Sinogram Upsampling via Sub-Riemannian Diffusion with Adaptive Weighting

JaKeoung Koo

School of Computing, Gachon University, 1342 Seongnam-daero, Seongnam-si 13120, Republic of Korea; jakeoung@gachon.ac.kr

Abstract: Computed tomography is a versatile imaging technique used to enable seeing internal structures of objects without opening or destroying them. This is possible through a process called tomographic reconstruction, which reconstructs images from projections of the object that are obtained by penetrating the object with beams of radiation, such as X-rays, from different angles. These projection data are often limited to low-resolution data in terms of projection angles. These limited or subsampled data make it difficult to obtain high-quality reconstruction results. Hence, upsampling projection data is necessary. In this paper, we propose a sinogram upsampling method via the sub-Riemannian diffusion process. We first lift the data into a feature space, and we fill in the missing angle parts by propagating information from the observed data to the missing parts. We observe that the sinogram with limited angle data has high directional dependency, and based on this observation, we suggest an adaptive weighting scheme to keep information propagating toward the missing regions. This adaptive weighting allows for diffusing toward the desired directions. The experimental results show the effectiveness of the proposed method in some scenarios regarding inpainting fine details, when compared to the existing model-based methods, such as Plug-and-Play and total generalized variation.

Keywords: computed tomography; tomographic reconstruction; sinogram upsampling; sinogram inpainting; sub-Riemannian diffusion

1. Introduction

Computed tomography (CT) is a computational imaging technique that uses radiation such as in X-rays to form cross-sectional images of the objects [1]. CT techniques are used in many applications, including industry, medical imaging and diagnosis [2–5]. The basic principle of CT is to measure the attenuation of radiation as it passes through the object, and the attenuation depends on the density of the object. By measuring the attenuation at multiple angles, it is possible to reconstruct the internal image of the object.

One of the challenges of CT imaging is that the sinogram, the raw data used to create the image, is often undersampled. This means that there are not enough measurements to accurately reconstruct the image. Sinogram upsampling is a technique that can be used to increase the resolution of the sinogram, which can lead to improved image quality. This is especially important for images of small objects or objects with complex shapes. Sinogram upsampling can also increase the contrast of CT images, which can make it easier to see subtle differences in the reconstructed images.

The sinogram upsampling problem can be considered as image inpainting because both problems involve filling in missing or corrupted data. In the case of sinogram upsampling, the missing data are the undersampled projection data, while in the case of image inpainting, the missing data are the pixels that have been corrupted or removed.

Among the image inpainting methods, there have been several works using sub-Riemannian diffusion, inspired by biological models of the primary visual cortex [6]. The primary visual cortex is a part of the brain that is responsible for processing visual information. It is
organized into a hierarchy of layers, with each layer responding to different features of the visual image. The neurons of the primary visual cortex are organized into a hypercolumn structure, which allows V1 to process information from the entire visual field in a systematic and efficient way.

Sub-Riemannian diffusion methods for image inpainting lift an image into a feature space and propagate information from the observed pixels to the missing pixels [7–9]. The lifting process can be performed by, e.g., feature extractions from a Gabor transform [7] with different orientations. The information is propagated along a set of paths that are defined by the sub-Riemannian geometry of the primary visual cortex. The sub-Riemannian geometry is a way of measuring the distance between points in the visual space.

In this paper, we propose a sinogram upsampling method, inspired by computational models of the visual cortex. We lift the sinogram data to the Lie group of the rotation-translation SE(2) and fill in the missing projection values using the sub-Riemannian diffusion process. Then, the upsampled sinogram is obtained by projecting the lifted sinogram onto the original sinogram space. We adopt an adaptive weighting scheme during the diffusion process to account for the fact that the missing angles lead to vertical missing lines. In summary, the contributions of this paper are the following:

- We formulate the sinogram upsampling problem as a sub-Riemannian diffusion process.
- We propose an adaptive weighting scheme to exploit the characteristics of sinogram upsampling problems with respect to projection angles.
- The experimental results verify that the proposed method is effective for upsampling a sinogram with respect to projection angles, compared to some model-based methods.

