

DOI:10.11918/201909195

非迭代收缩多方向预测的空域错误隐藏算法

王凯巡¹, 刘浩^{1,2}

(1. 东华大学 信息科学与技术学院, 上海 201620; 2. 上海交通大学 人工智能教育部重点实验室, 上海 200240)

摘要: 现有的空域错误隐藏算法通常以高复杂度的迭代逼近机制换取恢复质量的轻微提升, 且一些算法只适合于特定的丢失模式. 为此, 本文提出了一种非迭代收缩多方向预测 (Non-iterative Shrinkage Multi-directional, NSM) 的空域错误隐藏算法, 以便更好地平衡计算复杂度与恢复质量等性能指标, 且能处理各种各样的丢失模式. 对于当前受损块的错误隐藏, NSM 算法首先通过各向同性梯度检测器学习当前延拓区域的梯度特征; 随后, 基于具有 16 邻域像素和 8 预测方向的基本隐藏单元, 多方向预测器按照收缩填充次序逐一地恢复受损块的每一个像素, 根据该像素的邻域像素可用情况调整预测器的加权系数. 在每个受损块的内部, 不同的像素组根据它们的邻域级可用度一组接一组地进行恢复; 在一个像素组的内部, 不同丢失像素根据先验填充准则一个接一个地进行预测, 从而实现低复杂度的非迭代重构. 相比于其他空域错误隐藏算法, 实验结果表明, 所提 NSM 算法在各种丢失模式下均能够取得良好的综合性能, 在通用性、计算复杂度和恢复质量之间达到了一种具有竞争力的性能折衷.

关键词: 空域错误隐藏; 多方向预测器; 非迭代重构; 性能折衷

中图分类号: TN919.8

文献标志码: A

文章编号: 0367-6234(2020)10-0128-07

Non-iterative shrinkage multi-directional prediction for spatial error concealment

WANG Kaixun¹, LIU Hao^{1,2}

(1. College of Information Science and Technology, Donghua University, Shanghai 201620, China;

2. Key Laboratory of Artificial Intelligence (Ministry of Education), Shanghai Jiao Tong University, Shanghai 200240, China)

Abstract: At the cost of high complexity, the existing spatial error concealment algorithms may slightly improve the recovery quality by iterative approximation, and some algorithms are optimized only for certain loss patterns. To achieve a trade-off among performance indices, this paper proposes a non-iterative shrinkage multi-directional (NSM) prediction algorithm aiming at handling various types of loss patterns. For the error concealment of the current missing block, the proposed NSM algorithm firstly adopts an isotropic gradient detector to learn the local feature information of the current extrapolation region. Based on a basic concealment unit with 16-pixel neighbor and eight prediction directions, the multi-directional predictor recovers each pixel of the missing block one by one in a shrinkage filling order, and adjusts the weighting coefficients of the predictor according to the availability of adjacent pixels. During the shrinkage filling of a block, different pixel groups are recovered group by group according to their neighbor-level availabilities, and different missing pixels in a pixel group are predicted one by one according to a priori filling rule. Thus, non-iterative reconstruction with low complexity can be realized. Compared with other spatial error concealment algorithms, experimental results show that the proposed NSM algorithm achieved better overall reconstruction performance under various loss patterns, and realized a competitive performance trade-off among versatility, computational complexity, and recovery quality.

Keywords: spatial error concealment; multi-directional predictor; non-iterative reconstruction; performance trade-off

图像通信按光栅扫描顺序执行分块编解码, 在传输过程中存在块丢失现象. 信宿端的错误隐藏不需要对信源端进行任何改变, 在恢复丢失像素的过程中完全不需要信源端的参与, 属于图像通信的一类后处理技术, 由于不需要增加编码冗余或改变码

流结构而受到广泛关注^[1-2]. 大多数自然图像具有较强的空域相关性, 空域错误隐藏算法需要通过可用信息来高效地预测丢失像素, 提高可视质量^[3].

