

## Research Article

# A Sparse Volume Reconstruction Method for Fetal Brain MRI Using Adaptive Kernel Regression

Qian Ni,<sup>1</sup> Yi Zhang,<sup>2</sup> Tiexiang Wen ,<sup>3,4</sup> and Ling Li<sup>5</sup>

<sup>1</sup>Shenzhen Hospital of Guangzhou University of Chinese Medicine, Shenzhen, China

<sup>2</sup>Radiology & Vascular Surgery, Department of Radiology, Zhongda Hospital, Medical School, Southeast University, Nanjing, China

<sup>3</sup>Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

<sup>4</sup>National Innovation Center for Advanced Medical Devices, Shenzhen, China

<sup>5</sup>Suzhou Institute of Advanced Technology, Chinese Academy of Sciences, Suzhou, China

Correspondence should be addressed to Tiexiang Wen; tx.wen@siat.ac.cn

Received 9 November 2020; Revised 25 January 2021; Accepted 18 February 2021; Published 8 March 2021

Academic Editor: Erlei Zhang

Copyright © 2021 Qian Ni et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Slice-to-volume reconstruction (SVR) method can deal well with motion artifacts and provide high-quality 3D image data for fetal brain MRI. However, the problem of sparse sampling is not well addressed in the SVR method. In this paper, we mainly focus on the sparse volume reconstruction of fetal brain MRI from multiple stacks corrupted with motion artifacts. Based on the SVR framework, our approach includes the slice-to-volume 2D/3D registration, the point spread function- (PSF-) based volume update, and the adaptive kernel regression-based volume update. The adaptive kernel regression can deal well with the sparse sampling data and enhance the detailed preservation by capturing the local structure through covariance matrix. Experimental results performed on clinical data show that kernel regression results in statistical improvement of image quality for sparse sampling data with the parameter setting of the structure sensitivity 0.4, the steering kernel size of  $7 \times 7 \times 7$  and steering smoothing bandwidth of 0.5. The computational performance of the proposed GPU-based method can be over 90 times faster than that on CPU.

## 1. Introduction

Magnetic resonance imaging (MRI) is an ideal diagnostic technique for researchers to investigate the development of the fetal brain [1]. Its advantages are the absence of ionizing radiation, the availability of different contrast options (T1-weighted, T2-weighted, and diffusion-weighted imaging), and the superior contrast of soft tissue compared with ultrasonography, and MRI is also a safe and noninvasive procedure for patients and fetuses [2–4]. For these reasons, MRI has been widely used to investigate the developing fetal brain in vivo [5]. For fetal brain MRI, the high-quality volume representation of 3D acquisition has significant clinical meaning [6]. By the observation of the reconstructed volume data, researchers can study the mechanism of brain development and maturation [7] and identify the fetal brain abnormality or potential injury [8, 9], such as brain tumors, vascular malformations, and

posterior fossa abnormalities. Fetal brain MRI can provide abundant information about aid clinical management, prognostication, and counseling [10].

The duration of an examination is typically 45 to 60 minutes for fetal brain MRI [1]. One major problem of fetal brain MRI is motion artifacts caused by fetal and maternal motion, because of the long acquisition times of 3D MRI scanning. Maternal motion may be avoided by some measures, but fetal motion is usually fast and unpredictable, especially for the younger fetus. Thus, it is still challenging to reconstruct high-fidelity image for fetal brain MRI due to the presence of fetal motion. For fetal motion, different strategies can be adopted to reduce the motion artifacts on MRI [11]. The first strategy tries to prevent the motion occurring during the examination, such as maternal sedation. The second one tries to quicken the data sampling speed. The faster the acquisition techniques for fetal brain MRI are, the lower the motion occurs. For example, the single-shot fast spin

echo (SSFSE) T2-weighted imaging can acquire a slice at 1-second speed [12]. On the other hand, sparse data sampling technique can be applied to shorten the time of data acquisition. The last strategy tries to reconstruct high-quality image through advanced postprocessing motion detection and correction algorithms, such as the SVR method [13].

For the SVR framework [14], it includes the following steps to reduce the fetal motion and reconstruct the high-quality 3D result: motion identification and exclusion step, registration step, reconstruction step, and regularization step. For the motion identification and exclusion step, we should estimate the amount of motion and exclude the slices with large amount of motion corruption. Early reconstruction approaches need to manually exclude the motion corrupted slices. The intersection-based motion correction approach can automatically detect and reject motion corrupted and incorrect registration slices by the abnormal level of their mean squared intensity difference with respect to all other intersecting slices [15]. In [16], Kainz et al. have proposed an approach to automatically estimate the amount of motion based on the low-rank decomposition for linearly correlated image slices [17]. Using this approach, we can reject stacks with large motion and choose the stack with the least motion as the template to prepare for the registration step. Registration step can be utilized to correct the motion between slice and the reconstructed volume. Rousseau et al. [18] combined the 2D/3D registration with the PSF to achieve the 3D reconstruction. PSF [14] is a mathematical function to model the actual appearance of data points in physical space. By PSF, we can physically correct estimation of the image acquisition process. Subsequently, the SVR method was modified to improve the robustness of the 2D/3D registration [19]. For the reconstruction step, superresolution methods [20, 21] are utilized to reconstruct the 3D volume. In [22], Gholi-pour and Warfield combined the superresolution method with slice-to-volume registration to reduce the burring effect. Because the motion identification and exclusion steps can exclude the slice of which the motion amount is greater than the threshold, the amounts of the slightly corrupted slices are still preserved for reconstruction. Using the robust superresolution volume reconstruction method [23], the weight of slightly corrupted and misaligned slices would be reduced to minimize the effect of motion. During the process of superresolution reconstruction, maximum likelihood estimation (MLE) is treated as an optimum solution to estimate the point's value [24]. To get better results, we should minimize the difference between the estimated slices and the acquired slices. Since the minimization only depends on the acquired samples, the estimation in the MLE framework is ill-posed and inaccurate when the samples are sparse [23]. The regularization step is used to solve the overfitting problem, and it can reduce image noise and registration errors. In [25], Charbonnier et al. proposed a deterministic edge-preserving regularization method to deal with image. However, this method makes it difficult to avoid the smoothing of edges. Adaptive regularization techniques can be employed to reduce the smoothing effects of regularization [26]. In [27], Rousseau et al. took advantage of

total variation regulation to extend the superresolution reconstruction method.

The general SVR framework with the superresolution reconstruction method has been developed in [28]. One important way to alleviate fetal motion is to quicken the data acquisition time by the sparse data sampling technique. However, the traditional SVR method could not deal well with the sparse sampling problem and cannot provide high-quality image. In this paper, we utilize the SVR method with adaptive kernel regression to cope with the sparse volume reconstruction with minimum motion artifacts under the condition of sparse data acquisition. The key improvements compared to previous works are as follows: firstly, we make use of the sparse samples to get faster speed of data acquisition in fetal brain MRI. Next, the adaptive kernel regression-based reconstruction method [29] with robust statistics calculation [24] can reconstruct high-quality volume under the condition of sparse sampling. In general, our comprehensive reconstruction method for fetal brain MRI mainly includes slice-to-volume registration, the robust statistics calculation, the PSF-based volume update, and adaptive kernel regression-based volume update.

The rest of the paper is organized as follows. The detailed methodology is discussed in Section 2. We design the actual implementation of the algorithm in Section 3. Section 4 involves the experiment results and compares with those of superresolution methods. In this section, we also discuss how to determine the optimal values of related parameters using GPU-based fast reconstruction. Finally, we make a brief conclusion in Section 5.

## 2. Methods

*2.1. Model of Data Acquisition and Motion Estimation.* During data acquisition of fetal brain MRI, we collected several stacks of 2D slices in different orientations. Because of the fetal motion, the movement could be observed between these slices. Assume that the acquired  $k$  misaligned 2D slices are  $I_j \in \mathbf{R}^{n \times h}$ ,  $j = 1, \dots, k$ , and the corresponding sparse 2D slices are  $I_j^s \in \mathbf{R}^{n \times h}$ ,  $j = 1, \dots, k$ . During the slice acquisitions of MRI, the inhomogeneity of the magnetic field  $B_j$ ,  $j = 1, \dots, k$ , affects the intensities of the slices and the scaling factor  $S_j^p$ ,  $j = 1, \dots, k$ , is potentially different for each acquired slices. In [30], the logarithmic transformation was chosen to make the bias additive. However, field in-homogeneities are known to be multiplicative. Differently, we use the multiplicative bias field to form the multiplicative exponential model which replaces the logarithmic model. So the scaled and bias corrected slice  $I_j^s$  can be modeled as

$$\begin{aligned} I_j^s &= \text{sparse}(I_j), \\ \text{vec}(I_j^s) &= S_j^p \cdot \exp(-B_j) \cdot \text{vec}(I_j^s), \end{aligned} \quad (1)$$

where  $I_j^s$  is the sparsely sampled slice coming from the sparse operator  $\text{sparse}(\bullet)$ ,  $\text{vec}(\bullet)$  is the vectorization operator that transforms a  $m$ -pixel ( $m = n \times h$ ) image  $\mathbf{R}^{n \times h}$  into a vector

of intensity values  $\mathbf{R}^m$ . The corresponding  $k$ -aligned 2D ground-truth slices are  $I_j^* \in \mathbf{R}^{n \times h}$ ,  $j = 1, \dots, k$ . The relationship between corrected slices  $I_j'$  and the ground-truth slices  $I_j^*$  can be denoted as follows:

$$\text{vec}(I_j') = \theta_j \cdot \text{vec}(I_j^*) + \text{vec}(e_j), \quad j = 1, \dots, k, \quad (2)$$

where  $e_j$  is the motion error, and  $\theta_j$  denotes the unknown motion transformation parameter of slice  $I_j^*$ . Then, we can define the following data matrix:

$$\begin{aligned} D &= [\text{vec}(I_1') ; \dots ; \text{vec}(I_k')] \in \mathbf{R}^{m \times k}, \\ X &= [\text{vec}(I_1^*) ; \dots ; \text{vec}(I_k^*)] \in \mathbf{R}^{m \times k}, \\ E &= [\text{vec}(e_1) ; \dots ; \text{vec}(e_k)] \in \mathbf{R}^{m \times k}, \\ T_{\text{total}} &= [\theta_1 ; \dots ; \theta_k] \in \mathbf{R}^{m \times k}. \end{aligned} \quad (3)$$

where  $D$ ,  $X$ ,  $E$ , and  $T_{\text{total}}$  denote the observed data matrix, reconstructed data matrix, motion error matrix, and the rigid transformation matrix. Given these definitions, the observed data matrix  $D$  can be described as  $D = T_{\text{total}} \cdot X + E$ . The motion error matrix  $E$  is mainly caused by misaligned slices. The misaligned slices can cause the inaccurate reconstructed volume, and we want to exclude the stack which has many misaligned slices. However, we cannot directly calculate the amount of stack motions for the observed data matrix  $D$ , but a low-rank approximation  $D^*$  as surrogate estimate can be used to evaluate the stack motion indirectly [16]. It has been shown that  $D^*$  provides the best approximation to  $D$  [31]. The difference value between  $D^*$  and  $D$  measures the motion error  $E$ . The smaller difference value indicates that the stack has fewer motions. To provide the low-rank approximation, the singular value decomposition is used to decompose the data matrix  $D$  as  $D_{m \times k} = U_{m \times k} S_{k \times k} V_{k \times k}^T$ . The singular value decomposition of  $D$  produces three matrices  $U$ ,  $S$ , and  $V$ .  $U$  and  $V$  are both orthogonal matrices, and  $S$  is the diagonal matrix containing the singular values on the diagonal. And the singular value decomposition of  $D^*$  is the first  $r$  singular values of the original matrix  $D$ , i.e.,  $D_{m \times k}^* = U_{m \times r}^* S_{r \times r}^* V_{r \times k}^{*T}$ ,  $r = 1, \dots, k$ .  $U^*$  and  $V^*$  are the first  $r$  columns of  $U$  and  $V$ , and  $S^*$  is the top left  $r \times r$  submatrix of  $S$ . The relative error based on the Frobenius norm  $\|D - D^*\|$  is used to measure the approximation between  $D^*$  and  $D$ , i.e.  $\delta_r = \|D - D_r^*\| / \|D\|$ . For the different values of  $r = 1, \dots, k$ , we can find the minimal rank  $r$  for each stack that satisfies the given threshold  $\beta$ , i.e.,  $\arg \min_r \{\delta_r < \beta\}$ . Combining  $\delta_r$  and  $r$ , the surrogate estimate for the amount of motion is given by  $\mu_r = \delta_r \cdot r$ .

Based on the low-rank decomposition method, we can choose one stack with minimal motion as the target template and first perform the 3D rigid volumetric registration between the target template and the other stacks (stack to

template registration). During the first registration, we can get the corresponding rigid global transformation matrix  $T_{\text{global}}$ . Then, second, the 3D rigid volumetric registration between the reconstructed volume and all slices (slice to reconstructed volume registration) can produce local transformation matrix  $T_{\text{local}}$ . The prerequisite for two registrations is that all stacks and reconstructed volume should be mapped to the world coordinates. Thus, we need to define two transformations to map each pixel in the 2D slice and each voxel in the reconstructed volume to a continuous location in the world coordinates. The first one is world transformation  $W_s = [\theta_1^w, \dots, \theta_k^w]$  that transforms the discrete coordinates of a pixel  $p_s = [i, j, 0, 1]^T \in I_j^s$  in the acquired slice to the continuous local world coordinates. The second one is world transformation  $W_r = [\theta_1^{w'}, \dots, \theta_k^{w'}]$  that transforms the discrete coordinates of a voxel  $p_r = [x, y, z, 1]^T \in X$  in the reconstructed volume to the continuous local world coordinates. Meanwhile, the mapping and registrations can be combined and formulated as Equation (4). Thus, Figure 1 illustrates the whole transformation process from the pixels in the sparse slice to voxels in the 3D reconstructed volume.

$$p_r = [W_r^{-1} \cdot T_{\text{total}} \cdot W_s] \cdot p_s = [W_r^{-1} \cdot (T_{\text{global}} \cdot T_{\text{local}}) \cdot W_s] \cdot p_s. \quad (4)$$

**2.2. PSF-Based Volume Update.** To model the actual appearance of sampling data points in physical space, the point spread functions (PSFs) are used to make the exact estimation for every voxel value in the reconstructed target volume. For the MRI ssFSE sequence in this paper, the exact shape of the PSF has been measured using a phantom and rotating imaging encoding gradient in [14]. The resulting shapes of the PSF in in-plane and in through-slice are given by a sinc function and the slice profile, respectively. Since the ideal rectangle profile has the very dense and inefficient spatial sampling, Kuklisova-Murgasova et al. [28] have proposed to use the 3D Gaussian function with the full width at half maximum (FWHM) equal to the slice thickness as an approximation for the sinc function. The PSF function based on 3D Gaussian profile is used to approximately model the SSFSE sequence and is expressed as follows:

$$\text{PSF}_G = \exp \left( \frac{-dx^2}{2\sigma_x^2} + \frac{-dy^2}{2\sigma_y^2} + \frac{-dz^2}{2\sigma_z^2} \right), \quad (5)$$

where  $dx$ ,  $dy$ , and  $dz$  are the offsets from the center of a reconstructed voxel,  $\sigma_x$  and  $\sigma_y$  are the full width at half maximum (FWHM) in the in-plane  $x$ - and  $y$ -directions, and the  $\sigma_z$  equals to the slice thickness in the through-plane direction. For each pixel in the sampled slice, the  $\text{PSF}_G$  is applied to obtain the corresponding PSF coefficient matrix. Since every sampling pixel (i.e.,  $p_s$ ) does not perfectly align itself with the reconstructed voxel (i.e.,  $p_r$ ), one  $p_s$  contributes to more than one  $p_r$ . To model this, every voxel is sampled

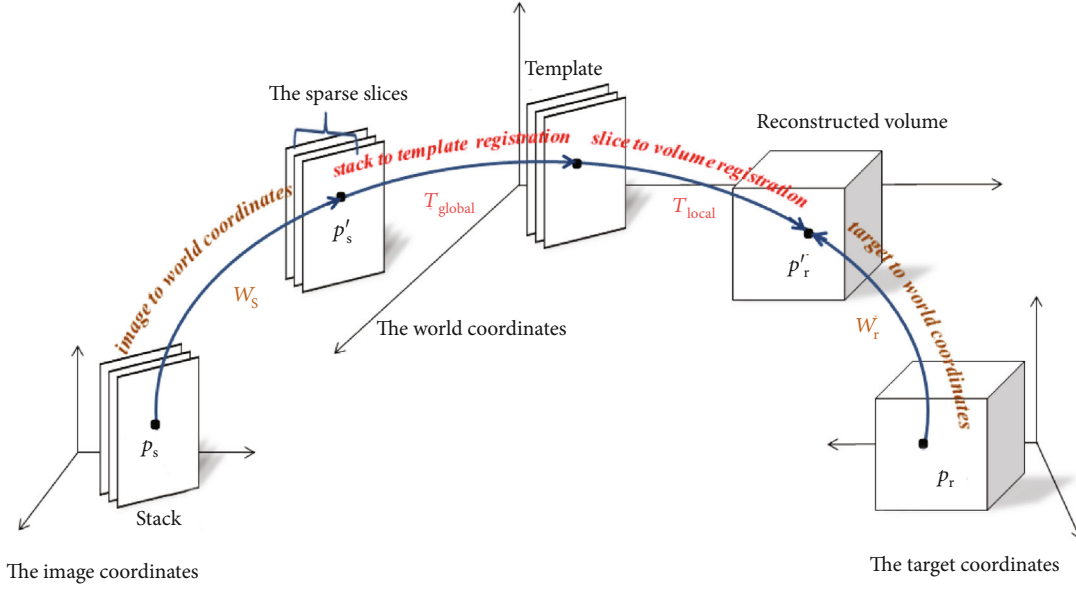


FIGURE 1: The illustration of the whole transformation process from pixels  $p_s$  to voxel  $p_r$ .

around its local surrounding neighbor in the reconstructed volume to make sure that it has at least one corresponding pixel in the acquired slices. Then, the PSF coefficients are used to weigh the pixel's contribution during  $n$ th iteration.

$$p_r = \lfloor (W_r^{-1} \cdot T_{\text{total}} \cdot W) \cdot p_s \rfloor, \tilde{p}_s = (W_r^{-1} \cdot T_{\text{total}} \cdot W)^{-1} p_r, \\ X(p_r^{n+1}) = \text{PSF}(p_s - \tilde{p}_s) \cdot S'_j \cdot \exp(-B_j) \cdot I_j^i(p_s) + X(p_r^n), \quad (6)$$

where  $\lfloor \cdot \rfloor$  is the operation that finds the nearest voxel center in the space of the reconstructed volume. The reconstructed volume  $X$  is updated iteratively through the PSF-based data sampling model, and every voxel of  $X$  is filled at an arbitrarily chosen voxel size.