This paper is organized as follows. In Section 2, we describe the related work on sinogram inpainting and sub-Riemannian diffusion for image processing. Section 3 explains the proposed method, and Section 4 demonstrates the numerical experiments and results with different scenarios. Section 5 discusses and concludes the paper.

2. Related Works

2.1. Sinogram Inpainting

A straightforward approach for sinogram inpainting is to regard a sinogram as an image and apply the image inpainting algorithm. This approach has been adopted in several works via variational methods [10] or interpolation [11].

Another approach is a data-driven inpainting approach in a supervised or unsupervised way. Lee et al. [12,13] employed convolutional neural networks patchwise to refine an initial guess obtained from interpolation. These methods belong to the category of supervised learning. Recently, there have been many attempts using unsupervised learning. Generative Adversarial Networks (GAN) [14] were adopted in [15]. To fill in large missing regions, an auto-encoder type called the context encoder was proposed for image inpainting [16] and this context encoder was used in [17] for sinogram inpainting. Valat et al. [18] incorporated the shape prior information into the GAN framework. One disadvantage of using a GAN is that it requires large computational resources and so it is difficult to scale to large data. Unlike the data-driven approach, our method does not require training data, often providing a more fast and reliable solution with greater interpretability than the data-driven approaches.

2.2. Image Processing via Sub-Riemannian Diffusion

Our method adopts the sub-Riemannian diffusion used in the image completion task, so we review such work. Sub-Riemannian diffusion is a type of diffusion process that is defined on a sub-Riemannian manifold. A sub-Riemannian manifold [19] is a manifold where the tangent space at each point is equipped with a non-degenerate inner product, but not all vectors in the tangent space are allowed to move [20]. The allowed vectors form a distribution, which is a subset of the tangent space. In 1989, Hoffman [21] considered the hypercolumn structure of V1 as a contact-bundle structure. It was developed further by several works, including [7,22], which proposed a mathematical model of the structure of V1 as the group of rotations and translations SE(2) and the receptive profiles as Gabor
functions. This line of model is often called the Citti–Petitot–Sarti model, following the author names. This novel framework is extended to considering more features, such as multiscale or multifrequency [23]. Based on such a mathematical model of the visual cortex, sub-Riemannian diffusion is employed for image completion problems [24–28] and for explaining visual illusions [29–31]. Unlike the existing sub-Riemannian diffusion work, we suggest an adaptive weighting scheme that is direction-dependent, which is suitable for sinogram inpainting.

3. Method

In this section, we propose an algorithm to upsample the sinogram with respect to projection angles. Before introducing the proposed method, we review the data acquisition process in CT imaging. In CT, the data acquisition process can be modeled as a process called Radon transform. The Radon transform of an image \( I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R} \) maps the image into a measurement space, defined as:

\[
p(r,s) = R[I] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x,y) \delta(x \cos r + y \sin r - s) \, dx \, dy
\]

where \( s \) represents the position of the detector and \( r \) is the rotation angle. This measurement is called projection and the projection value \( p(r,s) \) represents the line integral of \( I \) along the line: \( x \cos r + y \sin r - s \).

From this setting, sinogram upsampling aims to upsample signals with respect to projection angles \( r \). For this problem, the proposed method consists of three stages: the lifting stage, diffusion stage and projection stage, summarized in Algorithm 1.

