

实验六 基于压缩感知的医学图像重建实验

一、 实验目的

- 利用压缩感知算法对医学图像进行重建，理解压缩感知理论在图像重建中的应用原理，掌握基于压缩感知的图像重建算法的实现过程。

二、 实验原理

- 传统医学图像数据采集过程较为缓慢。压缩感知理论指出，对于具有稀疏性或可压缩性的信号，可以通过远低于奈奎斯特采样率的方式进行采样，并利用信号的稀疏特性从少量的测量数据中准确地重建出原始信号。
- 参考文献: Lustig M , Donoho D , Pauly J M .Sparse MRI: The application of compressed sensing for rapid MR imaging[J].Magnetic Resonance in Medicine, 2007, 58(6):1182-1195.DOI:10.1002/mrm.21391

三、 实验设备与环境

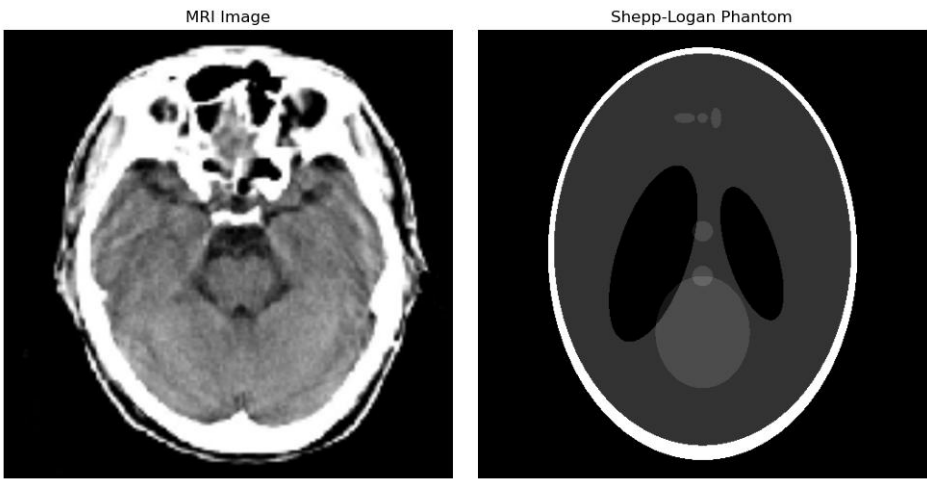
- 软件工具: MATLAB、Python (NumPy, SciPy, PyTorch 等)、或其他科学计算软件。
- 硬件要求: 一台性能良好的计算机，足够的内存和处理能力来运行模拟和重建算法。

四、 实验内容与步骤

- 1) 生成模拟物体模型 (10 分)
 - 选择一个合适的图像作为原始图像，可以从研学课上获取的大脑磁共振图像，也可以是 Shepp-Logan phantom。
 - 在实验中，我选择了第一节研学实验课的 hbh.BMP 和生成的 Shepp-Logan phantom 进行实验。代码及结果如下：

```
# 读取 CT 图像
ct_image = cv2.imread(r"hbh.BMP", cv2.IMREAD_GRAYSCALE)
if ct_image is None:
    raise FileNotFoundError("Image not found at specified path.")
ct_image = cv2.resize(ct_image, (512, 512)) # 确保 CT 图像大小为 512x512

# 生成 Shepp-Logan phantom
phantom = shepp_logan(512)
```



