

区别结构和纹理的稀疏表示图像修复算法

高 飞¹, 王佳君², 喻 强², 张晶晶²

(1. 西藏大学藏文信息技术研究中心, 拉萨 850000; 2. 西南交通大学信息科学与技术学院, 成都 610031)

摘 要: 现有基于稀疏表示的图像修复算法在修复破损区域时对纹理块和结构块不加区分, 导致修复平滑部分时处理时间较长, 同时在修复结构部分时精度较低。针对上述问题, 提出一种改进算法。将待修复块分成纹理块和结构块 2 类, 相应构造不同的学习字典。结构块所应用的字典用于确保修复精度, 而纹理块所应用的字典则用于在保证纹理清晰的基础上加快修复速度。实验结果表明, 该算法在不增加时间复杂度的前提下可有效恢复结构细节, 改善破损图像整体修复效果, 同时降低处理时间。

关键词: 图像修复; 稀疏表示; 结构块; 纹理块; 字典构造; 结构细节

中文引用格式: 高 飞, 王佳君, 喻 强, 等. 区别结构和纹理的稀疏表示图像修复算法[J]. 计算机工程, 2016, 42(3):242-248.

英文引用格式: Gao Fei, Wang Jiajun, Yu Qiang, et al. Image Inpainting Algorithm Based on Sparse Representation Distinguishing Structure and Texture[J]. Computer Engineering, 2016, 42(3):242-248.

Image Inpainting Algorithm Based on Sparse Representation Distinguishing Structure and Texture

GAO Fei¹, WANG Jiajun², YU Qiang², ZHANG Jingjing²

(1. Research Center of Tibetan Information Technology, Tibet University, Lhasa 850000, China;

2. School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610031, China)

【Abstract】 Existing image inpainting algorithms based on sparse representation do not distinguish structure patch and texture patch when constructing the learning dictionary. This may increase the processing time when inpainting the smooth part of the damaged region, and decrease the inpainting accuracy when inpainting the structure part. To solve this problem, this paper proposes an improved algorithm. It classifies the patches around the damaged region into two categories where one category includes structure patches while the other includes texture patches, and correspondingly constructs two kind of learning dictionaries. For structure patches, their corresponding dictionaries are used to ensure high accuracy. For texture patches, their corresponding dictionaries are used to accelerate inpainting under the condition of clear inpainted texture. Experimental result shows that the proposed algorithm can effectively inpaint the structure detail without increasing the complexity, while the inpainting effect is better with low time consumption.

【Key words】 image inpainting; sparse representation; structure patch; texture patch; dictionary construction; structure detail
DOI: 10.3969/j.issn.1000-3428.2016.03.044

1 概述

近年来, 稀疏表示理论已成功应用到图像修复领域, 学者们针对基于稀疏表示的图像修复问题进行了研究^[1-2]。该理论的关键在于如何进行有效的稀疏编码以及构建最适合待修复图像的字典。

目前, 最常见的构建字典有显示字典和隐式(学

习)字典, 显示字典有过完备离散余弦变换(Discrete Cosine Transformation, DCT)、曲波^[3-5]、轮廓波^[6-8]、Bandelets^[9-10]和小波等, 隐式(学习)字典有最优方向(Method of Optimal Direction, MOD)字典、K次奇异值分解(K-singular Value Decomposition, K-SVD)字典^[11]等。与显示字典相比, 隐式(学习)字典的针对性和执行力更强, 能够达到更好的修复目的。在

基金项目: 国家自然科学基金资助项目(61461048); 国家社会科学基金资助项目(12EF19); 国家级大学生创新创业训练计划基金资助项目(201210694019); 西藏自治区重点科技计划基金资助项目(Z2013B28G28/02)。

作者简介: 高 飞(1980-), 男, 副教授、硕士, 主研方向为数字图像处理; 王佳君(通讯作者)、喻 强、张晶晶, 硕士研究生。

收稿日期: 2015-01-21 **修回日期:** 2015-03-24 **E-mail:** 337679107@qq.com

构造隐式(学习)字典时,其学习算法基本都遵循K-均值算法的框架,即每次迭代都包含2个步骤:(1)稀疏编码,即求解训练样本在字典上的稀疏系数;(2)字典更新,即更新字典原子以更好地适应样本。目前,K-SVD是最常用的字典学习算法之一。

文献[11]提出了K-SVD字典训练算法,并将其应用于图像去噪等真实图像数据的处理中。之后,文献[12]又采用K-SVD字典训练算法来解决图像修复问题,得到了较好的修复效果。为了进一步提高K-SVD算法的收敛速度,降低稀疏表示误差,文献[13]引入块稀疏表示进行字典优化。该优化算法是K-SVD算法的扩展,它采用对字典块结构和字典原子的交替迭代更新以更好地适应数据,进而提高算法的收敛速度,减少表示误差。