**2.3. Robust Outlier Removal.** Once the target volume is updated based on the Gaussian PSF, the simulated slices  $I^{\text{ss}} = [I_1^{\text{ss}}, \dots, I_k^{\text{ss}}] \in \mathbf{R}^{n \times h}$  can be generated from the updated reconstructed volume. Then, the misaligned error  $e^*$  between the corrected acquired sparse slices  $I'$  and simulated slices  $I^{\text{ss}}$  can be computed as

$$E(e^*) = I'(p_s) - I^{\text{ss}}(p_s). \quad (7)$$

In [28], an EM model-based robust statistics approach was proposed to classify each slice pixel into two classes: inliers and outliers. Specially, the probability density function (PDF) for the inlier class is modeled as a zero-mean Gaussian distribution with variance  $\sigma^2$ :  $E \sim N(0, \sigma^2)$ , and the PDF for the outlier class is modeled as a uniform distribution with constant density, which is a reciprocal of the range  $[a, b]$ :  $E \sim U(a, b)$ . Then, the likelihood of the observing error  $e^*$

can be expressed as

$$P(e^* | \sigma, c) = c \cdot N_\sigma(e^*) + (1 - c) \cdot U, \quad (8)$$

where  $c$  is a mixing proportion of inliers representing the correctly matched voxels. Then, the posterior probability of a voxel being an inlier can be computed using the expectation step as

$$p_{ij} = \frac{c \cdot N_\sigma(e_{ij}^*)}{c \cdot N_\sigma(e_{ij}^*) + (1 - c) \cdot U}. \quad (9)$$

The variables  $\sigma$  and  $c$  are updated by the following maximization step:

$$\sigma = \sqrt{\frac{\sum p_{ij} \cdot (e_{ij}^*)^2}{\sum p_{ij}}}, \quad (10) \\ c = \frac{\sum p_{ij}}{\sum N_j},$$

where  $N$  is the number of the pixels in the slice. By constantly iterating, we can get the best parameters  $\sigma$  and  $c$ . The inlier probability can be used to weigh the PSF-based volume update. By the same way, each slice is classified into inlier and outlier as well using the EM algorithm. The probability of an inlier slice is defined as  $p_j^{\text{slice}} = \sqrt{\sum_i p_{ij}^2 / N_j}$ . The slices inferred to be an outlier are excluded from the PSF-based volume update to remove artifacts of motion corruption and misregistration.

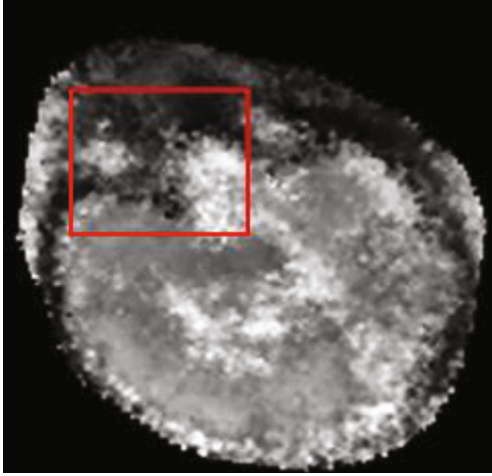


FIGURE 2: The reconstructed volume after PSF-based volume update.

The purpose of the outlier removal is to make the framework more robust by rejecting the outlier slices. The outlier removal module is adopted directly from the cited previous work [16], where the accuracy of the motion recognition and outlier removal has been evaluated in detail by simulating the slice motion at a variety of amplitudes and comparing the known motion amplitude to the surrogate measure provided through rank approximation. They have shown that there was strong correlation between the amplitude of the known motion and the values of  $\mu_r$  derived from the stack data matrices.

**2.4. Steering Kernel Regression-Based Volume Update.** For sparse reconstruction, it is experimentally found that the reconstructed volume still remains unallocated or inaccurate voxels after PSF-based volume update and the reconstructed result is noise as shown in Figure 2.

In [32], the kernel regression can make better nonparametric estimation for the empty pixels. In this paper, the steering kernel regression approach [29] is introduced to update the voxels for the previous sparse volume data. The model for the kernel regression is expressed as

$$Y_i = r(X_i) + \varepsilon_i, \quad i = 1, \dots, M, \quad (11)$$

where  $r(\bullet)$  is the function of kernel regression,  $X_i = (x_i, y_i, z_i)$  is the 3D coordinate of the voxel,  $\varepsilon_i$  is a zero-mean Gaussian noise with variance  $\sigma_0^2$  as  $X \sim N(0, \sigma_0^2)$ , and  $Y_i$  is the voxel after PSF-based Gaussian volume update.

Assuming that the voxel  $X_i$  is close to the known voxel  $X$  in the reconstructed volume, we have the following approxi-

mation for  $r(X_i)$  using the  $N$ -term-order Taylor series:

$$\begin{aligned} r(X_i) &\approx r(X) + \{\nabla r(X)\}^T (X_i - X) \\ &\quad + \frac{1}{2!} (X_i - X)^T \{Hr(X)\} (X_i - X) + \dots \\ &= \beta_0 + \beta_1^T (X_i - X) + \beta_2^T \text{vech} \\ &\quad \cdot \left\{ (X_i - X)(X_i - X)^T \right\} + \dots, \end{aligned} \quad (12)$$

where  $\nabla$  and  $H$  are, respectively, the gradient ( $3 \times 1$ ) and Hessian ( $3 \times 3$ ) operators;  $\beta_0 = r(X)$ , which is the voxel value of interest; and the vectors  $\beta_1$  and  $\beta_2$  are defined as

$$\beta_1 = [G_x, G_y, G_z]^T = \left[ \frac{\partial r(X)}{\partial x}, \frac{\partial r(X)}{\partial y}, \frac{\partial r(X)}{\partial z} \right]^T, \quad (13)$$

$$\begin{aligned} \beta_2 = \frac{1}{2} \left[ \frac{\partial^2 r(X)}{\partial x^2}, 2 \frac{\partial^2 r(X)}{\partial x \partial y}, 2 \frac{\partial^2 r(X)}{\partial x \partial z}, \right. \\ \left. \frac{\partial^2 r(X)}{\partial y^2}, 2 \frac{\partial^2 r(X)}{\partial y \partial z}, \frac{\partial^2 r(X)}{\partial z^2} \right]^T. \end{aligned} \quad (14)$$

$\text{vech}(\bullet)$  is the half-vectorization operator that transforms the upper triangular portion of a symmetric matrix into a column-stacked vector, i.e.,

$$\text{vech} \left( \begin{bmatrix} a & b & c \\ b & d & e \\ c & e & f \end{bmatrix} \right) = [a \ b \ c \ d \ e \ f]^T. \quad (15)$$

Based on the least-squares formula, we can optimize Equation (12) as

$$\begin{aligned} \min_{\{\beta_n\}_{n=0}^L} \sum_{i=1}^L [Y_i - \beta_0 - \beta_1(X_i - X) \\ - \beta_2(X_i - X)^2 - \dots]^2 \cdot \frac{1}{h} K \left( \frac{X_i - X}{h} \right), \end{aligned} \quad (16)$$

where  $L$  is the number of known voxels within the neighborhood window,  $K(\bullet)$  is the distance-weighted kernel function which penalizes distance away from the local position, and  $h$  is the smoothing parameter that controls the strength of the penalty. The kernel function is chosen as the exponential function, Gaussian function, or other functions which satisfy the following conditions:

$$\begin{aligned} \int tK(t)dt &= 0, \\ \int t^2K(t)dt &= c. \end{aligned} \quad (17)$$

For the computation simplicity, the Gaussian-based

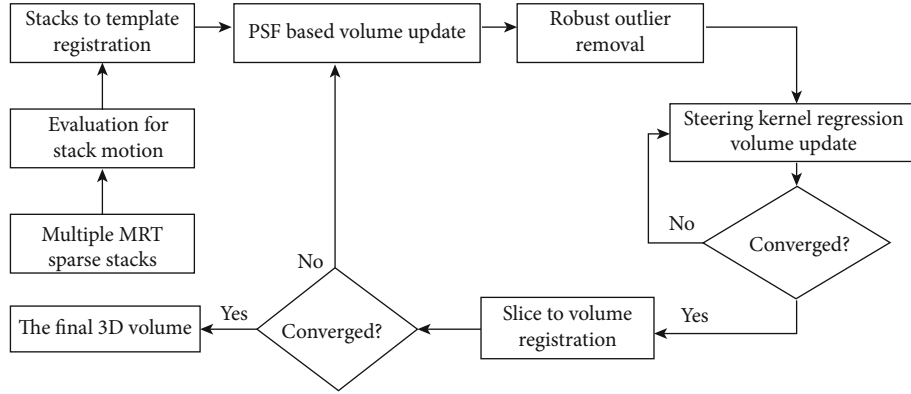


FIGURE 3: Flowchart of the proposed algorithm.

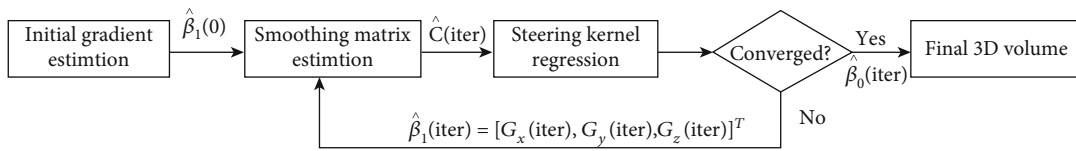


FIGURE 4: The iterative steering kernel regression.