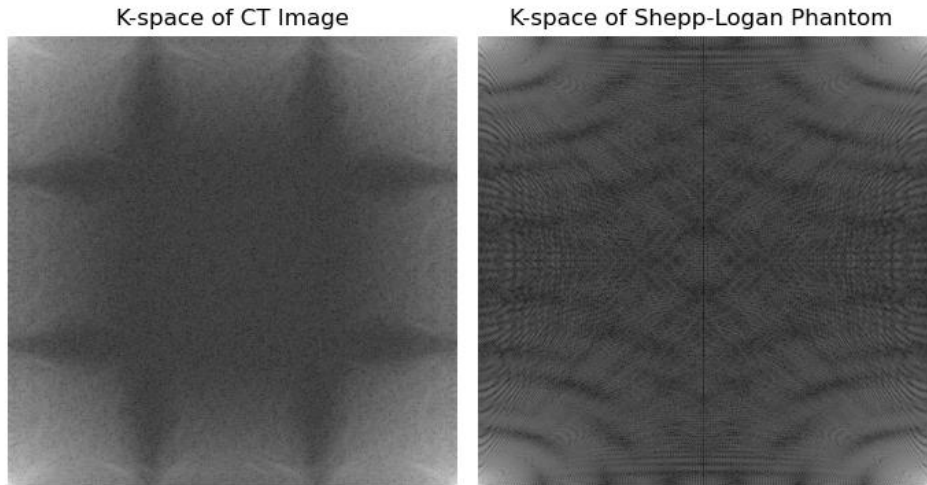
- 2) 模拟磁共振成像 k 空间数据获取 (10 分)
- 对原始图像进行二维离散傅里叶变换 (2D-FFT)，得到其在 k 空间的完整数据。

```

K-space data for CT Image:
[[16.98335449 15.87583946 15.42729743 ... 13.91126732 15.42729743
  15.87583946]
 [15.11709197 14.64135023 14.66330564 ... 14.71690766 14.65329298
  14.03551036]
 [15.10452367 14.23124619 14.61103452 ... 14.07072196 14.58478199
  14.59554817]
 ...
 [14.32257198 13.67768884 13.4410256 ... 14.13986623 13.56799421
  13.66137216]
 [15.10452367 14.59554817 14.58478199 ... 13.39770428 14.61103452
  14.23124619]
 [15.11709197 14.03551036 14.65329298 ... 14.67584397 14.66330564
  14.64135023]]

K-space data for Shepp-Logan Phantom:
[[10.38368591 9.50809367 8.51755382 ... 7.91467146 8.51755382
  9.50809367]
 [ 8.03457462 8.50362238 8.54481705 ... 8.31211774 8.49547255
  8.37369525]
 [ 8.4986277 7.83597691 7.94827893 ... 8.02343137 7.95376689
  7.72516232]
 ...
 [ 7.42907838 7.38118111 7.67005619 ... 7.92261192 7.48245758
  7.07495925]
 [ 8.4986277 7.72516232 7.95376689 ... 8.11636826 7.94827893
  7.83597691]
 [ 8.03457462 8.37369525 8.49547255 ... 8.42077113 8.54481705
  8.50362238]]

```



3) 设计稀疏采样模式（15 分）

- 设计一种稀疏采样模式，例如采用随机欠采样或均匀欠采样方式或高斯欠采样方式。确定至少三种欠采样率（如 25%、50%、75% 等），即从完整的 k 空间数据中随机选取一定比例的数据点，其余数据点置零，得到欠采样的 k 空间数据。

```
# 稀疏采样函数
def sparse_sampling(k_space, sampling_rate):
    total_points = k_space.size
    num_samples = int(total_points * sampling_rate)
    # 随机选择数据点
    indices = np.random.choice(total_points, num_samples, replace=False)
    sparse_k_space = np.zeros_like(k_space)
    sparse_k_space.flat[indices] = k_space.flat[indices]

    return sparse_k_space

# 设计稀疏采样模式
sampling_rates = [0.25, 0.5, 0.75] # 25%, 50%, 75%
sparse_k_spaces_ct = {rate: sparse_sampling(k_space_ct, rate) for rate in sampling_rates}
sparse_k_spaces_phantom = {rate: sparse_sampling(k_space_phantom, rate) for rate in sampling_rates}
```

4) 对图像进行稀疏变换（15 分）

- 选择一种稀疏变换方法——小波变换。对原始图像进行变换，得到其在变换域的系数表示。

```
def wavelet_transform(image):
    coeffs = pywt.wavedec2(image, 'haar', level=3)
    return coeffs

coeffs_ct = wavelet_transform(ct_image)
coeffs_phantom = wavelet_transform(phantom)
```

- 在变换域中，观察是否大部分系数接近零，如是，则该变换为稀疏变换。

```
# 检查系数的稀疏性
def check_sparsity(coeffs):
    # 低频系数
    low_freq = coeffs[0].flatten()
    # 高频系数
    high_freqs = np.concatenate([c.flatten() for c in coeffs[1:]]) # 高频系数是一个元组
    # 合并低频和高频系数
    all_coeffs = np.concatenate([low_freq, high_freqs])
    sparsity = np.mean(np.abs(all_coeffs) < 1e-5) # 观察是否大部分系数接近零
    # 打印稀疏性信息
    if sparsity > 0.5:
        print("该变换是稀疏变换.")
    else:
```

```
print("该变换不是稀疏变换.")
return sparsity
```

该变换不是稀疏变换.

该变换是稀疏变换.