无论是文献[12]用K-SVD算法训练得到的字典,还是文献[13]提出的块稀疏字典,在利用其进行图像修复时,都对输入的图像块不加区分,一视同仁地进行修复,这就导致在修复破损区域的纹理部分时浪费了大量精力,增加了算法的处理时间,毕竟一般图像的破损区域中多数都是纹理部分,而过于精细地修复纹理部分并非十分必要。反之,在修复破损区域的结构部分时,现有算法又显得不够精细,如K-SVD算法在迭代到一定次数时,修复效果难以再有提高。即使是块优化字典,在修复一些线形区域时也不能较好地保持结构连贯性,对结构的细节修复效果有待提升。

针对文献[12-13]方法存在的缺点,本文提出一种改进算法。该算法将输入的图像块划分成2类:纹理块和结构块,分别利用块稀疏表示优化算法和原始K-SVD算法构造,同时使用不同形式的目标函数。2类字典的区别体现在原子类型选择、原子个数设置和字典训练的迭代次数上。字典原子选择和原子个数设置都与输入的待修复块类型相适应,一类待修复块对应一类字典。使用不同字典进行修复的同时也选择使用相适应的目标函数形式。目标函数形式的选择则借鉴了文献[14]在图像去噪中的实验结果。本文算法应用稀疏表示理论对输入的破损图像块采取区别对待、分别修复的策略,一方面从整体上降低修复算法对破损图像的处理时间,另一方面也可改善破损区域结构信息的细节修复效果。

2 基于稀疏表示的图像修复

本文算法应用稀疏表示理论完成对破损图像的修复。首先,寻求一个合理的矩阵字典 $D \in \mathbf{R}^{N \times K}$;其次,利用字典 D 对待修复块向量 $y_i \in \mathbf{R}^N$ 进行合适的稀疏表达(找到 y_i 对应的稀疏系数向量 $x_i \in \mathbf{R}^K$,此过程也称为稀疏编码);最后,完成对待修复块的修复。以一个待修复块 y_i 为例,具体步骤如下:

(1)构造一个合理的字典 D (可以是显示字典,也可以是隐式(学习)字典)。

(2)利用字典 D 和 y_i 中的已知像素点求得稀疏系数 x_i 。

(3)利用解得的稀疏系数 x_i 和字典 D 求得 y_i 中的未知像素点。

如图1所示,对于破损图像 I 中每一个待修复块 y_i ,都包含已知部分(已知像素点值的完好区域)向量 y_i^s 和未知部分(未知像素点值的破损区域)向量 y_i^u 。首先,需要寻求一个稀疏系数 x_i ,该系数 x_i 能够逼近待修复块 y_i 中的已知部分 y_i^s ,如式(1)所示。

$$y_i^s \approx D^s x_i \quad (1)$$

其中,矩阵 D^s 是通过字典 D 中对应 y_i 未知像素点的位置进行掩码获得的。利用式(1)求得稀疏系数 x_i ,然后利用式(2)近似恢复出待修复块 y_i 中未知部分 \hat{y}_i^u 。

$$\hat{y}_i^u = D^u x_i \quad (2)$$

其中,矩阵 D^u 是通过字典 D 中对应 y_i 已知像素点的位置进行掩码获得的。

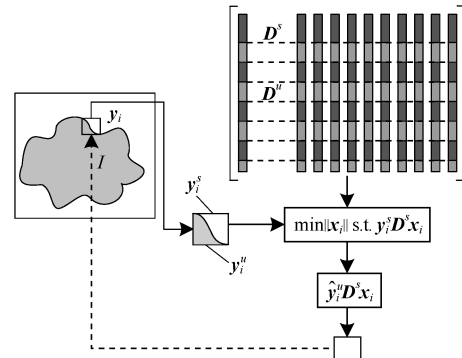


图1 基于稀疏表示的图像修复示例

3 图像去噪中的最佳解与最稀疏解

K-SVD算法可以同时构造、更新字典并进行稀疏编码。要求达到的目标函数如下:

$$\min_{D, X} \{ \|Y^s - D^s X\|_2^2 \} \text{ s. t. } \forall i, \|x_i\|_0 \leq T_0 \quad (3)$$

或:

$$\min_{D, X} \sum_i \|x_i\|_0 \text{ s. t. } \{ \|Y^s - D^s X\|_2^2 \} \leq \varepsilon \quad (4)$$