快速的空域错误隐藏算法均采用了低复杂度的线性插值机制, 基本原理是利用已正确接收的邻域像素来进行加权预测. 作为图像通信广泛使用的空域错误隐藏算法, 高级视频编码 (Advanced Video Coding, AVC) 插值算法^[4]利用受损块的邻近像素进行双线性插值, 通过计算上、下、左、右 4 个邻近像素的加权平均值来估计丢失像素, 其权重与像素间的

收稿日期: 2019-09-25

基金项目: 上海市自然科学基金 (18ZR1400300);

人工智能教育部重点实验室开放基金 (LAI202112)

作者简介: 王凯巡 (1997—), 男, 硕士研究生

通信作者: 刘浩 (1977—), liuhao@dhru.edu.cn

距离成反比. AVC 算法难以恢复边缘较多或纹理较丰富的图像区域. 内容自适应划分(Content Adaptive Division, CAD)插值算法^[5]利用边缘检测算子来评估受损块的边缘强度,根据强度大小将受损块分成三种类型:平滑块、纹理块和边缘块,分别选取相应的插值方式进行恢复,其计算复杂度较低. 作为对象移除应用的典型技术,耗时的图像补全机制主要关注视觉主观效果,但在松散的时间约束下也有学者将其用于解决错误隐藏问题,Chung 等^[6]提出了一种联合图像补全和线性插值(Hybrid Inpainting Interpolation, HII)算法,并为两种机制的切换提供了判决门限. 由于边缘结构比单一纹理在视觉上更为重要,边缘导向插值(Edge Guided Interpolation, EGI)算法^[7]首先通过自适应阈值 Sobel 算子估计显著的边缘信息,然后选择边缘效应最强的候选边缘进行多方向插值. 对于存在强边缘的受损块,EGI 算法有助于增强其边缘信息,然而要找到所有边缘的准确位置是非常困难的. 多方向插值算法具有运行速度快与通用性好的优点,但当块间内容变化较大时,候选像素与权重具有不稳定性,影响恢复质量.

大多数自然图像具有低通特性,图像块可以被建模为基函数的线性组合. 基于块的双边滤波(Block-based Bilateral Filtering, BBF)算法^[8]利用一对高斯函数来估计受损块的边缘信息:一个基函数用于测量受损块和可用邻域的相似度,另一个基函数用于测量两者的空域差异性,BBF 算法的复杂度较低. 基于最小均方误差估计器, Koloda 等^[9]提出了一种多模式基函数(Multi-Mode Kernel, MMK)算法,通过计算受损块的概率密度函数,利用邻域信息来确定支撑模型阶数,通过耗时的迭代逼近机制来预测丢失像素.

近年来,信号稀疏表达广泛用于图像处理. 稀疏线性预测(Sparse Linear Prediction, SLP)算法^[10]根据当前受损块的邻域情况调整扫描顺序和候选像素集合的形状,并用贝叶斯信息准则来确定预测器的阶数,预测器阶数的增加会导致复杂度呈指数级增长. 随着机器学习的兴起,双稀疏表达(Dual Sparsity Regularization, DSR)算法^[11]通过主成分分析进行字典构建,解决了因图像数据不足造成字典训练无法完成的难题. DSR 算法利用欧式距离小于某一阈值来确定用于字典构建的候选像素集合和模板像素集合,但固定阈值导致的候选块选取不准问题会降低预测模型的通用性. 进一步地,联合稀疏学习(Joint Sparse Learning, JSL)算法^[12]改进了稀疏表达的字典构建与系数求解,利用自适应阈值来提高搜索候选像素集合和模板像素集合的准确性. 目前,基于稀

疏表达的错误隐藏算法均采用线下训练的字典构建,在实际应用中需要大量图像作为训练数据,难以适用于各种丢失模式. 即使不考虑模型训练的耗时与泛化性问题,当前稀疏模型类错误隐藏算法的计算复杂度仍然过高.

在空域错误隐藏应用中,每个受损块相对较小但序号明确,恢复目标是在较低复杂度的约束下使重构误差达到可接受的程度. 迭代类隐藏算法以高复杂度换取了恢复质量的轻微提升;一些隐藏算法专门针对特定的丢失模式进行模型训练或算法优化,难以适用于各种各样的丢失模式. 因此,有必要提出一种新的空域错误隐藏算法,以更好地平衡通用性、计算复杂度和恢复质量等性能指标.

