计算机视觉与工业应用

2023.8.29

1. 目标和主要内容

图像压缩感知

JPEG图像应用

云端存储压缩和深度学习

高文院士 AVX编码

1. 课程
2. 背景

N维-》M维-》恢复------>fewer measurements than Nyquist sampling（某个稀疏domain）

？稀疏定义是什么

转换成：约束优化问题 零范式-非零系数越少越稀疏 一范式-元素和越小越稀疏

Advantages:fewer measurements/编码简单，解码复杂

Applications:CT/Romote sensing-编码简单、成像快、实时性好

Chanllenges(seek a domain with high sparsity):nonstationary-no universal domain

Fixed domains-poor rate-distortion

Related work

1. motivation:image Priors and sparsity??

"Image prior" 是指在图像处理和计算机视觉领域中，关于图像的一种先验知识或假设。这种先验知识可以是有关图像结构、特征、统计性质等方面的信息。使用图像先验可以帮助改善图像处理、分析和恢复任务的性能，尤其在存在噪声、模糊或不完整信息的情况下。 在很多情况下，图像先验是基于对真实世界图像的观察和分析得出的，或者是基于特定应用领域的背景知识建立的。以下是一些常见的图像先验概念： 1. \*\*平滑性先验（Smoothness Prior）：\*\* 认为图像中相邻像素之间的差异应该是平滑的，即图像的变化应该是连续的。平滑性先验在去噪、图像恢复等任务中有应用。 2. \*\*稀疏性先验（Sparsity Prior）：\*\* 假设图像中只有少数像素是非零或显著的，其余像素是零或接近零。稀疏性先验在图像压缩、特征选择等领域应用广泛。 3. \*\*边缘先验（Edge Prior）：\*\* 假设图像中边缘的分布是有限的，边缘是图像中突变的区域。边缘先验在边缘检测、边缘保留滤波等任务中使用。 4. \*\*纹理先验（Texture Prior）：\*\* 基于对纹理和图像结构的认识，假设图像中存在某种纹理特征。纹理先验在纹理合成、纹理分析等方面有用。 5. \*\*形状先验（Shape Prior）：\*\* 基于对象形状的先验信息，用于物体检测、分割和跟踪等任务。 图像先验可以用于正则化问题、先验驱动的最小二乘法等方法中，以引导图像处理算法在潜在噪声或不完整数据的情况下产生更合理的结果。这些先验在不同的问题和应用中具有重要作用，有助于提高图像处理和计算机视觉算法的性能和鲁棒性。

1. Image Priors:local smoothness and Nonlocal self-similarity（图像先验最直观方法）

Mathematic

①Local 2D Sparsity in space Domain:Gradient 局部平滑性表示：差分统计Laplace model

②Nonlocal 3D Sparsity in Transform Domain:卷积核-块匹配 Transform domain

1/2-DCT(类似频域分解-傅里叶变换实部/可逆)：能量值集中在左上角（越左上频率越小，能量越大）、再把左上角大值进行 DCT变换？

③collaborative sparsity(CoS)->约束优化问题（增广拉格朗日乘数法）/组合优化问题

1. Experimental Results

Objective:index PSNR/FSIM(模拟人眼)-measuremetns、Iteration、Time

Subjective Comparison:

1. Conclusions
2. <https://jianzhang.tech/>

2023.9.5

Group-based:

1. Background

Formulation --- H->不同问题

Classic regularization:edge√ details×

Proposed:dictionary自定义 零范式=元素个数

1. Motivation

Abs+ sparse(0/1)->Problem:large scale and no-connection(batches)\

advantages

1. Proposed
2. group construction:把一个patch一维化、拼接C个patch,提高还原效率->列中蕴含了local smooth
3. Self-Adaptive group dictionary learning

SVD奇异值分解-PCA：上一次D->U \* V->new(uV(T)) 迭代跟新！！！

1. 解优化问题：Bregman
2. Experimental results

Note:零范式效果优于一范式（Bregman）

Idea:自适应字典+零范式、优化求解方法