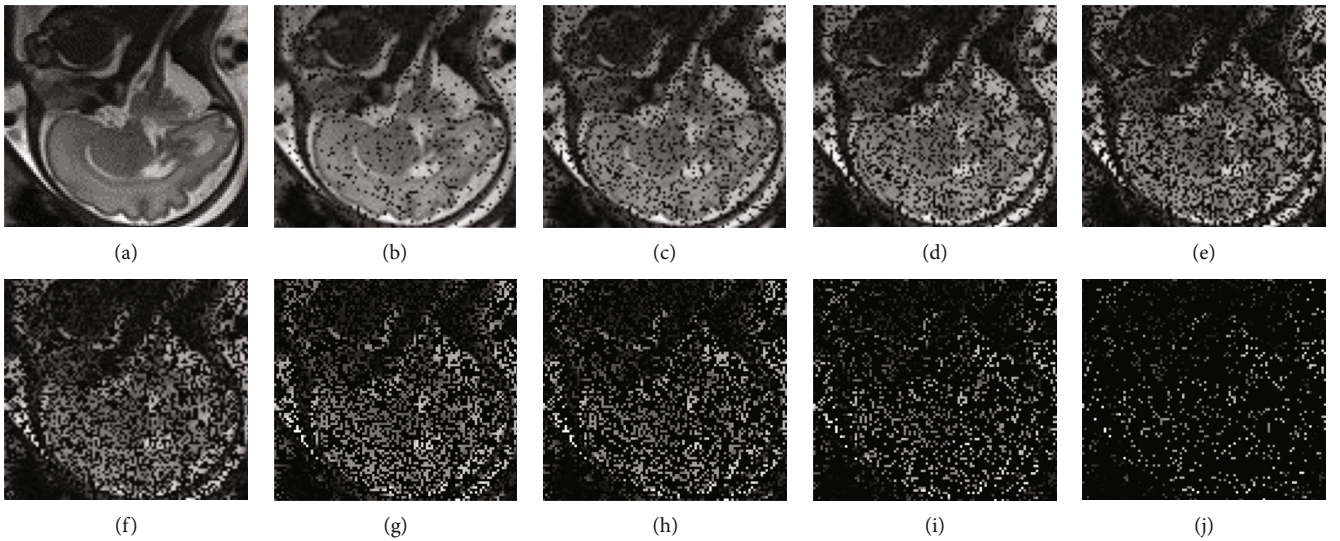


FIGURE 5: The original and spare slices: (a) the typical 30th original slice; (b–j) the corresponding simulated sparse slice by removing once every 10% proportion pixels ranging from 10% to 90%.

kernel function is chosen in the steering kernel regression [33]. The steering kernel adapts locally to image structures (e.g., edges, flat, and texture areas), which are captured by the kernel footprint. For example, the kernel footprint is large in the flat areas, elongated in edge areas, and compact in texture areas. The 3D steering kernel function takes from

$$K_s(X_i - X) = \frac{\sqrt{\det(C_i)}}{2\pi h^2} \exp \left\{ -\frac{1}{2h^2} \|C_i^{1/2}(X_i - X)\|_2^2 \right\}, \quad (18)$$

where  $\|\cdot\|_2^2$  is the  $L_2$  norm and  $C_i$  is the symmetric covariance matrix. Since the local image structure is highly related to the gradient covariance, we can make the data-dependent

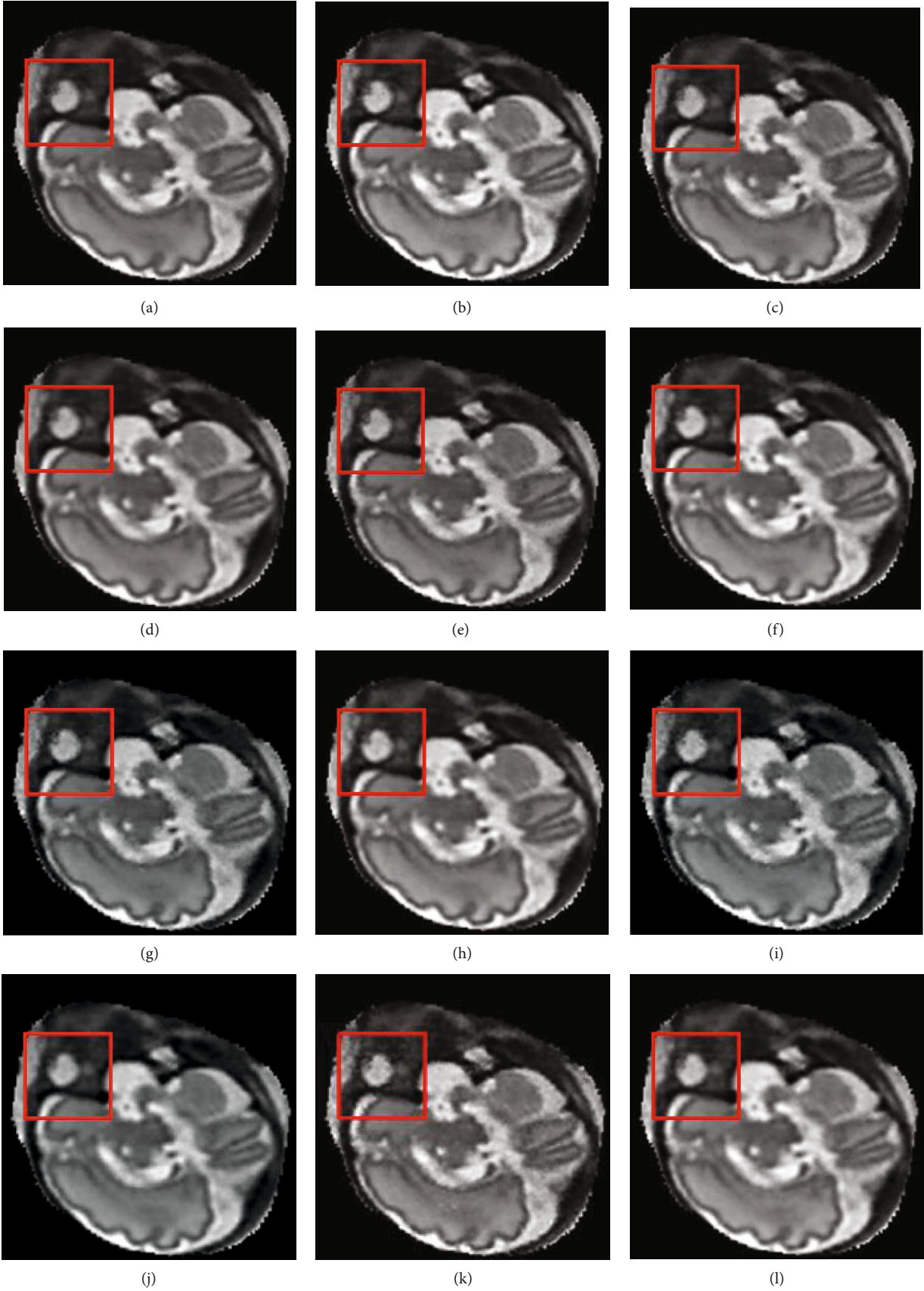


FIGURE 6: Continued.

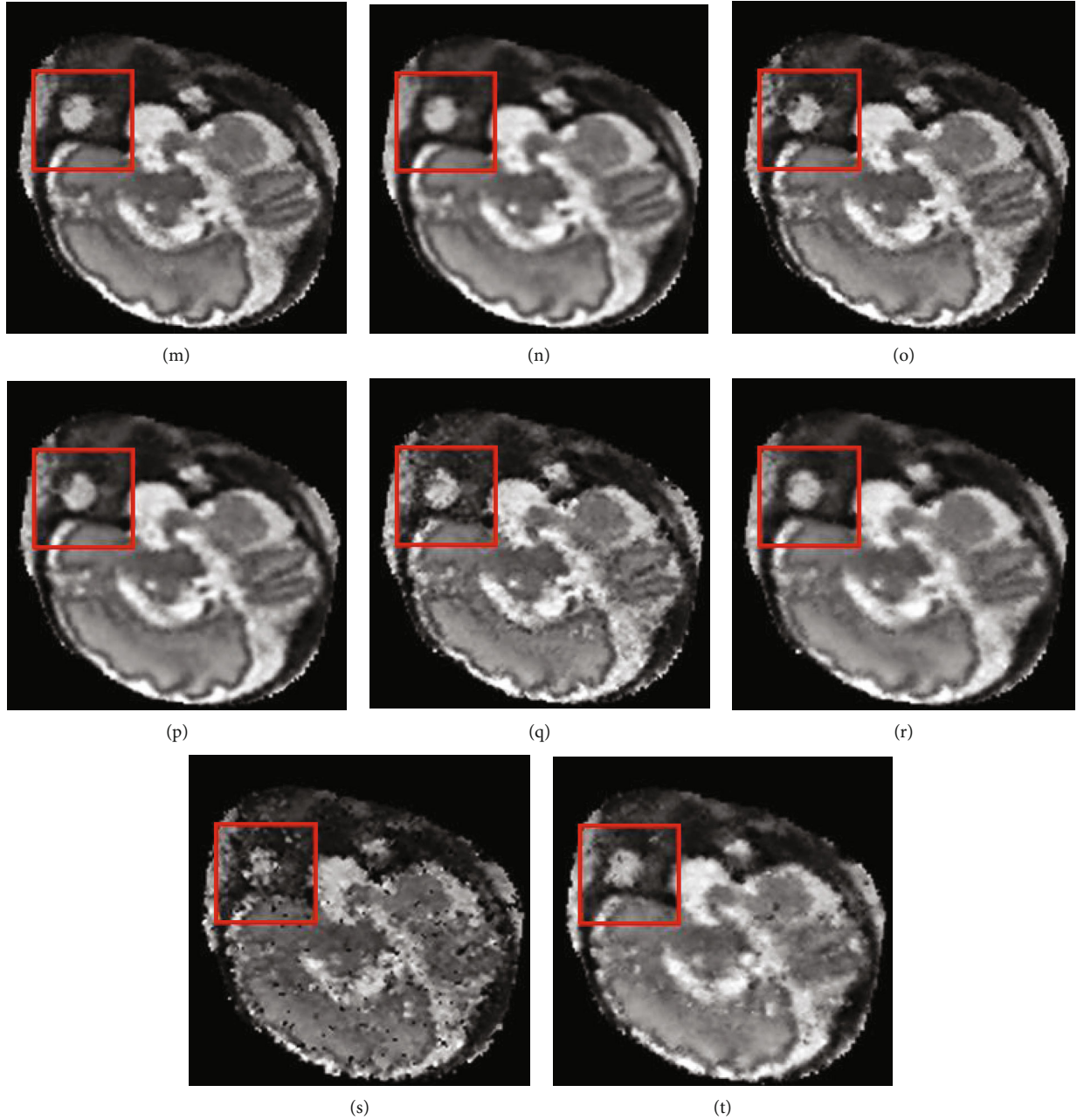


FIGURE 6: Reconstruction results of different data removal ratio by Kainz et al. method (2015) and our proposed method. (a), (c), (e), (g), (i), (k), (m), (o), (q), (s) are the reconstructed results by Kainz et al. method for the sparsely sampled dataset with once every 10% data removal ratio ranging from 0% to 90% respectively. (b), (d), (f), (h), (j), (l), (n), (p), (r), (t) are the reconstructed results by the proposed method for the sparsely sampled dataset with once every 10% data removal ratio ranging from 0% to 90%, respectively. The red rectangle points to the obvious difference, which appears as artifacts in the reconstructed image if no steering kernel regression volume updated is used.

