (12.5%) on the white surface.

3) CORTICAL PARCELLATION RECOVERY

Finally, we compared the prediction of the cortical surface mapping of the proposed algorithm and the conventional algorithms. Figures 8(a) to (d) show the reconstructed white matter surfaces with predicted cortical parcellation for one subject (subject id: 162228) using the HR surface (a), tricubic (b), SRCNN (c), and the proposed algorithm with 2x upscaling factor, for the medial (top) and bottom side (bottom). It is evident that the tricubic algorithm failed to distinguish between complex areas, such as the medial-orbitofrontal (L13) and the insula region (L34), highlighted using the dashed white circle. By contrast, the proposed algorithm and SRCNN were better at recovering the annotation without any significant mislabeling of the

regions. Figure. 8(e) shows the quantitative DC results of the proposed algorithm, SRCNN, and tricubic interpolation for the 867 test subjects in 34 regions with a factor of $2\times$ and a box plot for visualization. In addition, the proposed method and SRCNN were compared using unpaired Student's t-test for each region. The results are shown above the box plot. Among the 34 cortical areas, 15 areas exhibited a statistically significant difference (p -value < 0.05), and they are identified with asterisks. Among these 15 regions, the proposed algorithm achieved significantly better results than the SRCNN algorithm in 13 regions. This difference originated from the volume images, and our method performed well on the boundaries, which is important for surface composition analysis.

V. CONCLUSION

In this paper, we developed a cost-efficient novel fully 3D super-resolution method that targets the boundaries in MRI data by introducing a tensor using 3D gradient information. Cluster models were developed to create labels using the tensor shape and the orientation features obtained from the tensor, and filters were trained by collecting similar patches for each label. We propose a patch span reduction method by collapsing the variations of the tensor orientation to learn fine patch features. Testing on MR volumes showed that the proposed method achieved comparable recovery of HR details compared to conventional methods, effectively reducing the training time and the number of trained subjects required. In addition, the results for the boundary recovery revealed that the proposed algorithm significantly outperformed conventional methods in the cerebral cortex. In practical medical applications, it facilitates successive surface-based analyses that would otherwise yield inaccurate results with low-quality MRI data. The method considers only the primary orientation of the patch, which accurately recovers the boundaries, but the improvement may be small in complex parts such as internal textures. In future work, to solve the problem, we will focus on considering all eigenvectors including principal eigenvector as features. Also, we will perform experiments with other datasets and extend this technique to different brain imaging modalities.

REFERENCES

- [1] D. Peressutti, M. Sinclair, W. Bai, T. Jackson, J. Ruijsink, D. Nordsletten, L. Asner, M. Hadjicharalambous, C. A. Rinaldi, D. Rueckert, and A. P. King, "A framework for combining a motion atlas with non-motion information to learn clinically useful biomarkers: Application to cardiac resynchronisation therapy response prediction," *Med. Image Anal.*, vol. 35, pp. 669–684, Jan. 2017.
- [2] J. K. Gahm and Y. Shi, "Riemannian metric optimization on surfaces (RMOS) for intrinsic brain mapping in the Laplace–Beltrami embedding space," *Med. Image Anal.*, vol. 46, pp. 189–201, May 2018.
- [3] J. C. Pruessner, "Volumetry of hippocampus and amygdala with high-resolution MRI and three-dimensional analysis software: Minimizing the discrepancies between laboratories," *Cerebral Cortex*, vol. 10, no. 4, pp. 433–442, Apr. 2000.
- [4] F. Shi, J. Cheng, L. Wang, P.-T. Yap, and D. Shen, "LRTV: MR image super-resolution with low-rank and total variation regularizations," *IEEE Trans. Med. Imag.*, vol. 34, no. 12, pp. 2459–2466, Dec. 2015.
- [5] E. Plenge, D. H. Poot, M. Bernsen, G. Kotek, G. Houston, P. Wielopolski, L. van der Weerd, W. J. Niessen, and E. Meijering, "Super-resolution methods in MRI: Can they improve the trade-off between resolution, signal-to-noise ratio, and acquisition time?" *Magn. Reson. Med.*, vol. 68, no. 6, pp. 1983–1993, 2012.
- [6] C. E. Duchon, "Lanczos filtering in one and two dimensions," *J. Appl. Meteorol.*, vol. 18, no. 8, pp. 1016–1022, 1979.
- [7] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.
- [8] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP, Graph. Models Image Process.*, vol. 53, no. 3, pp. 231–239, May 1991.
- [9] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 25–47, 2000.
- [10] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [11] X. Lu, Z. Huang, and Y. Yuan, "MR image super-resolution via manifold regularized sparse learning," *Neurocomputing*, vol. 162, pp. 96–104, Aug. 2015.
- [12] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surf.* Avignon, France: Springer, 2010, pp. 711–730.
- [13] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
- [14] R. Timofte, V. De, and L. V. Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1920–1927.
- [15] R. Timofte, V. D. Smet, and L. V. Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. Asian Conf. Comput. Vis.* Singapore: Springer, 2014, pp. 111–126.
- [16] C.-Y. Yang and M.-H. Yang, "Fast direct super-resolution by simple functions," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 561–568.
- [17] Y. Romano, J. Isidoro, and P. Milanfar, "RAISR: Rapid and accurate image super resolution," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 110–125, Mar. 2017.
- [18] S. He and B. Jalali, "Fast super-resolution in MRI images using phase stretch transform, anchored point regression and zero-data learning," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2876–2880.
- [19] S. Schuler, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3791–3799.
- [20] J. Salvador and E. Perez-Pellitero, "Naive Bayes super-resolution forest," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 325–333.
- [21] E. Perez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn, "PSyCo: Manifold span reduction for super resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1837–1845.
- [22] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 349–356.
- [23] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.* Zürich, Switzerland: Springer, 2014, pp. 184–199.
- [24] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 391–407.
- [25] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1874–1883.
- [26] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 624–632.
- [27] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1646–1654.
- [28] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 136–144.
- [29] J. Shi, Z. Li, S. Ying, C. Wang, Q. Liu, Q. Zhang, and P. Yan, "MR image super-resolution via wide residual networks with fixed skip connection," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 3, pp. 1129–1140, May 2019.