Sparsity of CT image coefficients: 0.3430

Sparsity of Phantom image coefficients: 0.7946

由结果可知，MRI 图像并未完全进行稀疏变换，而 phantom 进行了稀疏变换。

5) 图像重建（30 分）

➤ 选择一种压缩感知图像重建算法，并对其实现。

➤ 以迭代收缩阈值算法（ISTA）为例：

- 第一步，初始化重建图像（可以是全零图像或其他简单的初始估计）：对 k 空间填零后进行傅里叶逆变换得到初始图像。

```
def initialize_reconstruction(sparse_k_space):
    if np.all(sparse_k_space == 0):
        raise ValueError("Input sparse k-space is empty.")
    return np.fft.ifft2(sparse_k_space).real
.....省略代码.....
reconstructed_image = initialize_reconstruction(sparse_k_space)
previous_image = reconstructed_image.copy()
```

- 在每次迭代中：计算当前重建图像的稀疏变换系数。

```
for i in range(max_iter):
    coeffs = pywt.wavedec2(reconstructed_image, 'haar', level=3)
```

- ◆ 根据数据一致性约束和稀疏性先验，更新重建图像的估计值。

```
# 数据一致性约束
k_space_reconstructed = np.fft.fft2(reconstructed_image) # 将重建
的图像转换到 k 空间
k_space_reconstructed[sparse_k_space == 0] = 0 # 保持欠采样 k 空间的
值
reconstructed_image = np.fft.ifft2(k_space_reconstructed).real #
反变换回图像
```

- ◆ 具体来说，通过求解一个优化问题来调整系数，使得重建图像的 k 空间数据与欠采样的 k 空间数据尽可能匹配，同时保持系数的稀疏性。

```
# 软阈值处理
coeffs = pywt.wavedec2(reconstructed_image, 'haar', level=3) # 计
算稀疏变换系数
coeffs_thresholded = list(coeffs)
coeffs_thresholded[0] = pywt.threshold(coeffs[0], lambda_reg,
mode='soft') # 对低频系数进行软阈值处理
coeffs_thresholded[1] = tuple(pywt.threshold(c, lambda_reg,
mode='soft') for c in coeffs[1]) # 对高频系数进行软阈值处理
reconstructed_image = pywt.waverec2(coeffs_thresholded, 'haar') #
用处理过的系数重建图像
```

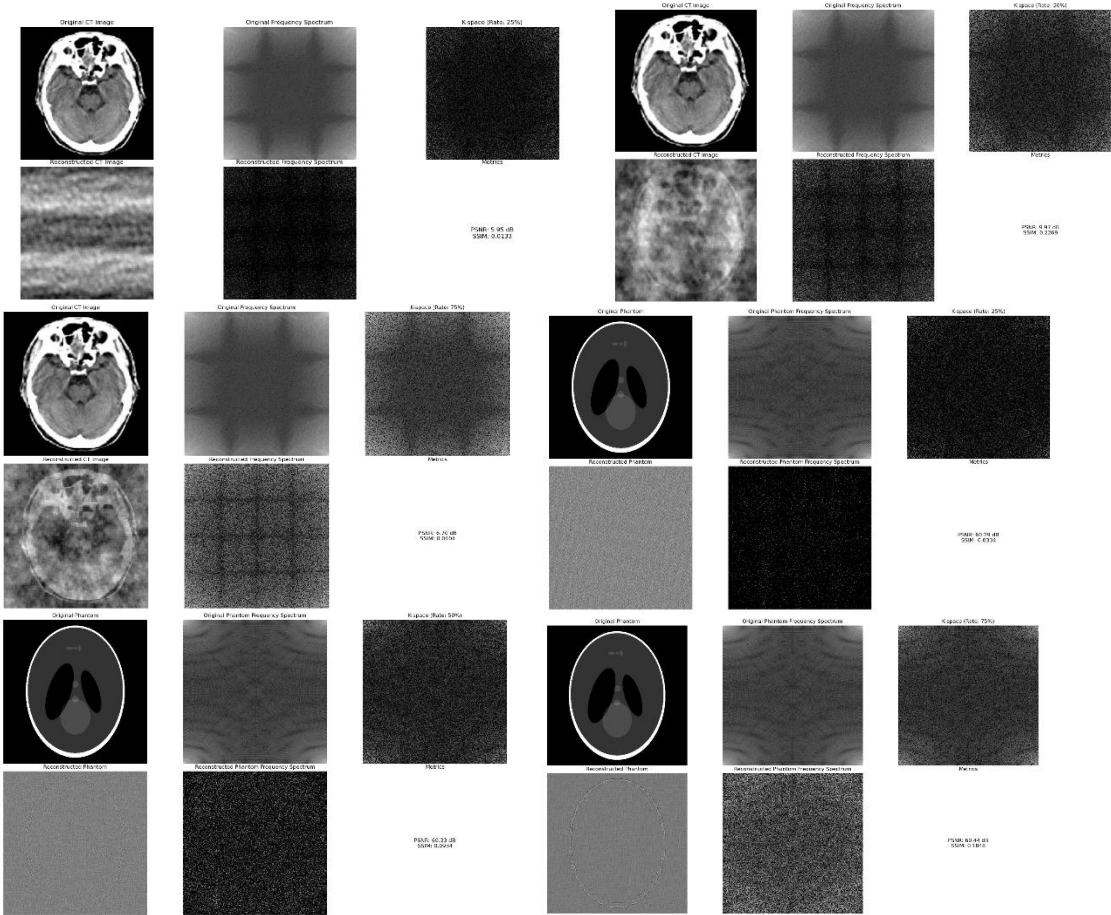
- ◆ 检查迭代终止条件，如达到最大迭代次数或重建图像的变化小于一定阈值。

```
# 检查终止条件
if i > 0 and np.linalg.norm(reconstructed_image -
previous_image) < tol:
    print(f'Converged after {i} iterations.')
    break
```