Algorithm 1 Sinogram upsampling method

1: Input: Sinogram data \( p \), projection angles \( M \) to be upsampled
2: Lifting the sinogram to obtain \( O \) (Section 3.1)
3: Sub-Riemannian diffusion (Section 3.2)
4: Projection of the lifted sinogram to the original space (Section 3.3)
5: Output: Upsampled sinogram

3.1. Lifting

We lift the sinogram data to the lifting space \( M = \mathbb{R}^2 \times [0,2\pi) \) by extracting the orientation features by Gabor filters. The Gabor function \( G \) is defined as

\[
G(r,s,\theta) := \frac{1}{2\sigma^2} e^{-i(f(-r \sin \theta + s \cos \theta) - \phi)} e^{-\frac{r^2+s^2}{2\sigma^2}}
\]

where \( \phi \) is a phase component, \( f \) is the spatial frequency and \( \sigma \) is the scaling parameter. In the experiments, we set \( \phi \) to 0. The lifted sinogram \( F \) is obtained by the convolution of \( G \) and the sinogram data \( p \) as follows:

\[
F(r,s,\theta) = \int G(r-r',s-s',\theta)p(r',s') \, dr' \, ds'.
\]

3.2. Sub-Riemannian Diffusion

From the lifted sinogram, the goal is two-fold: (i) we propagate the information from the observed projections to the missing projections in the unknown projection angles; (ii) we denoise the lifted sinogram in an anisotropic way. To achieve this goal, we employ the sub-Riemannian diffusion process, where smoothing can be imposed in some directions more than others.

To consider connections in the geometry \( M \) of the lifting space, sub-Riemannian geometry considers a subbundle called the horizontal vector fields \( HM \) of the tangent bundle \( TM \). A subbundle of a tangent bundle is a proper subset of the tangent bundle,
which is itself a vector bundle. In our rotation–translation lifting space \( M \), we consider the following horizontal vector fields:

\[
X_1 = \cos \theta \partial_r + \sin \theta \partial_s, \\
X_2 = \partial \theta.
\]  

(4)  

(5)  

Note that for each point \( p \in M \),

\[
\text{span}\{X_1, X_2\}|_p \subset T_pM,
\]  

(6)  

but the horizontal vector fields have the bracket-generating property such that

\[
\text{span}\{X_1, X_2, [X_1, X_2]\}|_p = T_pM,
\]  

(7)  

where \([X_1, X_2] = \sin \theta \partial_r - \cos \theta \partial_s\).

From the horizontal vector fields, we can take the corresponding integral curves. An integral curve of a vector field \( X \) is a curve \( \gamma : [0, 1] \rightarrow M \) such that \( \gamma'(t) = V_{\gamma(t)} \) for all \( t \in I \). Integral curves of horizontal vector fields are called horizontal integral curves, and following these curves, we can fill in the missing parts of the sinogram. Citti and Sarti [7] consider the \( \text{SE}(2) \) sub-Riemannian diffusion operator \( X_1^2 + \beta X_2^2 \).

3.2.1. Adaptive Weighting

In this paper, we focus on the sinogram upsampling with respect to projection angles. We observe that the information should be less propagated from the vertical directions on the missing regions. With this motivation, we suggest using the diffusion operator with adaptive weighting, as follows:

\[
\mathcal{L} = a(r, s, \theta)X_1^2|_{(r, s, \theta)} + \beta X_2^2|_{(r, s, \theta)}
\]  

(8)  

where we choose the weighting parameter \( a \) as

\[
a(r, s, \theta) = 1_A(r, s, \theta) \exp(-a_0 \sin^2 \theta) + (1 - 1_A(r, s, \theta)),
\]  

(9)  

where \( A \) is the set of missing positions in the sinogram and \( 1_A(r, s, \theta) \) is the indicator function, defined as

\[
1_A(x) = \begin{cases} 
1 & \text{if } x \in A \\
0 & \text{if } x \notin A 
\end{cases}
\]  

(10)  

The motivation of this choice is to diffuse less along the vertical directions on the missing positions. Another weighting parameter \( \beta \) is chosen as \( T/(RS) \), where \( T \) is the number of lifting angles \( \theta \)'s, \( R \) is the number of projection angles \( r \) and \( S \) is the number of detector pixels \( s \).

From the sub-Riemannian diffusion operator, we update the lifted sinogram \( P \) as follows:

\[
\partial_t P = \mathcal{L} P
\]  

(11)  

where \( t \) denotes the time variable.