- [30] C.-M. Feng, K. Wang, S. Lu, Y. Xu, and X. Li, "Brain MRI super-resolution using coupled-projection residual network," *Neurocomputing*, vol. 456, pp. 190–199, Oct. 2021.
- [31] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, "DenseNet: Implementing efficient ConvNet descriptor pyramids," 2014, *arXiv:1404.1869*.
- [32] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4799–4807.
- [33] D. Liu, B. Wen, Y. Fan, C. Change Loy, and T. S. Huang, "Non-local recurrent network for image restoration," 2018, *arXiv:1806.02919*.
- [34] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," 2019, *arXiv:1903.10082*.
- [35] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 286–301.
- [36] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4681–4690.
- [37] O. Oktay, W. Bai, M. Lee, R. Guerrero, K. Kamnitsas, J. Caballero, A. de Marvao, S. Cook, D. O'Regan, and D. Rueckert, "Multi-input cardiac image super-resolution using convolutional neural networks," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Athens, Greece: Springer, 2016, pp. 246–254.
- [38] S. Wang, Z. Su, L. Ying, X. Peng, S. Zhu, F. Liang, D. Feng, and D. Liang, "Accelerating magnetic resonance imaging via deep learning," in *Proc. IEEE 13th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2016, pp. 514–517.
- [39] C. Sazak and B. Obara, "Contrast-independent curvilinear structure enhancement in 3D biomedical images," in *Proc. IEEE 14th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2017, pp. 1165–1168.
- [40] C. Wang, M. Oda, Y. Hayashi, Y. Yoshino, T. Yamamoto, A. F. Frangi, and K. Mori, "Tensor-cut: A tensor-based graph-cut blood vessel segmentation method and its application to renal artery segmentation," *Med. Image Anal.*, vol. 60, Feb. 2020, Art. no. 101623.
- [41] J. Jin, L. Yang, X. Zhang, and M. Ding, "Vascular tree segmentation in medical images using hessian-based multiscale filtering and level set method," *Comput. Math. Methods Med.*, vol. 2013, pp. 1–9, Oct. 2013.
- [42] C.-H. Pham, C. Tor-Díez, H. Meunier, N. Bednarek, R. Fablet, N. Passat, and F. Rousseau, "Multiscale brain MRI super-resolution using deep 3D convolutional networks," *Computerized Med. Imag. Graph.*, vol. 77, Oct. 2019, Art. no. 101647.
- [43] Y. Chen, Y. Xie, Z. Zhou, F. Shi, A. G. Christodoulou, and D. Li, "Brain MRI super resolution using 3D deep densely connected neural networks," in *Proc. IEEE 15th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2018, pp. 739–742.
- [44] Y. Chen, A. G. Christodoulou, Z. Zhou, F. Shi, Y. Xie, and D. Li, "MRI super-resolution with GAN and 3D multi-level DenseNet: Smaller, faster, and better," 2020, *arXiv:2003.01217*.
- [45] J. Wang, Y. Chen, Y. Wu, J. Shi, and J. Gee, "Enhanced generative adversarial network for 3D brain MRI super-resolution," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 3627–3636.
- [46] D. C. Van Essen, D. C. V. Essen, S. M. Smith, D. M. Barch, T. E. Behrens, E. Yacoub, and K. Ugurbil, "The WU-minn human connectome project: An overview," *NeuroImage*, vol. 80, pp. 62–79, Oct. 2013.
- [47] J. K. Gahm, G. Kindlmann, and D. B. Ennis, "The effects of noise over the complete space of diffusion tensor shape," *Med. Image Anal.*, vol. 18, no. 1, pp. 197–210, Jan. 2014.
- [48] P. Getreuer, I. Garcia-Dorado, J. Isidoro, S. Choi, F. Ong, and P. Milanfar, "BLADE: Filter learning for general purpose computational photography," in *Proc. IEEE Int. Conf. Comput. Photography (ICCP)*, May 2018, pp. 1–11.
- [49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
- [50] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Design Implement.*, 2016, pp. 265–283.
- [51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [52] A. M. Dale, B. Fischl, and M. I. Sereno, "Cortical surface-based analysis: I. Segmentation and surface reconstruction," *NeuroImage*, vol. 9, no. 2, pp. 179–194, Feb. 1999.



SEONGSU PARK received the B.S. degree in computer science and engineering from Pusan National University, Republic of Korea, in 2020, where he is currently pursuing the M.S. degree in artificial intelligence. His research interests include computer vision, medical image analysis and image synthetic, generative networks, and animating vision.



JIN KYU GAHM received the B.S. degree in computer engineering from Seoul National University, Republic of Korea, in 2000, and the Ph.D. degree in computer science from the University of California at Los Angeles, Los Angeles, USA, in 2014. He was a Postdoctoral Associate with the Laboratory of Neuro Imaging, University of Southern California, USA, from 2014 to 2019. He is currently an Assistant Professor with the School of Computer Science and Engineering, Pusan National University. His research interests include medical image processing, machine learning, and brain shape analysis.

• • •