1 问题描述

1.1 块的丢失模式

在图像通信中,为了便于码流传输,多个块封装成一个包,一幅图像通常编码成若干独立的包,每个包可以单独恢复,当一幅图像出现了包丢失,信宿端根据受损块的位置,利用邻域的无错块或已隐藏块进行恢复. 空域错误隐藏不同于图像去噪或对象移除等应用,图像通信的信宿端能够获取受损块在图像中的具体位置. 在光栅扫描顺序下,一个丢失像素的重构不仅依赖于正确接收的像素,也依赖于已经隐藏处理过的像素. 为了便于描述,“可用像素”表示正确接收的像素或者已经隐藏过的像素;如果某种块丢失情形在接收图像中重复出现,本文称之为“规则”丢失. 现有的错误隐藏算法均能很好地应对低丢失率的情形. 根据受损块在图像中的整体分布情况,可归为 4 种丢失模式:规则间隔丢失模式、规则连续丢失模式、随机突发丢失模式、随机行丢失模式.

对于图像块的丢失,图 1 展示了 4 种具有代表性的丢失模式,图中每个方块代表一个 8×8 或 16×16 或其他像素大小的图像块,正确接收的块用白色标记,受损块用黑色标记. 如图 1(a)所示,规则间隔丢失模式是一种广为运用的丢失模式,受损块在图像中的分布比较离散,其 8 个相邻块都已经被正确接收,由于可用的邻域信息比较充分,许多错误隐藏算法都能对此种丢失模式进行有效的恢复. 图 1(b)给出了规则连续丢失模式的块分布情形,其中,当前受损块的 4 个相邻块被正确接收. 图 1(c)给出了随机突发丢失模式的示意图. 图 1(d)展示了随机行丢失模式的一个示例,受损块集中出现在图像中的某些整行,由于受损块水平方向的邻域信息已经损失,可资利用的相关信息较少,许多隐藏算法针对此丢失模式都达不到满意的效果.

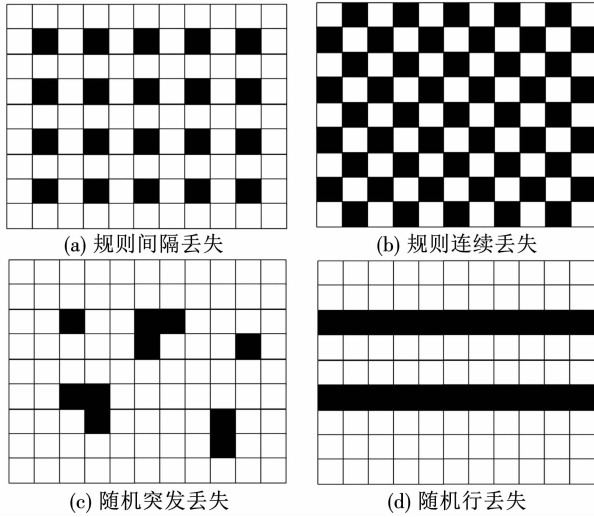


图 1 典型的丢失模式. 正确接收的块采用白色方块标记, 受损块采用黑色方块标记

Fig. 1 Typical loss patterns. White squares denote the correctly received blocks and black squares denote the missing blocks.

空域错误隐藏的主要应用场合是图像通信的实时后处理. 对于现有的空域错误隐藏机制, 一方面, 插值机制的计算复杂度较低, 但图像细节恢复的效果一般; 而稀疏表达机制的计算复杂度较高, 不太适合实时性要求严格的场合. 另一方面, 许多空域错误隐藏算法只针对特定的丢失模式进行优化, 难以适应不同的丢失模式. 因此, 有必要在通用性、计算复杂度和恢复质量之间寻求更具竞争力的算法.

1.2 最大后验预测

延拓区域的像素相关性通常被建模为一个条件随机过程, 空域错误隐藏可以采用最大后验准则来估计当前受损块的像素^[4,12]. 从统计学的观点来看, 错误隐藏问题可以等价于在一个延拓区域内, 通过一个可用像素子集 $\{q_m\}$ ($1 \leq m \leq M$) 的信息来预测当前丢失像素子集 $\{p_k\}$ ($1 \leq k \leq K$). K 和 M 的大小关系取决于不同的隐藏策略. $\{p_k\}$ 的最大后验估计能表示为最大化联合条件概率 $\rho(p_1, \dots, p_K | q_1, \dots, q_M)$, 但是 K 或 M 的取值范围通常是比较大的, 难以进行准确地建模, 在实际应用中获得这一联合条件概率几乎是不可能的. 由于一个延拓区域通常是一幅图像的小块区域, 往往能够近似成为一个准平稳过程, 上述最大后验估计的简化策略是基于线性加权的方式来预测每个丢失像素.