其中, Y^s 是边缘上所有待修复块已知部分像素点构成的矩阵; T_0 代表稀疏度的固定值; ε 代表误差的固定值。由式(3)或式(4)得到字典 D 和系数矩阵 X ,再利用式(5)得到边缘上所有待修复块未知像素点构成的矩阵 Y^u 的估计值,完成破损边缘的第一次修复。

$$\hat{Y}^u = D^u X \quad (5)$$

然后更新破损边缘,重复利用式(3)或式(4)、式(5),重新更新字典和重新进行稀疏编码,逐渐完

成对整个破损区域的修复。

与图像修复类似,在图像去噪问题中也常利用 K-SVD 算法进行更新字典和稀疏编码。而 K-SVD 算法在进行稀疏编码时,通常使用正交匹配追踪 (Orthogonal Matching Pursuit, OMP) 算法计算出最稀疏的系数,即在正交匹配算法中,每次只寻找一个基底。近似逼近 y_i 时所用的基底越少,稀疏编码得到的系数越稀疏。利用式(6)得到的解即为最稀疏的系数向量 $x_{i\min}$ 。

$$x_{i\min} = \min_{x_i} \|x_i\|_0 \text{ s. t. } \|Dx_i - y_i\|_2 \leq \varepsilon \quad (6)$$

如果放宽限制,不局限于寻找最稀疏的系数,而是寻找所有满足稀疏度小于 K 的系数,利用 Random-OMP^[14] 算法可以得到一组个数为 J 的候选系数向量 x_{ij} ,如式(7)所示。

$$\left\{ \begin{array}{l} x_{ij} \\ \text{s. t. } \forall j \left\{ \begin{array}{l} \|x_{ij}\|_0 \leq K \\ \|Dx_{ij} - y_i\|_2 \leq \varepsilon \end{array} \right. \end{array} \right. \quad (7)$$

每一个候选系数 x_{ij} 与字典 D 相乘都可以得到一个 y_i 的近似逼近。本文参考文献[14]实验中的部分结果,即最稀疏系数 $x_{i\min}$ 和一组 Random-OMP 候选系数:

$$g = \|x_{ij}^{\text{Rand-OMP}}\|_0 \quad (8)$$

对比两者所具备的基底数 g 和产生的近似误差,结果如图 2 和图 3 所示。在图 2 中,横轴坐标表示利用 Rand-OMP 算法解得的一系列候选系数所具备的基底数,纵轴坐标表示具备相同基底数的候选系数的个数,虚线表示利用 OMP 算法求出的最稀疏系数(唯一)所具备的基底数。

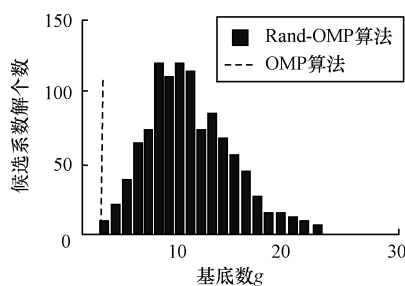


图 2 候选系数解个数与基底数的关系

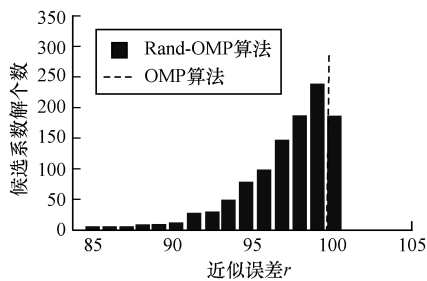


图 3 候选系数解个数的近似误差

通过比较图 2 和图 3 可知,OMP 算法可以给出

最稀疏的解,但最稀疏的解对于图像去噪来说并不是最有效的。

本文推知最稀疏的系数 $x_{i\min}$ 所含的基底数最少,但并没有取得最小的近似误差,即 $x_{i\min}$ 与 D 相乘并不一定是 y_i 的最佳近似,误差计算公式为:

$$r = \|Dx_{i,j} - y_i\|_2^2 \quad (9)$$

因此,在图像去噪中,找到最稀疏的系数可以使计算变得更简单,但并不表示利用最稀疏系数 $x_{i\min}$ 恢复得到的去噪图像就是最佳的。

受此启发,笔者认为在图像修复中,式(3)和式(4)不仅是一个问题的 2 个角度,而是两者各有侧重,即:式(3)旨在找出在系数 X 满足一定稀疏度条件下, D^* 与 X 相乘可以最佳逼近 Y^* 的字典 D ;式(4)侧重于在满足一定近似误差的条件下,找出使计算简单的最稀疏的系数 X 。本文从最稀疏解和最佳解的角度出发,选择使用不同形式的目标函数,结合 K-SVD 算法和块稀疏表示优化算法构造 2 类不同字典,完成对破损图像的修复。

4 本文算法

本文提出区别结构和纹理的稀疏表示图像修复算法,具体步骤如下:

(1) 利用块结构稀疏度 $\rho(p)$ ^[15] 的概念将图 1 中破损边缘上所有的待修复块划分成 2 类:一类是包含大量结构信息的待修复块,称为结构块,这类待修复块组成集合 A ;另一类是含有大量纹理信息的待修复块,称为纹理块,这类待修复块组成集合 B 。如图 4 所示,将属于集合 A 的待修复块 y_i 标记为 a ,此类待修复块的结构稀疏度数值都大于规定的阈值;其余未标记的待修复块则属于集合 B ,此类待修复块的结构稀疏度数值则小于规定的阈值。在一般情况下,在一幅图像中,集合 A 中所含的待修复块(结构块)个数要远少于集合 B 中所含的待修复块(纹理块)个数,因此,提高对集合 A 处理精度的同时,在一定前提下,与现有算法相比,简化对集合 B 的处理反而会减少整个算法的处理时间。

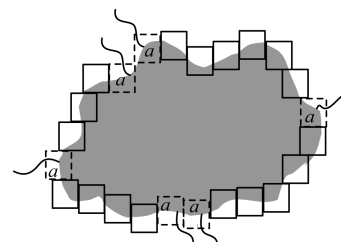


图 4 图像块划分示意图

(2) 构造 2 类不同学习字典。具体原因如下:

1) 利用破损图像的有效信息,对所有图像块统一进行字典训练,可以达到一定的修复效果。但是

有时不能较好地保持结构连贯性,在结构线性区域的细节修复上存在缺陷。因为 K-SVD 字典是一种满足一定规则的随机学习字典,而对图像块进行修复时,虽然选择了约束条件下的最优线性组合,但仍不失随机性,其本质是一种随机概率问题,只有更逼近没有完全真实。如果某一图像块表征纹理的区域面积大,且纹理块之间相似性高,最终选取的线性组合近似匹配原图像块的概率就高。而包含线性区域等一些结构细节的图像块,相似的区域面积小、独特性强,如果不针对其特意进行修复,很容易被现有算法忽略。

2) 本文考虑在修复集合 B 中的待修复块时,主要针对破损区域纹理进行修复,对速度的要求优先于对精度的要求。因此,在选取字典原子个数时, K_B 的数目应适当小于 K_A ,以提高算法的计算效率。理由是过完备字典具备冗余性,即过完备字典中列的数目远多于表达一图像块所必须的列数。这种字典的优势是更有利于表达图像块数据。因为列的备选组合很多,在满足约束条件时,选用最合理的组合可以尽可能逼近图像块。但是在利用过完备字典对图像平滑部分(纹理块)进行稀疏表示时,如果降低完备字典的冗余度,在较少的备选组合中也可以找到满足修复要求的合理组合。因此,对图像的纹理块和图像块统一进行字典训练完成修复,降低了计算效率。

3) 利用原 K-SVD 算法构造字典时,在迭代过程中,当迭代满足一定次数时,图像修复质量难以得到进一步改善,如图 5 所示。该图显示了选择 2 幅破损图像(本文实例验证中划痕修复的 2 组实验的破损图像),利用原 K-SVD 算法构造字典进行修复时,图像质量与迭代次数的关系。因此,本文算法在两类字典训练时,在迭代次数的设置上也有所改进。

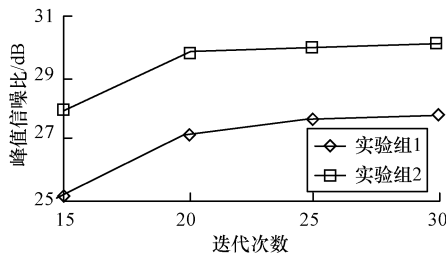


图 5 图像质量与迭代次数的关系

(3) 选择与所使用字典相适应的不同形式的目标函数,即式(3)或式(4),具体原因在第 3 节中已阐述。

本文算法通过对集合 A 和集合 B 中的待修复块区别对待,分别修复,以期能更好地修复图像的结构信息,并更快地完成修复过程。

5 字典构造与目标函数选择

构造用于修复集合 A 中待修复块的字典矩阵 D_A

$\in \mathbf{R}^{N \times K_A}$ 。 D_A 的初始字典矩阵设为 D_A^0 , D_A^0 中选择的 K_A 个初始原子应与 A 中待修复块相似,具有大量的结构信息。每个初始原子都来自于图像未破损部分的结构区域。

修复集合 B 中待修复块时,构造字典 D_B 。设 D_B 的初始字典为 D_B^0 , D_B^0 中选择的 K_B 个初始原子应与 B 中待修复块相似,来自于图像未破损部分的平滑纹理区域,每个初始原子都具有大量的纹理信息。

为明确 K_A 和 K_B 的取值,当某一待修复块的结构稀疏度 $\rho(p)$ 大于阈值 ξ_0 时,可认定该图像块为结构块。由实验可得,对于结构块,如果图像块的大小为 $n \times n$,则字典原子个数设置大致为 $4n^2$; $\rho(p)$ 为 ξ 小于阈值 ξ_0 时,则认定为纹理块。对于纹理块,设置字典原子个数为 $\xi_0/\bar{\xi}n^2$ 。 $\bar{\xi}$ 为所有纹理块结构稀疏度的平局值。构造的原子个数函数,如式(10)所示。

$$K_A = 4n^2; K_B = \frac{\xi_0}{\bar{\xi}}n^2 \quad (10)$$

接下来,对字典 D_A 首先应用 K-SVD 进行字典更新和稀疏编码,式(3)可以改写成式(11),将式(11)作为目标函数,以期精确修复结构区域。

$$\min_{D_A, X_A} \{ \|Y_A^i - D_A X_A\|_2^2 \} \text{ s. t. } \forall i, \|x_i\|_0 \leq T_0 \quad (11)$$