3.2.2. Discretization

To discretize the diffusion process (11), we consider its discrete version:

\[
p^{t+1} = p^t + (\Delta t)\mathcal{L}p^t
\]  

(12)  

where \( \Delta t \) is the step size. We also need to discretize the operators \( X_1^2 \) and \( X_2^2 \). We discretize the lifted space, and for simplicity, the intervals \( \Delta r \) and \( \Delta s \) are set to 1. For the Gabor filter angle \( \theta \), we introduce the index \( i \) such that we are given \( \{\theta_1, \theta_2, \ldots, \theta_i, \ldots, \theta_T\} \), with \( T \) the
number of Gabor filter angles during the lifting. From this setting, we adopt a second-order centered difference approximation of derivatives as follows:

\[ X_1^2P = P(r + \cos \theta_i, s + \sin \theta_i, \theta_i) - 2P(r, s, \theta_i) + P(r - \cos \theta_i, s - \sin \theta_i, \theta_i) \]  
\[ X_2^2P = \delta(P(r, s, \theta_{i+1}) - 2P(r, s, \theta_i) + P(r, s, \theta_{i-1})) \]  

where we consider \( \delta = 1 \) in our experiments for numerical stability when considering a large number of angles in the Gabor filters. Note that the values of \( P \) are available on the uniform grid. Hence, to evaluate \( P(r + \cos \theta_i, s + \sin \theta_i, \theta_i) \) and \( P(r - \cos \theta_i, s - \sin \theta_i, \theta_i) \), we need to interpolate, and we use B-spline interpolation [33] from the values on the uniform grid.

3.3. Projection

After finishing the diffusion process, we need to project back the lifted sinogram to the original sinogram space. We adopt the summation projection as follows:

\[ p(r, s) = \left\| \sum_i P(r, s, \theta_i) \right\|, \]  

where the absolute operator is taken to convert complex values to real values. An alternative way for lifting and projection is to use cake wavelets [34,35], which are invertible. However, we observe that due to the dependency on Fourier transform, cake wavelets have some numerical issues when we have vertical missing regions.

4. Results

In this section, we perform the numerical experiments to test the proposed method.

4.1. Test Dataset

We consider three phantom images, as shown in the first row of Figure 1. From the phantom images, we generate ground-truth sinograms by Radon transform (1) with 180 projection angles and 363 detector pixels, visualized in the third row. From these ground-truth sinograms, we remove 44, 88 or 132 projection angles and impose Gaussian noise with variance, 0.0001. The goal of the experiments is to upsample and denoise the undersampled noisy sinogram data.

![Phantom 1](image1.png) ![Phantom 2](image2.png) ![Phantom 3](image3.png)

![Sinogram 1](image4.png) ![Sonogram 2](image5.png) ![Sinogram 3](image6.png)

**Figure 1.** The first row images show the phantom images used in the experiments to generate the ground-truth sinograms in the second row.
4.2. Comparison Methods

In imaging applications, many problems can be formulated as optimizing a cost function, which consists of a data fidelity term and regularization term. To compare our method, we formulate the sinogram upsampling problem as a variational imaging problem as follows:

$$\min_p \| p_{data} - A(p) \|^2_2 + \lambda \text{Reg}(p)$$  \hspace{1cm} (16)

where $A$ is the undersampling operator to map an upsampled sinogram $p$ to the original data space. In Equation (16), the first term is the data fidelity term, the second term Reg is the regularization term and the parameter $\lambda$ controls the trade-off between the two terms.

**Plug-and-Play (PnP) with median filtering.** Some first-order optimization algorithms [36] alternatively optimize the two terms. When optimizing (16), minimizing the regularization term often corresponds to a denoising problem. To take advantage of the state-of-the-art denoising methods, the Plug-and-Play (PnP) scheme [37] provides a flexible framework to replace the denoising step with sophisticated denoising schemes. In our experiments, we use median filtering as a denoiser. We use Half-Quadratic Splitting [38] as an optimizer.