$$p_k = \begin{cases} \sum_{m=1}^M \beta_m \cdot q_m, & k = 1; \\ \sum_{n=1}^{k-1} \alpha_n p_n + \sum_{m=1}^M \beta_m \cdot q_m, & 2 \leq k \leq K. \end{cases} \quad (1)$$

对于公式(1)中的加权系数 α_n 和 β_m , 自适应系数相比于固定系数能更好地表达图像的局部相关

性. 作为错误隐藏问题的一个一般性假设, 空域错误隐藏算法大都假设合成的边缘在穿越受损块时是准直线的, 因此可利用多方向插值来恢复受损边缘. 在光栅扫描顺序下, 多方向插值逐一地估计受损块在不同方向上的像素相关性, 然后对邻域像素值与相关系数进行加权叠加. 每个当前受损块对应着一个延拓区域. 如图 2 所示, 一个延拓区域 A 含有 3×3 个块: 当前受损块 C 位于中心, 其它 8 个块分别属于以下 3 种类型之一: 正确接收的无错块 E 、已经执行错误隐藏的已隐藏块 R 、还未进行错误隐藏的后续受损块 S .

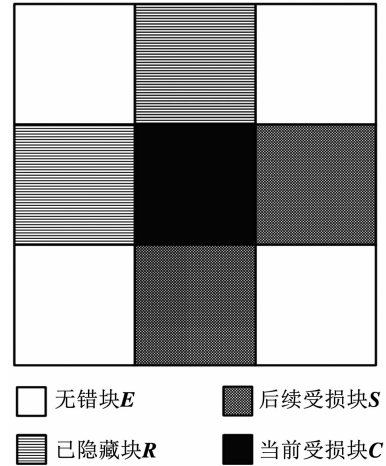


图 2 一个延拓区域 A 的例子, 该延拓区域包含当前受损块 C 、无错块 E 、已隐藏块 R 、后续受损块 S

Fig. 2 Example of an extrapolation region A as union of the current missing block C , errorless block E , recovered block R , and subsequent missing block S

2 所提算法

本文提出了一种非迭代收缩多方向预测 (Non-iterative Shrinkage Multi-directional, NSM) 算法, 所提算法吸收了现有多方向插值与稀疏模型学习的优点, 通过引入自适应的邻域梯度特征学习机制, 高效地获取受损块的邻域信息, 随后通过精细设计的收缩填充次序进行逐像素的多方向插值.

2.1 基于梯度特征的自适应学习

现有的空域错误隐藏算法表明, 根据梯度特征对受损块进行自适应插值, 可以较好地保持图像的结构和边缘信息. 所提 NSM 算法通过在当前延拓区域的无错块 E 和已隐藏块 R 中执行梯度特征统计过程, 累计当前延拓区域的梯度分布; 基于无错块 E 和已隐藏块 R , 分别累计 8 个方向中各方向上的梯度幅值. $p_{i,j}$ 表示位于坐标 (i, j) 的像素, 梯度检测可以表示如下:

$$\nabla_{i,j} = \begin{bmatrix} g_x(i,j) \\ g_y(i,j) \end{bmatrix} = \begin{bmatrix} \partial p_{i,j} / \partial x \\ \partial p_{i,j} / \partial y \end{bmatrix}. \quad (2)$$

式中, $\mathbf{g}_x(i, j)$ 和 $\mathbf{g}_y(i, j)$ 分别表示在 x 水平方向和 y 垂直方向的像素强度变化率. 现有空域错误隐藏算法的梯度检测采用了各向异性的 Sobel 或 Canny 算子, 以利于检出较为准确的边缘信息. 而本文算法仅需统计受损块邻域的总体梯度特征, 各个方向的像素之间距离越远、相关性越小, 掩模位置的加权系数应与到中心像素的距离成反比, 而与方向无关, 各向同性梯度检测器更有利于实现这一目标. 因此, 本文采用的各向同性梯度检测器含有两个 3×3 像素的梯度掩模算子, 中心像素 $p_{i,j}$ 的梯度分量 $\mathbf{g}_x(i, j)$ 和 $\mathbf{g}_y(i, j)$ 对其邻域执行如下两个 3×3 像素的梯度掩模算子.

$$\mathbf{g}_x(i, j) = \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}, \quad (3)$$

$$\mathbf{g}_y(i, j) = \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}. \quad (4)$$

基于 NSM 算法, 图 3 所示的例子说明了如何利用各向同性梯度检测器来计算像素 $p_{i,j}$ 的水平梯度 $\mathbf{g}_x(i, j)$ 和垂直梯度 $\mathbf{g}_y(i, j)$, 其中, 各向同性的梯度掩模算子用于计算各个延拓区域的水平或垂直梯度分布. 各个邻域像素对中心像素的影响效应并不相同, 所以不同的掩模位置有着不同的加权系数.