在修复集合 A 中待修复块时,再采用 K-SVD 扩展算法(块稀疏表示优化算法)改进已经训练出的 D_A ,得到具有块结构向量 $q \in \mathbf{R}^k$ 的块稀疏字典 $D_N^{[13]}$ 。 D_N 中的原子聚类成少量的块,原子以块排列,如图 6 所示。

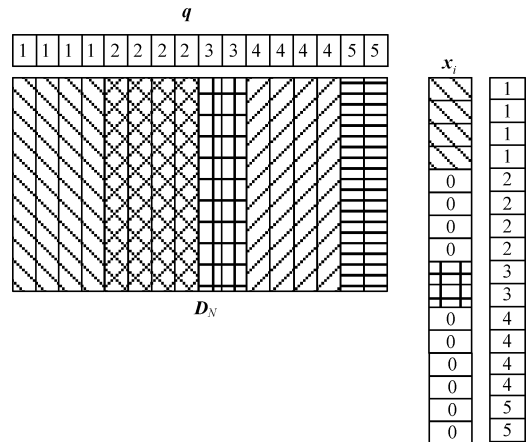


图 6 块稀疏表示

在图 6 中,每一个块都被给定一个索引数字, D_N 中的索引数字为 1 ~ 5,即 D_N 中含有 5 个块,块结构 $q \in \mathbf{R}^5$,用来将 D_N 中的各个原子划分给每个块。如果用来近似逼近待修复块 y_i 的多个原子集中在 D_N 的 k 个块上(图 6 中 $k=2$),那么待修复块 y_i 所对应的系数 x_i 中的非零项也仅集中在相对应的 k 个部位上(见图 6 中 x_i 的 2 处阴影部位),此时称系数 x_i 为 k -块稀疏表达,定义式如下:

$$\|x_i\|_{0,q} = k \quad (12)$$

引入块稀疏字典后,将式(11)重新整理如下:

$$\begin{aligned} & \min_{D_{NA}, q, X_A} \|Y_A^s - D_{NA}^s X_A\|_2^2 \\ & \text{s. t. } \|x_i\|_{0,q} \leq k_A, i = 1, 2, \dots, L \\ & \|q_j\| \leq s, j \in [1, J] \end{aligned} \quad (13)$$

其中, D_{NA} 是用以修复集合 A 中待修复块的块稀疏字典; q 是字典的块结构向量; X_A 是所有稀疏系数 x_i 组成的矩阵; Y_A^s 是由每一个待修复块 y_i 已知像素点构成的矩阵; $\|x_i\|_{0,q}$ 表示稀疏系数 x_i 中的非零项集中的部位数(见图 6 中 x_i 所表示的系数向量 2 个阴影部分); k_A 表示个数; $|q_j|$ 表示字典中第 j 个块包含的原子数,即块的尺寸; s 表示允许块的最大尺寸; J 是字典中块结构的数目。

以上是本文算法对结构块进行修复时采取的策略。而在修复纹理块时,只利用 K-SVD 算法构造字典矩阵 D_B, D_B 在训练中的迭代达到一定次数后,图像的修复质量难以在后续的迭代中得到明显改进。为了提高计算效率,本文算法就此停止迭代,设此时 D_B 训练的迭代次数为 m (根据图 5 所得实验数据,本文算法中设 $m = 20$)。而在对结构块进行修复时,采用的 K-SVD 扩展算法已克服了这项缺点,因此,为了更精确地修复结构细节,本文算法把训练 D_{NA} 时的迭代次数预设 3 m 。

同样,出于速度优先的考虑,修复集合 B 中待修复块(纹理块)时,把式(4)改写成式(14),将其作为目标函数。

$$\min_{D_B, X_B} \sum_i \|x_i\|_0 \quad \text{s. t. } \|Y_B^s - D_B^s X_B\|_2^2 \leq \varepsilon \quad (14)$$