**Total generalized variation (TGV).** One of the most popular image priors is total variation regularization [39], which was generalized by Bredies et al. [40] in 2010 as total generalized variation (TGV). Specifically, TGV aims to optimize

$$\min_p \| p_{data} - A(p) \|^2_2 + \lambda_1 \| r \|_{1,2} + \lambda_2 \| J(\nabla p - r) \|_{1,F}$$  \hspace{1cm} (17)

where $r$ is an auxiliary variable, $\nabla$ is the discrete gradient operator, $J$ is the Jacobian operator and $\| \cdot \|_{1,F}$. We use the Alternating Direction Method of Multipliers (ADMM) [41] as an optimizer. We use the deepinv library [42] for the implementation of PnP and TGV.

4.3. Experimental Results

We use the same parameters throughout the numerical experiments as follows: the step size is $\Delta t = 0.5$, the weighting hyperparameter is $\alpha_0 = 10$ and the number of iterations is 200.

In the first experiment, we removed 44 projection angles, and the goal was to upsample 44 projection angles. Figure 2 shows the convergence behavior for Sinograms 1, 2 and 3 during the iteration. Figure 3 shows the qualitative upsampling results from the noisy sinogram data from the different methods. The first row shows the results for Sinogram 1. The PnP method misses the top and bottom boundaries of the sinogram. Overall, the proposed method estimates fine details well, but some artifacts are observed near the boundary of the sinogram. The second row shows the results for Sinogram 2 and the proposed method can also estimate fine structures well and shows fewer artifacts near the sinogram border, compared to that of Sinogram 1. This is because the contrast is small in Sinogram 1, but in Sinogram 2, due to the large contrast, we observe some blurring effects in our results.

![Figure 2](image_url)

Figure 2. Convergence behavior with respect to the metric PSNR on data from three sinograms. The final PSNR values are 33.591, 34.933 and 42.215 for Sinogram 1, 2 and 3, respectively.
Figure 3. Qualitative results to upsample 63 missing angles from noisy sinogram data. The first column shows the input noisy sinogram. The 2nd–4th columns show the upsampled sinogram by Plug-and-Play with median filtering, total generalized variation [40] and the proposed method. The rows are organized by Sinograms 1, 2 and 3.
Figure 4 presents the results of upsampling 88 projection angles from noisy sinograms. PnP with median filters yields some artifacts due to large missing areas. TGV produces typically blurry results, while the proposed method produces overall good results.

As shown in Figure 5, the numerical results are demonstrated to upsample 132 projection angles from noisy sinograms. PnP with median filters shows severe artifacts on the right part due to large missing areas. TGV again produces blurry results to fill in the large missing areas. On the other hand, the proposed method overall captures fine details better than TGV. Figure 6 shows the reconstructed images from the upsampled sinograms in Figure 5. It is clear from the reconstructed images that the the proposed method reconstructs fine details well with better contrasts, which is a distinct advantage of the proposed method.

Table 1 provides the quantitative comparison in terms of the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM) [43]. Interestingly, PnP with median filtering often gives the best results in terms of the SSIM but shows poor performance in terms of the PSNR. For Sinogram 2, the proposed method shows the best performance. For Sinogram 1, TGV presents the best performance in terms of the PSNR, compared to the proposed method.

Table 1. Quantitative comparison of sinogram upsampling methods. The best results are in bold.