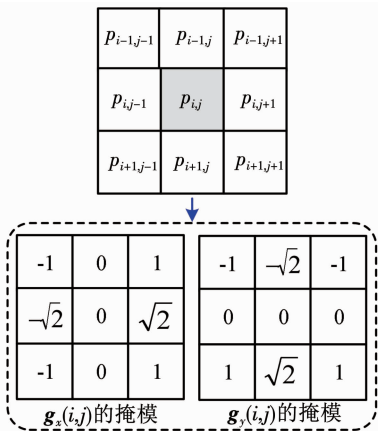


图 3 各向同性梯度检测器

Fig. 3 Isotropic gradient detector

在获取中心像素 $p_{i,j}$ 的梯度分量 $\mathbf{g}_x(i, j)$ 和 $\mathbf{g}_y(i, j)$ 之后, 坐标 (i, j) 处的梯度幅值可通过如下公式推导:

$$|\nabla_{i,j}| = \sqrt{[\mathbf{g}_x(i, j)]^2 + [\mathbf{g}_y(i, j)]^2}. \quad (5)$$

$|\nabla_{i,j}|$ 表示在像素坐标 (i, j) 处的梯度幅值, 对应的梯度角度表示为

$$\theta_{i,j} = \arctan[\mathbf{g}_y(i, j)/\mathbf{g}_x(i, j)]. \quad (6)$$

研究表明, 超过 8 个预测方向已难以提升错误

隐藏的恢复质量^[7]. 因此, 每个梯度角度被归入到 8 个方向中的某一角度, 每个方向的角度范围是成对的 22.5° . 在当前延拓区域的无错块 E 和已隐藏块 R 中, 每个像素对应的梯度幅值和梯度角度分别通过式(5)和式(6)进行计算, 由此为 8 预测方向累计梯度幅值. 在获得延拓区域的梯度分布信息之后, 非迭代收缩填充过程将对当前受损块的所有像素逐一执行多方向预测.

2.2 非迭代收缩填充

对于当前受损块中的不同像素, 它们的填充先后次序会影响最终的恢复质量, 因为已经隐藏过的像素可能导致错误扩散. 对于由外至内的受损块像素填充, 现有的空域错误隐藏算法预设了与受损块无关的像素填充次序, 并没有考虑不同的丢失模式应具有差异化的像素填充次序. 因此, 本文算法在像素级可用度的基础上, 新增了邻域级可用度的定义, 从而设计出模式自适应的“收缩填充次序”. 在大多数情况下, 正确接收的像素和已经完成错误隐藏的像素有着相似的利用价值. 对于任一像素 p_l , 它的像素级可用度 $\alpha(p_l)$ 被定义为

$$\alpha(p_l) = \begin{cases} 1, & \text{若 } p_l \text{ 被正确接收或已隐藏;} \\ 0, & \text{若 } p_l \text{ 是丢失或未知.} \end{cases} \quad (7)$$

如果某个像素已经被正确接收到, 它的像素级可用度被设置为 1. 对于一个丢失像素, 它的像素级可用度被初始化为 0; 在进行错误隐藏之后, 它的像素级可用度被设置为 1. 一般而言, 一个丢失像素的恢复质量会随着它的 16 邻域像素中可用像素的数量增加而提高. 对于一个丢失像素 X_k , 它的邻域级可用度 $\beta(L_k)$ 被定义为在它的 16 邻域像素 L_k 中所有像素级可用度之和

$$\beta(L_k) = \sum_{p_l \in L_k} \alpha(p_l). \quad (8)$$

有着更高邻域级可用度的像素应该被赋予更高的优先权. 在当前受损块里, 有着相等邻域级可用度 $\beta(L_k)$ 的所有像素形成一个像素组, 不同的像素组按照一组接一组的顺序进行隐藏. 根据当前受损块内像素组的邻域级可用度, 非迭代收缩填充首先恢复第 1 个像素组, 它的像素有着最高的邻域级可用度. 当第 1 个像素组内的所有像素按照先验填充准则进行错误隐藏之后, 非迭代收缩填充将恢复第 2 个像素组, 它的像素在受损块的所有像素中有着第二高的邻域级可用度, 第 2 个像素组的所有像素同样按照先验填充准则进行错误隐藏, 以此类推. 光栅扫描顺序可能导致图像中的错误大致地从左上块向右下块进行扩散. 一个像素组内的所有像素具有相等的邻域级可用度, 各像素的填充顺序需要预先设