最终算法流程如图 7 所示。

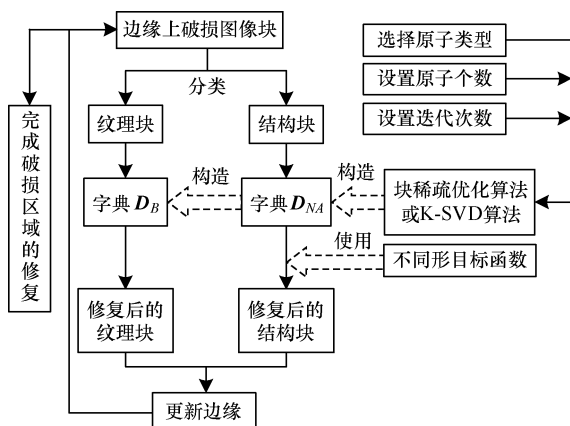


图 7 本文算法流程

算法步骤如下:

- (1) 已知待修复块集合中的 Y_A^s (或 Y_B^s), 设定个数 k_A 和块的最大尺寸 s 。
- (2) 设置初始字典 D_{NA}^0 (或 D_B^0)。
- (3) 分别利用不同目标函数形式(式(13)或式(14))。式(13)通过稀疏聚类(Sparse Agglomerative

Clustering, SAC) 和块奇异值分解(Block K-Singular Value Decomposition, BK-SVD) 算法^[13] 找到字典 D_{NA} , 块结构向量 q 和相应的稀疏系数 X_A 。式(14)直接通过 K-SVD 训练得到字典 D_B 和相应的稀疏系数 X_B 。

(4) 利用字典 D_{NA} (或 D_B) 和稀疏系数 X_A (或 X_B), 求得 Y_A^u (或 Y_B^u), 完成一次边缘修复。

(5) 更新边缘, 重新组合待修复块集合 Y_A^s (或 Y_B^s), 此时, 个数 k_A 、块的最大尺寸 s 以及初始字典 D_{NA}^0 (或 D_B^0) 不再需要重新变化。

(6) 再次利用式(13)和式(14), 通过 SAC 算法、BK-SVD 算法^[13] 或 K-SVD 算法更新字典 D_{NA} (或 D_B), 块结构向量 q 和相应的稀疏系数 X_A (或 X_B)。

(7) 利用更新后的字典 D_{NA} (或 D_B) 和稀疏系数 X_A (或 X_B) 求得新的 Y_A^u (或 Y_B^u), 再完成一次边缘修复。

(8) 反复迭代步骤(5)~步骤(7), 直至修复完成整个破损区域。

6 实例验证分析

为验证本文算法的性能, 本文采用 Matlab2009b 作为工具, 在 Intel 2.5 GHz CPU 的计算机上设计 3 组实验, 并与文献[12-13]算法的修复结果进行对比。