covariance matrix estimation utilizing the local edge gradients:

$$\hat{C}_i \approx \begin{bmatrix} \sum_{X_i \in \omega} G_x(X_i)G_x(X_i) & \sum_{X_i \in \omega} G_x(X_i)G_y(X_i) & \sum_{X_i \in \omega} G_x(X_i)G_z(X_i) \\ \sum_{X_i \in \omega} G_x(X_i)G_y(X_i) & \sum_{X_i \in \omega} G_y(X_i)G_y(X_i) & \sum_{X_i \in \omega} G_y(X_i)G_z(X_i) \\ \sum_{X_i \in \omega} G_x(X_i)G_z(X_i) & \sum_{X_i \in \omega} G_y(X_i)G_z(X_i) & \sum_{X_i \in \omega} G_z(X_i)G_z(X_i) \end{bmatrix}, \quad (19)$$



TABLE 1: The RMSE and SSIM value comparison of fetal brain reconstruction with different removal proportions, respectively.

Different removal proportions	RMSE		MSSIM	
	Kainz et al.'s method (2015)	Our method	Kainz et al.'s method (2015)	Our method
0%	19.096	19.096	1	1
10%	32.578	29.120	0.9752	0.9759
20%	38.043	33.947	0.9690	0.9691
30%	43.171	36.790	0.9591	0.9608
40%	49.194	41.027	0.9478	0.9458
50%	55.894	45.480	0.9202	0.9366
60%	67.053	53.305	0.9063	0.9271
70%	81.522	62.964	0.8667	0.8993
80%	111.917	78.886	0.7862	0.8449
90%	180.483	112.249	0.6338	0.7407

where  $w$  is a local analysis window and  $G_x(\bullet)$ ,  $G_y(\bullet)$ , and  $G_z(\bullet)$  are the gradients along the  $x$ -,  $y$ -, and  $z$ -directions.

Equation (16) can be expressed in the matrix form as

$$\hat{\mathbf{B}} = \min_{\mathbf{b}} \|\mathbf{Y} - \mathbf{X}_F \mathbf{B}\|_{\mathbf{W}}^2 = \min_{\mathbf{b}} (\mathbf{Y} - \mathbf{X}_F \mathbf{B})^T \mathbf{W} (\mathbf{Y} - \mathbf{X}_F \mathbf{B}), \quad (20)$$

where  $\mathbf{Y} = [Y_1, Y_2, \dots, Y_L]^T$  is the vector set of all known voxels,  $\mathbf{B} = [\beta_0, \beta_1^T, \dots, \beta_N^T]^T$  is the vector set of all estimated parameters,  $\mathbf{W} = \text{diag}[K_s(X_0 - X), K_s(X_1 - X), \dots, K_s(X_L - X)]$  is the diagonal matrix whose elements on the diagonal are the value of  $K_s(\bullet)$ , and the other elements are zero. According to the least-squares method, we have the following solution:

$$\hat{\mathbf{B}} = (\mathbf{X}_F^T \mathbf{W} \mathbf{X}_F)^{-1} \mathbf{X}_F^T \mathbf{W} \mathbf{Y}, \quad (21)$$

where  $\mathbf{B} = [\beta_0, \beta_1^T, \dots, \beta_N^T]^T$ ,  $\hat{r}(X_i) = \hat{\beta}_0 = \mathbf{e}_1^T (\mathbf{X}_F^T \mathbf{W} \mathbf{X}_F)^{-1} \mathbf{X}_F^T \mathbf{W} \mathbf{Y}$  is the voxel value estimated by the steering kernel regression,  $\hat{\beta}_1 = [G_x, G_y, G_z]^T$  is applied for computing the symmetric covariance matrix  $\hat{\mathbf{C}}_{i+1}$  iteratively, and  $\mathbf{X}_F$  is a coordinate matrix expressed as follows:

$$\mathbf{X}_F = \begin{bmatrix} 1 & (X_0 - X)^T & \text{vech}^T \{(X_0 - X)(X_0 - X)^T\} & \dots \\ 1 & (X_1 - X)^T & \text{vech}^T \{(X_1 - X)(X_1 - X)^T\} & \dots \\ \cdot & \cdot & \cdot & \dots \\ \cdot & \cdot & \cdot & \dots \\ 1 & (X_L - X)^T & \text{vech}^T \{(X_L - X)(X_L - X)^T\} & \dots \end{bmatrix}. \quad (22)$$

Once the reconstructed volume is updated based on the steering kernel regression, we update the simulated slices  $I^{ss}$  and the misaligned error  $E(e^*)$  according to Equation (7). To remove artifacts caused by motion corruption and misregistration and enhance image edges, we further update

the reconstructed volume using the following equation:

$$X(p_r^{n+1}) = \text{PSF}(p_s - \tilde{p}_s) \cdot p_j^{\text{slice}} \cdot p_{ij} \cdot E(e^*) + X(p_r^n). \quad (23)$$

### 3. Implementation

The experiment computer is equipped with Intel Core i5 2.6 GHz CPU, and the operating system is Windows 7 64 bit. We have implemented the proposed algorithm using the Microsoft Visual Studio 2012 and Image Registration Toolkit (IRTK) software package which includes many useful methods to do registration, transformation, and other image processing. In this section, we discuss the key implementation details. The diagram of the total algorithm is expressed in Figure 3.

The first step is to evaluate the stack motion according to the method of low-rank decomposition. We estimate the amount of the stack motion by the surrogate  $\mu_r = \delta_r \cdot r$  and choose the stack with the minimum amount of stack motion as the template. The second step is to perform the global registration, which calculates the matrix of global transformation  $T_{\text{global}}$  from the other stacks to the template. The third step is the iterative registration-based volume reconstruction, which consists of the outer registration step and the inner reconstruction step. The outer loop step includes the PSF-based volume update, robust outlier removal, steering kernel regression-based volume update, and slice to volume registration. The PSF-based volume update step makes the initial estimation of the reconstructed volume based on Equation (6). Then, the simulated slices are created and used for the robust misaligned error calculation between the simulated slices and the acquired slices as described in Section 2.3. The robust statistic calculation achieves the classification of outlier slices and inlier slices. The outlier slices are excluded to remove artifacts of motion corruption and misregistration. The slice to volume registration is to calculate the local transformation  $T_{\text{local}}$  from slices to reconstructed volume. The whole transformation process is described by Equation (4). The volume update based on the adaptive steering kernel regression is aimed at reconstructing the accurate volume iteratively as shown in Figure 4. The initial gradients  $\hat{\beta}_1(0) = [G_x(0), G_y(0), G_z(0)]^T$  are estimated by the classical kernel regression. Then, the gradient information is used to calculate the covariance smoothing matrix  $\hat{\mathbf{C}}(\text{iter})$  (i.e., Equation (19)). We use smoothing matrix to update the voxel value  $\hat{\beta}_0(\text{iter})$  and its corresponding gradients  $\hat{\beta}_1(\text{iter})$  according to Equation (21), respectively. To obtain a more reliable voxel estimation, the process is iterated three times in our experiment.

### 4. Experimental Results and Evaluation

**4.1. Evaluation of Image Quality.** In the experimental evaluation, we used the datasets from the fetal MRI datasets [16], which were acquired by a Philips Achieva 3 T MR scanner. During the experiment, the volunteers were lying at a 20° tilt on the left side to avoid the pressure on the inferior vena cava.

TABLE 2: The running time comparison of the adaptive kernel regression method for fetal brain reconstruction based on CPU and GPU, respectively.

Processor	Single-threaded CPU	Multithreaded CPU	GPU	Single-threaded CPU vs. GPU	Multithreaded CPU vs. GPU
Gradient information (s)	416.960	108.762	5.047	82.62	21.55
Covariance smoothing matrix (s)	46.082	48.902	0.580	79.45	84.31
Steering kernel regression (s)	1402.727	887.629	15.091	92.95	58.82
Total time (s)	1865.769	1045.293	20.718	90.06	50.45

TABLE 3: The RMSE and MSSIM values and running time comparison of fetal brain reconstruction with different window sizes ranging from  $3 \times 3 \times 3$  to  $9 \times 9 \times 9$ , respectively.

Window size $w$	$w = 3$	$w = 5$	$w = 7$	$w = 9$
RMSE	125.061	125.869	129.298	126.929
MSSIM	0.6796	0.6663	0.6606	0.6763
TIME (s)	$7.056 = (4.810 + 0.534 + 1.712)$	$10.268 = (4.822 + 3.714 + 1.732)$	$124.678 = (4.801 + 118.211^* + 1.666)$	$222.353 = (4.798 + 215.888^* + 1.667)$

Note: TIME denotes the time caused only by running the adaptive kernel regression method.  $T = (A + B + C)$ :  $A$  is the time to calculate the gradient information.  $B$  is the time to calculate the covariance smoothing matrix.  $C$  is the time to calculate steering kernel regression function.  $T$  is the sum of  $A$ ,  $B$ , and  $C$ . \* indicates that CPU is chosen as the running processor for the covariance matrix calculation due to the limitation of GPU kernel memory for the large window size.