定,图 4 给出了本文所提的先验填充准则示意图,图中每个方框内的数字表示相应像素的填充次序。

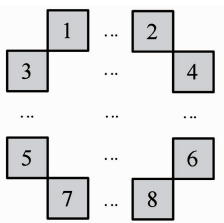


图 4 一个像素组内所有像素的先验填充准则。方框内的数字越小,表示该像素越早进行填充

Fig. 4 Priori filling rule for pixels in a pixel group. The smaller the number is, the earlier the pixel is filled.

在当前受损块内,位于外层的像素组由于有着较高的邻域级可用度,通常具有更高的优先级,因此收缩填充次序呈现出从外层像素组到内层像素组的大致趋势。图 5(a)给出了对尺寸为 8×8 像素的当前受损块进行非迭代收缩填充的一个例子,当前受损块的相邻块都被正确接收,图中颜色越深的像素越晚进行错误隐藏,各像素上对应的数字表示该像素的填充次序。图 5(b)给出了当前受损块在非迭代重构过程中的次序幅度图,有着更高幅度的像素将越晚进行错误隐藏。一组接一组的像素组重构将大致从受损块的外层向内层进行,同时,丢失像素按照先验填充准则逐个进行恢复,这种像素级的恢复顺序被称为“收缩填充次序”。

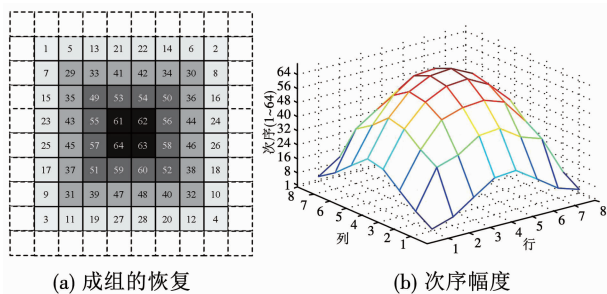


图 5 非迭代收缩填充的一个例子,一个 8×8 块遇到规则间隔丢失模式(一个小方块代表一个像素)

Fig. 5 An illustration of non-iterative shrinkage filling for an 8×8 block with regular isolated loss pattern (one square denotes one pixel)

梯度检测是为了获取延拓区域的梯度分布,丢失像素与邻域像素之间存在 8 个主要的预测方向。对于每个延拓区域,NSM 算法利用无错块 E 和已隐藏块 R 的梯度特征统计,对多方向预测器的加权系数进行局部自适应调整。在恢复当前丢失像素时,可用像素的加权系数正比于当前丢失像素与该可用像素所在方向上的梯度幅值,当存在多个预测方向时,多方向预测器按照各方向上的梯度比例进行联合加权,使恢复的图像块具有与延拓区域类似的几何结

构特征。NSM 算法为每个丢失像素提供了一个基本隐藏单元,该单元由一个丢失像素周围的 16 邻域像素组成,存在 8 个等间隔的预测方向。基于一个基本隐藏单元,每一丢失像素利用在 8 个方向的可用像素及其权重进行最大后验预测。 c_i^k 表示在当前丢失像素 X_k 和它的基本隐藏单元可用像素 p_i 之间方向上的梯度幅值,NSM 算法通过多方向邻域像素的加权叠加来恢复每个丢失像素:

$$X_k = \sum_{p_l \in L_k} \delta_l \cdot p_l = \sum_{p_l \in L_k} \left(\frac{c_l^k}{\sum c_l^k} \right) p_l. \quad (9)$$