6.1 划痕破损修复效果比较

2 组划痕破损修复实验的效果比较分别如图 8 和图 9 所示。

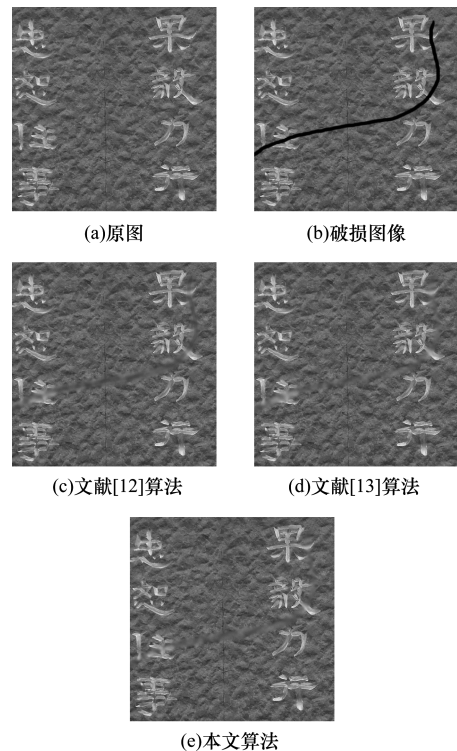


图 8 划痕破损修复效果比较 1

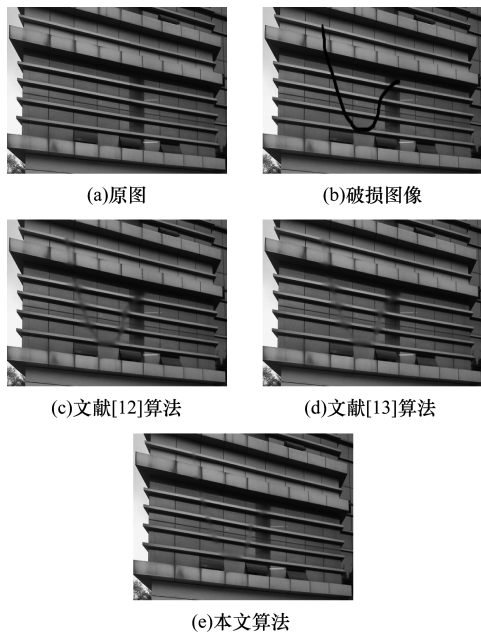


图 9 划痕破损修复效果比较 2

将 3 种算法修复结果中的某一破损部位放大后,可以清晰地观察出不同算法的修复效果,分别如图 10 和图 11 所示。

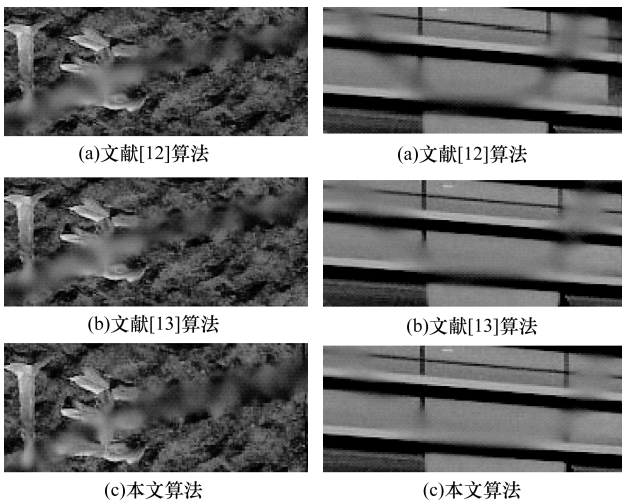


图 10 划痕破损修复效果放大比较 1

图 11 划痕破损修复效果放大比较 2

可以看出,文献[12-13]算法和本文算法在对图像线性结构细节进行修复时效果存在差异,文献[13]算法和本文算法的修复效果优于文献[12]算法。如在实验组 1 中,不同算法针对破损“任”字的修复差异,以及实验组 2 中灰白色直线窗沿的修复差异。文献[13]算法和本文算法在视觉效果上的修复结果差别不大,在客观评价指标上本文算法略高于文献[13]算法。但是在算法的处理速度上,本文算法明显优于文献[13]算法,具体实验数据如表 1 所示。

表 1 客观评价指标比较

实验组号	文献[12]算法		文献[13]算法		本文算法	
	PSNR/dB	处理时间/s	PSNR/dB	处理时间/s	PSNR/dB	处理时间/s
1	27.7	139.5	28.3	142.3	28.9	110.6
2	30.1	162.6	32.4	184.1	32.6	120.7

6.2 块破损修复效果比较

图 12 给出了块破损图像修复比较结果。可以看出,在结构连贯性方面,本文算法修复结果最佳,这是因为本文的修复算法与文献[12-13]算法相比,更合理和充分地利用了破损图像中的结构信息,在修复破损结构区域时,细节上表现得更细腻。例如,原图破损区域右下部分处的凹陷,本文算法修复结果显示被成功修复。在纹理清晰性方面,本文算法对纹理的修复与另外 2 种算法在视觉效果上无明显优劣,理论上文献[12-13]算法还应较本文算法更理想,这是因为本文算法侧重于对结构的细致修复,纹理方面在保证一定的修复效果时只求迅速。

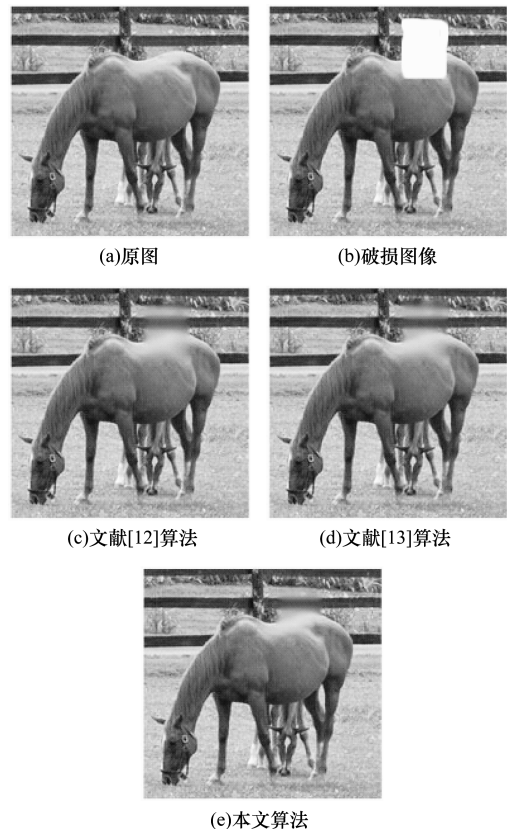


图 12 块破损图像修复效果比较

7 结束语

本文提出一种区别结构与纹理的稀疏表示图像修复算法。对图像破损区域组成的待修复块,在利用字典进行修复之前,先依据规则进行分类。对不同类别的图像块,从多方面构造并使用不同的字典,

设置不同的迭代次数,应用不同形式的目标函数进行修复。与文献[12-13]算法相比,本文算法在保证一定纹理清晰的前提下,对图像结构细节和线形区域的修复效果有所改善,修复效率较高。今后将针对字典原子的选择算法、原子个数的设置原则和目标函数的改进方法进行深入研究,在破损区域纹理修复和结构修复两方面寻求最佳平衡点,以期进一步改善修复效果。