由于算法效果较差，并未提前终止，也未打印对应 iteration。如下图：

```
PS C:\Users\18148\Desktop\CT> & D:/anaconda/envs/myenv/python.exe c:/Users/18148/Desktop/CT/CT.py
Sparsity of CT image coefficients: 0.3430
Sparsity of Phantom image coefficients: 0.7946
```

- 6) 结果评估（20 分）
- 计算重建图像与原始图像之间的峰值信噪比（PSNR）和结构相似性指数（SSIM）等指标，以定量评估重建图像的质量。
 - 可视化原始图像、欠采样后的 k 空间数据、重建图像以及它们的频谱图等，直观地比较和分析重建效果。
 - 下图分别是 MRI 图像和 phantom 图像 25%、50%、75%欠采样率下对应图片。



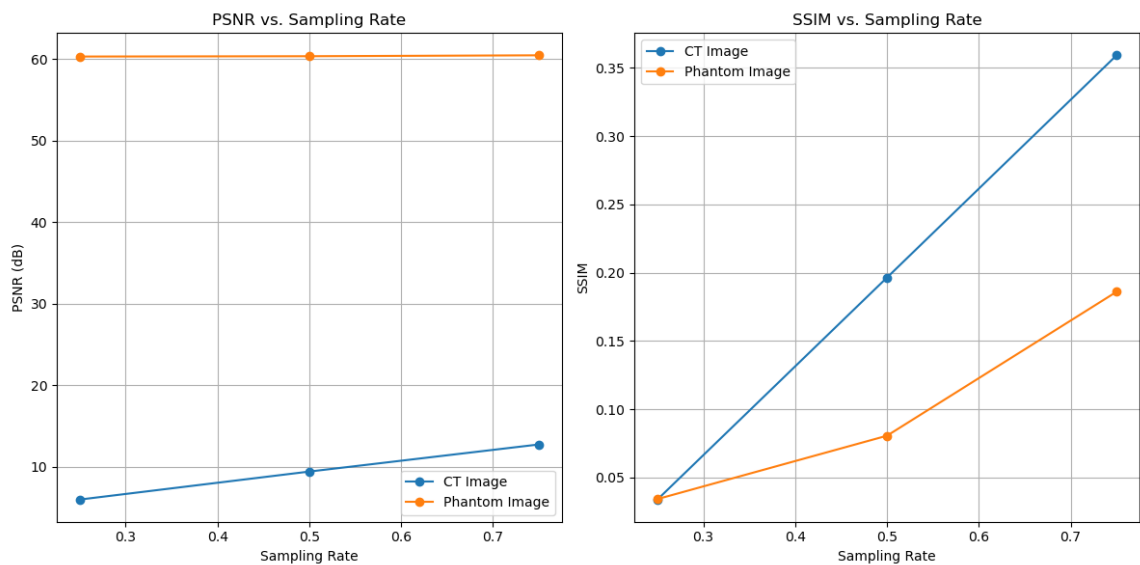
五、 源程序

见附件CT.py。

六、 实验收获与总结

➤ 分析不同稀疏采样率下重建图像的质量
答：MRI图像PSNR随着不同采样率从5.95dB到9.97Db,再到6.70Db, SSIM从0.0133到0.2269再到0.0404。phantom图像PSNR从60.29dB到60.33dB再到60.44Db, SSIM从0.0334到0.0934再到0.1848。显然，较低的采样率会导致图像质量显著下降。除此之外，MRI图像在不同采样率下的重建质量变化更为明显，PSNR和SSIM值都显示出较大的波动，表明其对采样率的敏感性较高。这可能是因为MRI图像通常包含更多复杂的组织结构和细节，而phantom图像则相对简单，容易被重建；而Phantom图像的PSNR值保持在一个较高的水平，说明它在不同采样率下的重建效果相对稳定，也反映了压缩感知技术在处理相对简单几何形状的图像时的优势。