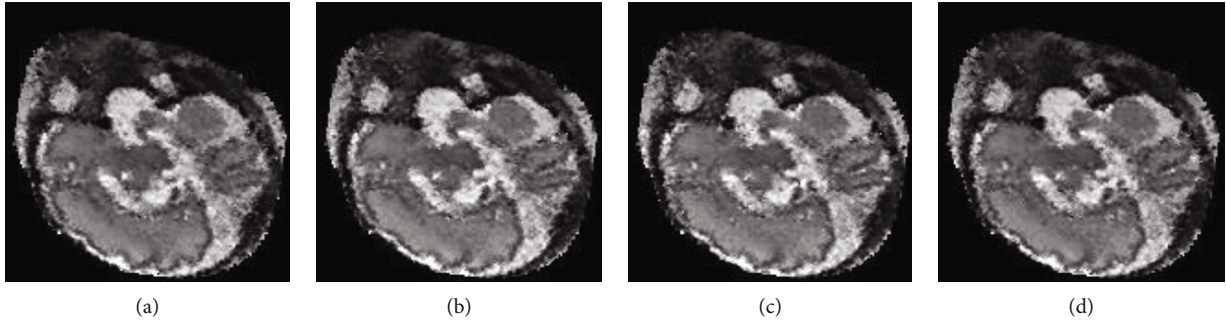


FIGURE 7: Reconstructed results of the MRI data with different window sizes  $w$ : (a)  $w = 3$ , (b)  $w = 5$ , (c)  $w = 7$ , and (d)  $w = 9$ .

The volunteer's womb was scanned with single-shot fast spin echo (SSFSE) T2-weighted sequence. Three stacks of images from axial, coronal, and sagittal orientation are used to construct the final high-resolution volume. To obtain the sparse stacks, we randomly remove different proportions of pixels of the stack once every 10% proportion ranging from 10% to 90%. The different removal proportions control the removal number of pixels. The typical 30<sup>th</sup> slice of the collected stack and its corresponding simulated sparse slices are illustrated in Figure 5.

For different data removal ratios, the sparse stacks are used to reconstruct the high-resolution 3D fetal brain MRI volume with the method of Kainz et al. [16] (SVR with super-resolution) and our proposed method. Figure 6 shows the reconstructed results by Kainz et al.'s method and the proposed method for the sparsely sampled dataset with once every 10% data removal ratio ranging from 0% to 90%, respectively. In Figure 6, we can observe that as the removal ratio increases, the reconstructed results by Kainz et al. method have much more noise for the sparse sampled dataset compared with our proposed method. On the other hand, the proposed method is capable of reconstructing high-

TABLE 4: The RMSE and MSSIM value comparison of fetal brain reconstruction with different structure sensitivities  $\alpha$  ranging from 0.1 to 0.5, respectively.

Structure sensitivity	$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$
RMSE	125.06	112.10	99.927	91.073	97.556
MSSIM	0.6823	0.7117	0.7584	0.7712	0.7523

resolution images without obvious artifacts even for the 90% data removal ratio.

For the sake of quantitative evaluation, the image quality assessment index of root mean square error (RMSE) [9] and mean structure similarity (MSSIM) [34] is introduced to quantitatively assess the algorithms under different removal ratios. The RMSE score can be computed by the following equation:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (z(i) - g(i))^2}, \quad (24)$$

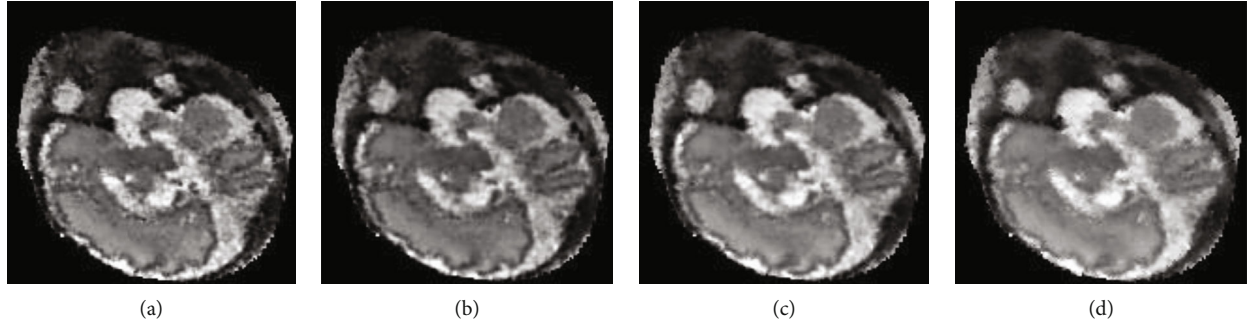


FIGURE 8: Reconstructed results of the MRI data with different structure sensitivities  $\alpha$ : (a)  $\alpha = 0.2$ , (b)  $\alpha = 0.3$ , (c)  $\alpha = 0.4$ , and (d)  $\alpha = 0.5$ .

where  $z(\bullet)$  is the reconstructed result,  $g(\bullet)$  is the ground-truth volume, and  $N$  is the number of voxels. A good reconstruction method is capable of estimating the removed data very close to the original data. Given  $z(\bullet)$  and  $g(\bullet)$ , a low RMSE value represents that the estimated result is satisfying while a high RMSE means that the interpolation accuracy is poor.

The structure similarity (SSIM) index explores the structural information for image quality assessment based on the main idea that the pixels have strong interdependency when they are spatially close. The SSIM metric is calculated based on the intensity, contrast, and structure and is computed as

$$\text{SSIM}(z, g) = \frac{(2\mu_z\mu_g + c_1)(2\sigma_{zg} + c_2)}{(\mu_z^2 + \mu_g^2 + c_1)(\sigma_z^2 + \sigma_g^2 + c_2)}, \quad (25)$$

where  $\mu_z, \mu_g, \sigma_z, \sigma_g,$  and  $\sigma_{zg}$  denote the mean, variance, and covariance on square window, which moves pixel by pixel in images  $z(i)$  and  $g(i)$ , respectively. The two variables  $c_1 = k_1L$  and  $c_2 = k_2L$  are used to stabilize the division with weak denominator. Here,  $L$  is the dynamic range of pixel value (e.g., 255 for 8-bit grayscale image), with  $k_1 = 0.01$  and  $k_2 = 0.03$  by default. Since the SSIM metric is calculated on various windows of a volume image, the mean SSIM (MSSIM) index is used in this experiment to assess the overall image quality:

$$\text{MSSIM}(z, g) = \frac{1}{M} \sum_{i=1}^M \text{SSIM}(z_i, g_i), \quad (26)$$

where  $M$  is the number of local windows in the image.  $\text{MSSIM}(z, g) \in [0, 1]$ ; the higher MSSIM indicates better structural similarity between two images.

For the clinical datasets, it is impractical to obtain the ground-truth volume in advance. For the sake of fair comparison among different methods, the quantitative evaluation is performed based on an average reconstructed volume. We first use the original stacks without data removal to reconstruct a complete volume by Kainz et al.'s method (2015) and our method (e.g., Figures 6(a) and 6(b)), respectively. Both volumes are adopted to create an average volume as the ground truth. Table 1 shows the quantitative results of the RMSE and MSSIM values with different data removal

TABLE 5: The RMSE and MSSIM value comparison of fetal brain reconstruction with the regularization parameter  $\lambda$  ranging from 0.1 to 2.0, respectively.

Regularization parameter $\lambda$	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 1.5$	$\lambda = 2.0$
RMSE	91.073	91.447	91.276	91.566	91.071
MSSIM	0.7720	0.7710	0.7708	0.7684	0.7732

ratios for each method. As can be seen, the results of Kainz et al.'s method produce the highest RMSE scores and lowest scores for all sampling rates. Both the high RMSE value and low MSSIM value for Kainz et al.'s method indicate poor image quality because of the artifacts and noise. For all levels of sampling rate, the proposed method performs better than the Kainz et al.'s method. More importantly, both of the difference of the RMSE and MSSIM index between Kainz et al.'s method and our method increase while the data removal ratio increases, indicating that our method outperforms much more compared with the Kainz et al.'s method when the data removal ratio increases.

**4.2. Evaluation of Computational Efficiency.** Our approach is capable of reconstructing the accurate volume from the highly sparse sampling dataset, but it requires largely computational burden as well due to the iterative kernel regression estimation. To reduce the long processing time of the adaptive kernel regression, the proposed method is accelerated by the GPU-based parallel implementation based on the NVIDIA GeForce GTX 1080 and CUDA 8.0 libraries. In the experiment, we make the evaluation of the computational efficiency of the adaptive kernel regression method, including the computation of the gradient information, the covariance smoothing matrix, and the steering kernel regression. The computational efficiency of the other modules (i.e., motion estimation, stack-to-template registration, PSF-based volume update, robust outlier removal, and slice-to-volume registration) has been evaluated in detail in [16]. The comparisons are based on the single-threaded CPU, multithreaded CPU, and GPU for the dataset of 80% data removal ratio under the parameter setting as the kernel size  $k_c = 5$  and the smoothing parameter  $h_c = 2.0$  in the initial gradient estimation step based on the classical kernel regression, the steering kernel size  $k_s = 7$  and the steering smoothing parameter  $h_s$

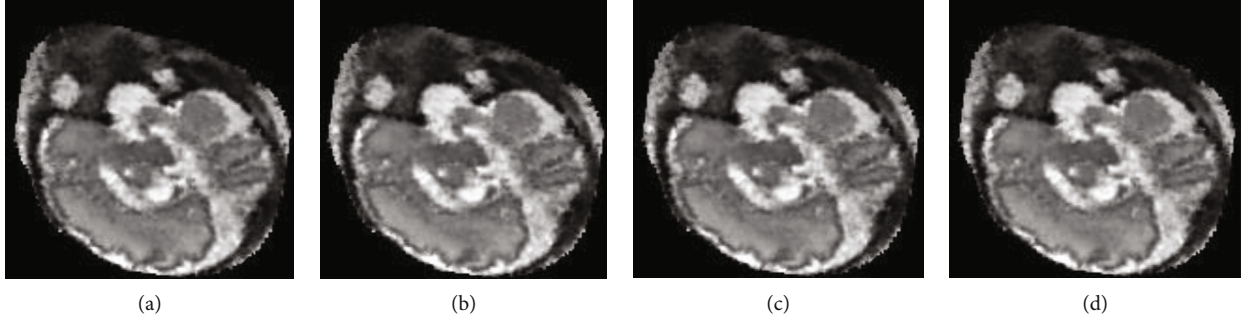


FIGURE 9: Reconstructed results of the MRI data with different regularization parameters  $\lambda$ : (a)  $\lambda = 0.5$ , (b)  $\lambda = 1.0$ , (c)  $\lambda = 1.5$ , and (d)  $\lambda = 2.0$ .