<table>
<thead>
<tr>
<th>Sinogram</th>
<th>Upsampled Angles</th>
<th>Metric</th>
<th>PnP-Median</th>
<th>TGV</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sinogram 1</td>
<td>44</td>
<td>PSNR</td>
<td>32.588</td>
<td>39.615</td>
<td>33.591</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSIM</td>
<td>0.972</td>
<td>0.935</td>
<td>0.907</td>
</tr>
<tr>
<td>Sinogram 2</td>
<td>44</td>
<td>PSNR</td>
<td>21.159</td>
<td>27.249</td>
<td>34.933</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSIM</td>
<td>0.913</td>
<td>0.899</td>
<td>0.917</td>
</tr>
<tr>
<td>Sinogram 3</td>
<td>44</td>
<td>PSNR</td>
<td>36.573</td>
<td>38.818</td>
<td>42.215</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSIM</td>
<td>0.974</td>
<td>0.919</td>
<td>0.930</td>
</tr>
<tr>
<td>Sinogram 1</td>
<td>88</td>
<td>PSNR</td>
<td>30.654</td>
<td>36.982</td>
<td>29.759</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSIM</td>
<td>0.944</td>
<td>0.927</td>
<td>0.909</td>
</tr>
<tr>
<td>Sinogram 2</td>
<td>88</td>
<td>PSNR</td>
<td>19.833</td>
<td>27.006</td>
<td>30.424</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSIM</td>
<td>0.831</td>
<td>0.881</td>
<td>0.9123</td>
</tr>
<tr>
<td>Sinogram 3</td>
<td>88</td>
<td>PSNR</td>
<td>31.8</td>
<td>36.275</td>
<td>40.914</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSIM</td>
<td>0.934</td>
<td>0.911</td>
<td>0.929</td>
</tr>
<tr>
<td>Sinogram 1</td>
<td>132</td>
<td>PSNR</td>
<td>11.888</td>
<td>32.768</td>
<td>24.220</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSIM</td>
<td>0.784</td>
<td>0.911</td>
<td>0.901</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSIM</td>
<td>0.621</td>
<td>0.843</td>
<td>0.894</td>
</tr>
<tr>
<td>Sinogram 3</td>
<td>132</td>
<td>PSNR</td>
<td>13.330</td>
<td>32.742</td>
<td>38.190</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSIM</td>
<td>0.708</td>
<td>0.891</td>
<td>0.922</td>
</tr>
</tbody>
</table>

Figure 7 shows the quantitative results with respect to the different projection angles available. Specifically, the sinogram data are available on every 4, 5 or 6 angles among the 180 angles. Figure 8 shows the corresponding reconstruction images. In particular, the proposed method is shown to capture small details in the bottom of the objects.

Table 2 provides the computational efficiency between the proposed method and other methods. We use a Linux server containing an NVIDIA GeForce RTX 4090 card. This GPU is used for PnP-Median and TGV. For the proposed method, we use a CPU for the lifting stage and a GPU for the diffusion and projection steps. Hence, there is room for improving the computational speed by parallelizing the lifting step.

Table 2. Comparison of computation time when upsampling 88 angles. The computation time in the proposed method is divided into three parts of lifting, diffusion and projection.

<table>
<thead>
<tr>
<th></th>
<th>PnP-Median</th>
<th>TGV</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computational time (in seconds)</td>
<td>0.26</td>
<td>27.82</td>
<td>3.11 = 1.3 + 1.7 + 0.11</td>
</tr>
</tbody>
</table>
Figure 4. Qualitative results to upsample 88 missing angles from noisy sinogram data. The first column shows the input noisy sinogram. The 2nd–4th columns show the upsampled sinogram by Plug-and-Play with median filtering, total generalized variation [40] and the proposed method. The rows are organized by Sinograms 1, 2 and 3.
Figure 5. Qualitative results to upsample 132 missing angles from noisy sinogram data. The first column shows the input noisy sinogram. The 2nd–4th columns show the upsampled sinogram by Plug-and-Play with median filter, total generalized variation [40] and the proposed method. The rows are organized by Sinograms 1, 2 and 3.
Figure 6. Reconstruction results after upsampling 132 missing angles from noisy sinogram data. The first column shows the input noisy sinogram. The 2nd–4th columns show the reconstructed images after upsampling by Plug-and-Play with median filter, total generalized variation [40] and the proposed method. The rows are organized by Sinograms 1, 2 and 3.