➤ 绘制重建质量指标随欠采样率变化的曲线，解释曲线的变化趋势，讨论压缩感知技术的适用性；
答：如图



- 1) 解释曲线变化趋势
- 信息丢失：随着采样率的降低，获取的原始信息量减少，导致重建算法难以准确恢复原始图像的细节和结构。PSNR和SSIM的下降与信息丢失成正比。
 - 算法鲁棒性：压缩感知技术在一定程度上能够缓解低采样率带来的影响，但其效果依赖于图像的稀疏性和重建算法的有效性。在高稀疏性图像下，压缩感知技术能够有效恢复图像。
 - 阈值效应：在某个特定的采样率阈值（如50%）可能会出现质量的急剧下降，表明在此采样率下，重建算法面临更大的挑战。
- 2) 压缩感知技术的适用性
- 优点：
- 减少数据采集：压缩感知技术允许在低采样率下重建图像，使得医学成像过程更快，同时减少患者的辐射剂量。
 - 高效处理稀疏信号：该技术利用图像的稀疏性，能够在一定程度上恢复图像质量，使其在医学图像重建中非常有效。

局限性:

- 对复杂图像的适应性不足: 在处理结构复杂或细节丰富的图像 (如MRI图像) 时, 压缩感知技术的效果可能不如预期, 导致重建质量下降。
- 依赖先验知识: 压缩感知重建效果往往依赖于对图像稀疏性的先验知识, 缺乏有效的先验知识可能导致重建效果不佳。
- 根据实验结果, 讨论压缩感知在医学图像重建中的优势和局限性, 并提出可能的改进方向。

答:

压缩感知在医学图像重建中的优势和局限性

1. 优势

- **减少采样需求:**
 - 压缩感知技术允许以远低于奈奎斯特采样率的方式获取数据, 从而显著减少数据采集时间和患者的辐射剂量。这在CT和MRI等医学成像中尤为重要。
- **有效利用稀疏性:**
 - 医学图像通常具有高度的稀疏性, 压缩感知技术能够利用这一特性, 通过适当的数学模型 (如L1范数最小化) 有效重建图像。
- **提高重建速度:**
 - 结合快速算法 (如ISTA、FISTA等), 压缩感知可以在较短的时间内实现图像重建, 提高医疗效率。
- **适应性强:**
 - 压缩感知能够适应多种成像模式和不同类型的医学图像, 使其在临床应用中具有广泛的适用性。

2. 局限性

- **对复杂图像的重建效果有限:**
 - 在处理具有复杂结构或细节丰富的图像 (如MRI图像) 时, 压缩感知技术的效果可能不如预期, 重建质量较差, 尤其是在低采样率下。
- **对先验知识的依赖:**
 - 压缩感知的性能高度依赖于对图像稀疏性的先验知识。缺乏有效的先验知识可能导致重建效果的显著下降。
- **计算复杂度:**
 - 尽管压缩感知可以减少采样量, 但其重建算法 (尤其是基于迭代的方法) 通常计算复杂度高, 需要强大的计算资源, 限制了其在实时成像中的应用。
- **伪影问题:**
 - 在低采样率下, 重建图像可能会出现伪影, 影响临床诊断的准确性。

3. 可能的改进方向

- **结合深度学习技术:**
 - 利用深度学习模型 (如卷积神经网络) 来改进压缩感知的重建过程, 可以从大量数据中学习更有效的重建方法, 提高重建质量。
- **自适应采样策略:**
 - 发展自适应采样技术, 根据图像内容动态调整采样率, 以优化重建效果和图像质量。
- **多模态成像融合:**

- 结合不同成像模态的信息（如CT与MRI），提高重建的鲁棒性和准确性，充分利用各模态的优点。
- **改进算法效率：**
 - 研发更高效的重建算法，降低计算复杂度，提高实时成像能力，增强临床应用的可行性。
- **增强先验知识：**
 - 通过引入更多的先验知识（如图像模型、物理特性等），增强重建算法对图像特征的理解，提升重建效果。