TABLE 6: The RMSE and MSSIM values and running time comparison of fetal brain reconstruction with different steering kernel sizes  $k_s$ , ranging from  $3 \times 3 \times 3$  to  $9 \times 9 \times 9$ , respectively.

Steering kernel size	$k_s = 3$	$k_s = 5$	$k_s = 7$	$k_s = 9$
RMSE	91.07	79.554	79.005	81.502
MSSIM	0.7791	0.7973	0.8054	0.7781
TIME (s)	$7.907 = (5.407 + 0.581 + 1.919)$	$11.995 = (5.404 + 0.581 + 6.010)$	$21.148 = (5.402 + 0.581 + 15.165)$	$37.333 = (5.408 + 0.581 + 31.344)$

Note: TIME denotes the time caused only by running the adaptive kernel regression method.  $T = (A + B + C)$ :  $A$  is the time to calculate the gradient information.  $B$  is the time to calculate the covariance smoothing matrix.  $C$  is the time to calculate the steering kernel regression function.  $T$  is the sum of  $A$ ,  $B$ , and  $C$ .

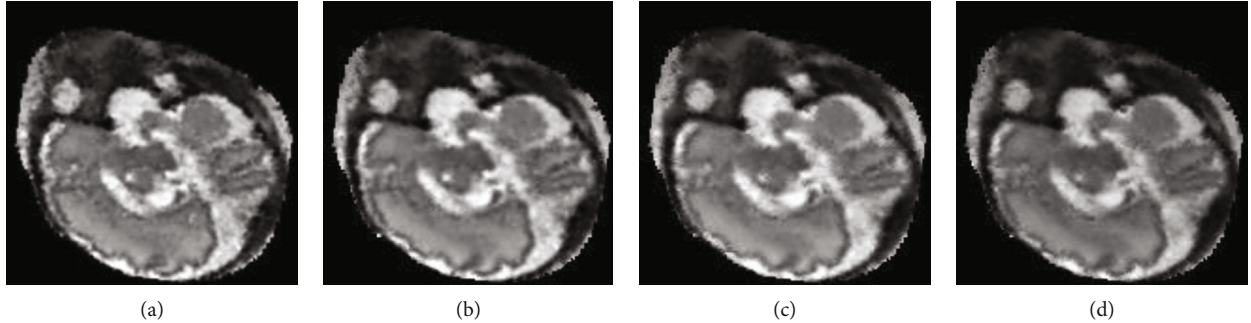


FIGURE 10: Reconstructed results of the MRI data with different kernel window sizes: (a)  $k_s = 3$ , (b)  $k_s = 5$ , (c)  $k_s = 7$ , and (d)  $k_s = 9$ .

TABLE 7: The RMSE and MSSIM value comparison of fetal brain reconstruction with different steering smoothing parameters  $h_s$ , ranged from 0.1 to 2.5, respectively.

Steering smoothing parameter	$h_s = 0.1$	$h_s = 0.5$	$h_s = 1.0$	$h_s = 1.5$	$h_s = 2.0$
RMSE	79.005	78.886	79.230	79.159	82.87
MSSIM	0.7927	0.8092	0.7933	0.7944	0.7946

$= 0.5$  in the steering kernel regression step, and the window size  $w = 3$ , the regularization parameter  $\lambda = 2.0$ , and the structure sensitivity  $\alpha = 0.4$ . The practical running time for the proposed method is shown in Table 2. For single-threaded CPU, the running time of the adaptive kernel regression method is 1865.769 s, which includes the computation of the gradient information in 416.96 s, the covariance smoothing matrix in 46.082 s, and the steering kernel regression in 1402.727 s. For the multithreaded CPU, we use 4 threads to run the adaptive kernel regression method and its computational time is 1045.293 s, indicating less improve-

ment compared with the single-threaded CPUs. The running time of the GPU implementation is 20.718 s in total. From Table 2, it can be observed that the GPU-based processing time has significantly decreased by 98.89% and 98.02%, compared with the single-threaded CPU and the multithreaded CPU, respectively.

#### 4.3. The Choice of the Adaptive Kernel Regression Parameters.

There are seven parameters which can be adjusted to affect the reconstructed image quality for the proposed method. These parameters include the kernel size  $k_c$  and the smoothing parameter (i.e., the kernel bandwidth)  $h_c$  in the initial gradient estimation step based on the classical kernel regression, the steering kernel size  $k_s$  and the steering smoothing parameter  $h_s$  in the steering kernel regression step, and the window size  $w$ , the regularization parameter  $\lambda$ , and the structure sensitivity  $\alpha$  ( $0 \leq \alpha \leq 0.5$ ) in the covariance matrix estimation step. In our method,  $k_c$  and  $h_c$  are related with the initial calculation of gradient information and have a negligible effect in the experiment. For the adaptive sparse

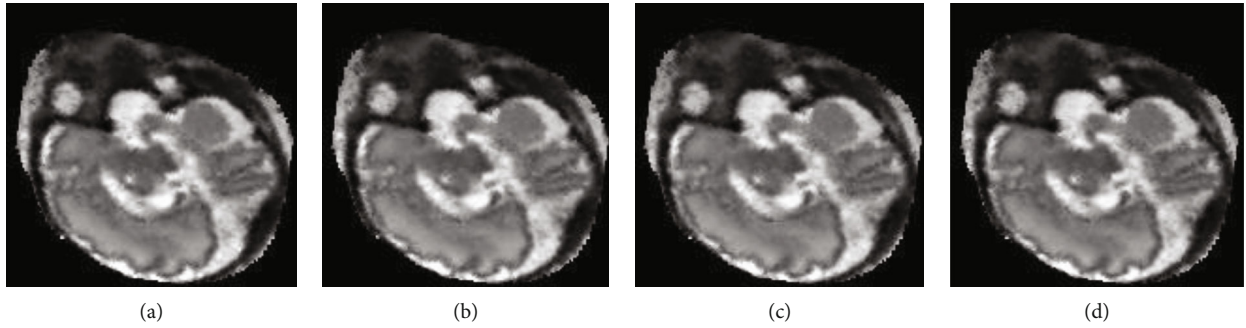


FIGURE 11: Reconstructed results of the MRI data with different steering smoothing parameters: (a)  $h_s = 0.5$ , (b)  $h_s = 1.0$ , (c)  $h_s = 1.5$ , and (d)  $h_s = 2.0$ .

reconstruction, covariance matrix estimation and steering kernel estimation are the two of the important steps and their parameters (i.e.,  $w$ ,  $\alpha$ ,  $\lambda$ ,  $k_s$ , and  $h_s$ ) play an important role in the volume reconstruction and deserve much more investigation.

With the help of GPU-based fast implementation, we firstly adjust the parameters (i.e.,  $w$ ,  $\alpha$ , and  $\lambda$ ) of the covariance matrix estimation one by one. The window size  $w$  decides how many neighbor points in the gradient matrix are taken for the estimation of the covariance matrix. Table 3 shows the RMSE and MSSIM values and the running time for different window sizes. Both of the RMSE and MSSIM values differ slightly, indicating that the window size has a negligible influence on the reconstructed image quality, as shown in Figure 7. However, the running time increases with the increase of window size. It can be observed that the window size of  $w = 3$  is chosen because of its faster implementation and lower RMSE value.

Table 4 shows the RMSE and MSSIM values influenced by the structure sensitivity parameter  $\alpha$ , and the lowest RMSE value and highest MSSIM value are obtained for the structure sensitivity  $\alpha = 0.4$  indicating the best performance of the algorithm. Figure 8 shows the corresponding reconstructed images for different  $\alpha$  values. As can be seen, the result with large structure sensitivity (e.g.,  $\alpha = 0.5$ ) results in oversmoothing image, while small structure sensitivity (e.g.,  $\alpha = 0.2$ ) overemphasizes the image edges. The experiment shows that the structure sensitivity  $\alpha$  has a significant influence on the reconstructed volume.

Under different regularization parameter settings, the RMSE and MSSIM measurements of the reconstructed results are calculated and shown in Table 5. The illustrative results are further shown in Figure 9. The regularization parameter  $\lambda$  is used to suppress the noise. However, the regularization parameter has negligible influence on the reconstructed image quality in the experiments.

The next group parameters (i.e.,  $k_s$  and  $h_s$ ) come from the steering kernel regression for the adaptive voxel value estimation. The kernel window size  $k_s$  has a great impact on the processing time for the kernel regression-based algorithm under different data removal proportions. When the kernel window increases, the estimation of each voxel involves more nearby pixels and leads to more computation [32]. The smaller the kernel window size is, the faster our algo-

rithm runs. On the other hand, if the size of the kernel window is too small, we could obtain the fault result, because there are not enough samples to make the current voxel estimation, especially for large data removal proportion. The larger the data removal proportion is, the sparser the sampled data will be. The RMSE and MSSIM index and processing time measurement of the reconstructed results under different kernel window sizes are shown in Table 6. With the increase of the kernel window size, the running time of steering kernel regression function is becoming longer. The corresponding images of different steering kernel sizes are shown in Figure 10. The proper kernel window size (i.e.,  $7 \times 7 \times 7$ ) produces a trade-off between the processing time and the reconstruction accuracy under different removal proportions.