Figure 7. Quantitative results with respect to different projection angles available, where the data are only available on every 4, 5 or 6 angles (x-axis). The PSNR number is evaluated between the reconstructed images and Phantom 1.
Figure 8. Qualitative results with respect to different projection angles available, where the data are only available on every 4, 5 or 6 angles (top to bottom). Reconstruction results are obtained after upsampling by PnP-Median, TGV and the proposed method (left to right).

Lastly, we test the proposed method on the Walnut dataset provided by [44]. This object is scanned by microCT with an exposure time of 1000 ms, an acceleration voltage of 40 kV and a current of 1 mA in the X-ray tube. The sinogram data have 180 projection angles. We preprocess the sinogram data for converting the fan-beam geometry to parallel-beam geometry, because the proposed method is currently developed for parallel-beam geometry. We further normalize the range of data lying in between 0 and 1 and extract 256 detector pixels. Because they are real data, there is no exact ground truth, but a reference image is provided from [44]. Figure 9 shows the reconstruction results after upsampling 44 missing angles. The first column shows the reference image. The 2nd, 3rd and 4th columns show the reconstruction results after upsampling via Plug-and-Play with the median filter, total generalized variation and the proposed method, respectively.

Figure 9. Reconstruction results after upsampling 44 missing angles from Walnut sinogram data [44]. The first column shows the reference image. The 2nd-4th columns show the reconstructed images after upsampling by Plug-and-Play with median filter, total generalized variation (TGV) and the proposed method.
5. Discussion and Conclusions

In this paper, we propose a sinogram upsampling method via sub-Riemannian diffusion. The proposed method first lifts the sinogram into a rotation-and-translation space. The lifted sinogram has many missing parts that are to be filled in by the diffusion process on the lifted space. After the diffusion process, it is projected back to the original sinogram space. In the diffusion process, we adopt a direction-dependent weighting scheme to keep noisy information propagating toward the missing regions. The numerical experiments on the simulated phantom and Walnut dataset [44] show the effectiveness of the proposed method, compared to the existing Plug-and-Play and total generalized variation work. The proposed method can also be applied to sinogram inpainting with respect to detector pixels, as well as projection angles. In this case, we can consider another adaptive weighting procedure and we leave it as future work.

A drawback of the proposed method is that lifting and projection are not invertible. For example, a clever transformation known as the cake wavelets lift [34,35] can be used to give invertible procedures. The idea of cake wavelets is to divide the frequency space into small parts, where each part can refer to some orientations. To these small parts, the inverse Fourier transforms are applied, and the real parts can serve as line detectors and the imaginary parts can serve as edge detectors with a specific orientation. However, due to the dependency on fast Fourier transform, applying that lift to sinogram upsampling with vertical missing regions yields some numerical artifacts. We leave it as future work to make lifting and projection invertible with numerical stability.

Further future work would be to extend the proposed method to sinogram inpainting. In this paper, we deal with sinogram data where projection angles are available for at least every 2 or up to 6 degrees. It has many applications, such as diagnosis or security applications. Another important problem, the so-called limited angle problem, considers missing angles in a specific range, such as $-20$ degrees to 20 degrees. The proposed method is not designed to deal with such limited angle data, as there is no information to propagate from the limited range. However, the data-driven approach using deep learning can be combined with the proposed method to tackle this problem. We leave it as future work.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2023-00240740) and was also supported by the Gachon University research fund of 2023 (GCU-20230660001).

Data Availability Statement: The original phantom images from Figure 1 can be download from this paper and the sinogram data can be generated using the Scikit-image library 0.21.0 [45].

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations
The following abbreviations are used in this manuscript:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT</td>
<td>Computed tomography</td>
</tr>
<tr>
<td>V1</td>
<td>Primary visual cortex</td>
</tr>
<tr>
<td>PnP</td>
<td>Plug-and-Play</td>
</tr>
<tr>
<td>TGV</td>
<td>Total generalized variation</td>
</tr>
</tbody>
</table>

References

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.