参考文献

- [1] 黄江林. 基于稀疏表示的图像修复算法研究[D]. 合肥:安徽大学,2013.
- [2] 张 健. 基于稀疏表示模型的图像复原技术研究[D]. 哈尔滨:哈尔滨工业大学,2014.
- [3] Preeti S, Jain A. Image Denoising by Supervised Adaptive Fusion of Decomposed Images Restored Using Wave Atom, Curvelet and Wavelet Transform[J]. Signal, Image and Video Processing, 2014, 8(3):443-459.
- [4] Do M, Vetterli M. The Contourlet Transform: An Efficient Directional Multiresolution Image Representation[J]. IEEE Transactions on Image Processing, 2005, 14(12):2091-2106.
- [5] Candes E, Demanet L, Donoho D, et al. Fast Discrete Curvelet Transforms[J]. SIAM Multiscale Modeling and Simulation, 2006, 5(3):861-899.
- [6] Do M, Vetterli M. Contourlets: A New Directional Multiresolution Image Representation[C]//Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers. Washington D. C., USA: IEEE Press, 2002:497-501.
- [7] Do M, Vetterli M. The Contourlet Transform: An Efficient Directional Multiresolution Image Representation[J]. IEEE Transactions on Image Processing, 2005, 14(12):2091-2106.
- [8] Do M, Lu Yue. A New Contourlet Transform with Sharp Frequency Localization[C]//Proceedings of IEEE International Conference on Image Processing. Washington D. C., USA: IEEE Press, 2006:1629-1632.
- [9] le Pennec E, Mallat S. Sparse Geometric Image Representations with Bandelets[J]. IEEE Transactions on Image Processing, 2005, 14(4):423-438.
- [10] Gabriel P, Stephane M. Surface Compression with Geometric Bandelets[J]. ACM Transactions on Graphics (TOG), 2005, 24(3):601-608.
- [11] Elad M. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing[M]. Berlin, Germany: Springer-Verlag, 2010.
- [12] Aharon M, Elad M, Bruckstein A. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation[J]. IEEE Transactions on Signal Processing, 2006, 54(11):4311-4322.
- [13] Zelnik-Manor L, Rosenblum K, Eldar Y. Dictionary Optimization for Block-Sparse Representations[J]. IEEE Transactions on Signal Processing, 2012, 60(5):2386-2395.
- [14] Elad M. Sparse Representations Are Most Likely to Be the Sparsest Possible[M]. Haifa, Israel: Hindawi Publishing Corporation, 2005.
- [15] Xu Zongben, Sun Jian. Image Inpainting by Patch Propagation Using Patch Sparsity[J]. IEEE Transactions on Image Processing, 2010, 19(5):1153-1165.
- [16] Huang Jiabin, Kang S B, Ahuja N, et al. Image Completion Using Planar Structure Guidance[J]. ACM Transactions on Graphics (TOG), 2014, 33(4):1-10.
- [17] Pérez P, Gangnet M, Blake A. Poisson Image Editing[J]. ACM Transactions on Graphics (TOG), 2003, 22(3):313-318.
- [18] Chum O, Matas J. Planar Affine Rectification from Change of Scale[M]//Kimmel R, Klette R, Sugimoto A. Computer Vision-ACCV 2010. Berlin, Germany: Springer-Verlag, 2011:347-360.
- [19] Zhang Zhengdong, Liang Xiao, Ganesh A, et al. TILT: Transform Invariant Low-rank Textures[M]//Kimmel R, Klette R, Sugimoto A. Computer Vision-ACCV 2010. Berlin, Germany: Springer-Verlag, 2011:314-328.
- [20] Lowe D G. Distinctive Image Features from Scale-invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.
- [21] Comaniciu D, Meer P. Mean Shift: A Robust Approach Toward Feature Space Analysis[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2002, 24(5):603-619.
- [22] Arias P. A Variational Framework for Exemplar-based Image Inpainting[J]. International Journal of Computer Vision, 2011, 93(3):319-347.
- [23] Darabi S. Image Melding: Combining Inconsistent Images Using Patch-based Synthesis[C]//Proceedings of SIGGRAPH 2012. New York, USA: ACM Press, 2012.
- [24] 吴 昊, 徐 丹. 基于块匹配中主几何变换的图像合成[J]. 系统仿真学报, 2013, 25(9):2239-2246.
- [25] He Kaiming, Sun Jian. Statistics of Patch Offsets for Image Completion[M]//Fitzgibbon A, Lazebnik S, Perona P, et al. Computer Vision-ECCV 2012. Berlin, Germany: Springer-Verlag, 2012:16-29.

编辑 金胡考

编辑 金胡考

(上接第241页)