Table 7 shows the RMSE and MSSIM values and running time with different steering smoothing parameters  $h_s$ . As can be seen, the results with the steering smoothing parameter (i.e.,  $h_s = 0.5$ ) achieve the lowest RMSE value and highest MSSIM value among these settings. The reconstructed results produced by different steering smoothing parameters are shown in Figure 11. In [33], it has been given that the steering smoothing parameter indicates the “footprint” of the kernel function. The large footprint of the kernel function could reduce the noise but at the cost of oversmoothing details, while small footprints are desirable to preserve the edges. In the experiment, the footprint setting  $h_s = 0.5$  is chosen for reaching a trade-off between the noise reduction and edge preservation. Finally, all parameters of the adaptive steering kernel regression algorithm are determined as follows: the window size  $w = 3$ , the regularization parameter  $\lambda = 2.0$ , the structure sensitivity  $\alpha = 0.4$ , the steering kernel size  $k_s = 7$ , and the steering smoothing parameter  $h_s = 0.5$ . Under such parameter setting, the RMSE value decreases from 126.47 to 78.89, indicating the quality improvement by 37.62%.

## 5. Conclusion

In this paper, we proposed an adaptive reconstruction method to deal with the sparse sampling dataset for fetal brain MRI. Our method combines the latest SVR framework, including the stack motion evaluation, PSF-based volume update, robust outlier removal, slice-to-volume registration,

and the proposed adaptive kernel regression-based volume update. Compared with the existing SVR framework, our method has advantages of sparse volume reconstruction and is capable of reconstructing superresolution image even for 80%~90% data removal. With the capability of sparse reconstruction, the data sampling time can be greatly shortened and thus, the motion artifacts can be reduced indirectly. To accelerate the voxel estimation, we use the CUDA to implement the steering kernel regression approach. For the proposed method, the running times of GPU-based implementation are speeded up to 90x than that of the CPU. The GPU-based parallel implementation of the proposed method is more practical to meet the requirements of fetal brain MRI. Meanwhile, we make the detailed investigation on the choice of parameters for the adaptive kernel regression-based volume reconstruction with the help of GPU-based fast implementation. To summarize, the structure sensitivity  $\alpha$  and the steering kernel window size  $k_s$  are two of the important parameters on sparse kernel regression volume reconstruction. Meanwhile, the kernel window size has a strong relationship with the running time. Larger window size requires longer processing time. Overall, our approach is used to reconstruct superresolution image from the highly sparse sampled dataset of fetal brain MRI corrupted with motion artifacts. One of its potential applications includes other motion organ MRI reconstruction, such as the heart MRI with the heart beating motion artifacts.

### Data Availability

The test data was downloaded from the publicly available dataset on GitHub (<https://github.com/bkainz/fetalReconstruction.git>).

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Authors' Contributions

Qian Ni and Yi Zhang contributed equally to this work and should be considered co-first authors.

### Acknowledgments

This study was financed partially by the National Key R&D Program of China (No. 2018YFA0704102), the National Natural Science Foundation of China (Nos. 81827805, 61401451), and the Shenzhen Key R&D Program (No. JCYJ20200109114812361).

### References

- [1] O. A. Glenn, "Normal development of the fetal brain by MRI," *Seminars in Perinatology*, vol. 33, no. 4, pp. 208–219, 2009.
- [2] D. Bulas and A. Egloff, "Benefits and risks of MRI in pregnancy," *Seminars in Perinatology*, vol. 37, no. 5, pp. 301–304, 2013.
- [3] S. C. O'Connor, V. J. Rooks, and A. B. Smith, "Magnetic resonance imaging of the fetal central nervous system, head, neck, and chest," *Seminars in Ultrasound, CT and MRI*, vol. 33, no. 1, pp. 86–101, 2012.
- [4] L. M. Tee, E. Y. Kan, J. C. Cheung, and W. Leung, "Magnetic resonance imaging of the fetal brain," *Hong Kong Medical Journal*, vol. 22, no. 3, pp. 270–278, 2016.
- [5] C. Limperopoulos and C. Clouchoux, "Advancing fetal brain MRI: targets for the future," *Seminars in Perinatology*, vol. 33, no. 4, pp. 289–298, 2009.
- [6] M. A. Rutherford, "Magnetic resonance imaging of the fetal brain," *Current Opinion in Obstetrics & Gynecology*, vol. 21, no. 2, pp. 180–186, 2009.
- [7] D. Prayer, G. Kasprian, E. Krampfl et al., "MRI of normal fetal brain development," *European Journal of Radiology*, vol. 57, no. 2, pp. 199–216, 2006.
- [8] N. Girard, C. Raybaud, D. Gambarelli, and D. Figarella-Branger, "Fetal brain MR imaging," *Magnetic Resonance Imaging Clinics of North America*, vol. 9, no. 1, pp. 19–56, vii, 2001.
- [9] V. Merzoug, S. Ferey, C. Andre, A. Gelot, and C. Adamsbaum, "Magnetic resonance imaging of the fetal brain," *Journal of Neuroradiology*, vol. 29, no. 2, pp. 76–90, 2002.
- [10] S. N. Saleem, "Fetal magnetic resonance imaging (MRI): a tool for a better understanding of normal and abnormal brain development," *Journal of Child Neurology*, vol. 28, no. 7, pp. 890–908, 2013.
- [11] C. Malamateniou, S. J. Malik, J. M. Allsop et al., "Motion-compensation techniques in neonatal and fetal MR imaging," *American Journal of Neuroradiology*, vol. 34, no. 6, pp. 1124–1136, 2013.
- [12] D. Prayer, P. C. Brugger, and L. Prayer, "Fetal MRI: techniques and protocols," *Pediatric Radiology*, vol. 34, no. 9, pp. 685–693, 2004.
- [13] F. Rousseau, O. Glenn, B. Iordanova et al., "A novel approach to high resolution fetal brain MR imaging," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 548–555, Springer-Verlag, 2005.
- [14] S. Z. Jiang, H. Xue, A. Glover, M. Rutherford, D. Rueckert, and J. V. Hajnal, "MRI of moving subjects using multislice snapshot images with volume reconstruction (SVR): application to fetal, neonatal, and adult brain studies," *IEEE Transactions on Medical Imaging*, vol. 26, no. 7, pp. 967–980, 2007.
- [15] K. Kim, P. Habas, F. Rousseau, O. Glenn, A. Barkovich, and C. Studholme, "Intersection based motion correction of multislice MRI for 3-D in utero fetal brain image formation," *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 146–158, 2010.
- [16] B. Kainz, M. Steinberger, W. Wein et al., "Fast volume reconstruction from motion corrupted stacks of 2D slices," *IEEE Transactions on Medical Imaging*, vol. 34, no. 9, pp. 1901–1913, 2015.
- [17] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, 2012.
- [18] F. Rousseau, A. Glenn, B. Iordanova et al., "Registration-based approach for reconstruction of high-resolution in utero fetal MR brain images," *Academic Radiology*, vol. 13, no. 9, pp. 1072–1081, 2006.
- [19] A. Bertelsen, P. Aljabar, H. Xue et al., "Improved slice to volume reconstruction of the fetal brain for automated cortex segmentation," in *Proceedings of the International Society for*

- Magnetic Resonance in Medicine*, p. 3437, Honolulu, Hawai'i, 2009.
- [20] H. Greenspan, *Super-resolution in medical imaging*, vol. 52, no. 1, 2009 Oxford University Press, 2009.
- [21] P. Milanfar, *Super-Resolution Imaging*, CRC Press, 2007.
- [22] A. Gholipour and S. K. Warfield, "Super-resolution reconstruction of fetal brain MRI," in *MICCAI Workshop on Image Analysis for the Developing Brain(IADB)*, London, UK, 2009.
- [23] A. Gholipour, J. A. Estroff, and S. K. Warfield, "Robust super-resolution volume reconstruction from slice acquisitions: application to fetal brain MRI," *IEEE Transactions on Medical Imaging*, vol. 29, no. 10, pp. 1739–1758, 2010.
- [24] P. J. Huber and E. M. Ronchetti, *Robust Statistics*, Wiley, 2nd edition, 2009.
- [25] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 298–311, 1997.
- [26] G. Peyre, "A review of adaptive image representations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 896–911, 2011.
- [27] F. Rousseau, K. Kim, C. Studholme, M. Koob, and J.-L. Dietemann, "On super-resolution for fetal brain MRI," in *Medical Image Computing and Computer Assisted Intervention*, pp. 355–362, Springer-Verlag, 2010.
- [28] M. Kuklisova-murgasova, G. Quaghebeur, M. A. Rutherford, J. V. Hajnal, and J. A. Schnabel, "Reconstruction of fetal brain MRI with intensity matching and complete outlier removal," *Medical Image Analysis*, vol. 16, no. 8, pp. 1550–1564, 2012.
- [29] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, 2007.
- [30] K. V. Leemput, F. Maes, D. Vandermeulen, and P. Suetens, "Automated model-based bias field correction of MR images of the brain," *IEEE Transactions on Medical Imaging*, vol. 18, no. 10, pp. 885–896, 1999.
- [31] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.
- [32] T. X. Wen, L. Li, Q. S. Zhu et al., "GPU-accelerated kernel regression reconstruction for freehand 3D ultrasound imaging," *Ultrasonic Imaging*, vol. 39, no. 4, pp. 240–259, 2017.
- [33] H. Takeda, P. Milanfar, M. Protter, and M. Elad, "Super-resolution without explicit subpixel motion estimation," *IEEE Transactions on Image Processing*, vol. 18, no. 9, pp. 1958–1975, 2009.
- [34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.