式中, L_k 代表在丢失像素 X_k 的基本隐藏单元中可用像素集合, δ_l 是一个正比于梯度幅值 c_l^k 的加权系数,NSM 算法通过基本隐藏单元的可用像素来预测丢失像素。在执行多方向预测器时,非迭代收缩填充偶尔会面临一种极端的情况:对于某一像素的多方向预测,式(9)的所有 $\delta_l p_l$ 项都等于 0。为了解决这一特殊问题,该像素将采用相邻块中与之最近的可用像素直接进行估计。填充次序与丢失像素的估计值无关,收缩填充的次序能够在当前块恢复之前确定下来。低复杂度的 NSM 算法能够避免严重的块效应,在受损边缘之间生成较为平滑的过渡效果。

2.3 复杂度分析

下面分析 NSM 算法的计算复杂度。在当前延拓区域 A 中,无错块 E 和已隐藏块 R 的可用像素被用于估计当前受损块 C 的梯度分布,计算开销主要是在每个可用像素处进行梯度检测,各向同性梯度检测器的计算复杂度取决于参与梯度掩模算子的像素数量 a 。假设块大小是 s^2 个像素, E 和 R 的块数量是 b ,则一个延拓区域包含 $b \times s^2$ 个可用像素。因此,梯度特征学习的计算复杂度为 $O(a \times b \times s^2)$,各像素的梯度可供多个相邻块使用。为了恢复当前受损块 C ,每一丢失像素执行多方向插值,相应的计算复杂度取决于该像素的 16 邻域像素 L_k 中可用像素的数量 d 。对于一个受损块,非迭代收缩填充的计算复杂度为 $O(d \times s^2)$ 。令 $g = a \times b + d$,NSM 算法的计算复杂度为 $O(g \times s^2)$ 。

类似的,非迭代类 AVC、CAD、BBF 或 EGI 算法的计算复杂度均可表示为 $O(q \times s^2)$,通常 g 或 q 都是较小的整数值。AVC 算法的计算复杂度为 $O(s^2)$,其实时性最好;BBF 算法的 q 值与 NSM 算法的 g 值基本相当。CAD 算法采用了三种不同的隐藏算法进行错误掩盖,需要预判受损块的类型,内容自适应的预判流程增加了实现难度。EGI 算法需要执行自适应的边缘提取,然而寻找所有边缘的准确位置是比较耗时的。因此,CAD 与 EGI 算法的 q 值相对较高。当前恢复质量最好的空域错误隐藏算法采用了迭代

逼近机制,然而每个像素的迭代重构过程是非常耗时的,迭代循环的次数往往超过 100 次^[10-12],虽然迭代过程可以通过穷举法搜索出最匹配的候选者,但这一过程往往导致大量无效的重复操作,计算复杂度由此至少增长了一个数量级。

3 实验结果

为了评估 NSM 算法,本文在标准测试图像上进行了大量实验,块尺寸采用现有错误隐藏算法常用的 8×8 和 16×16 像素,并采用图 1 所示的 4 种典型丢失模式:规则间隔丢失模式(丢失率 $\approx 25\%$)、规则连续丢失模式(丢失率 $\approx 50\%$)、随机突发丢失模式(丢失率 $\approx 20\%$)、随机行丢失模式(丢失率 $\approx 15\%$),这些模式也为其他错误隐藏算法所采用。随机数生成器采用同一缺省种子,以便针对不同图像产生重复的随机突发或随机行丢失。下列测试图像均来自 USC-SIPI 图像数据库^[13]: Aerial、Boat、Couple、Elaine、House、Lena、Mandrill、Peppers、Barbara、Lighthouse、Crowd、Bridge,包含人脸、动物、植物、房屋等多种类型,格式被统一为 8 位深度的 512×512 灰度图像。NSM 算法与绪论所述的空域错误隐藏算法进行了比较,包括 AVC^[4]、CAD^[5]、HII^[6]、EGI^[7]、BBF^[8]、MMK^[9]、SLP^[10]、DSR^[11] 和 JSL^[12]。为了比较不同算法的计算复杂度,本文在同一实验平台上测试了每种算法的运行时间,该平台包括 Intel i5-4210M CPU (2.60 GHz)、8 GB 内存、Windows10 的 64 位操作系统、MATLAB R2018b。由于这些错误隐藏算法采用了相似的基本指令集,它们的 MATLAB 程序能够为算法复杂度的评估提供

一定的参考。

两种最常用的重构误差测度:峰值信噪比 (PSNR) 和多尺度结构相似度 (SSIM) 用于评价受损图像的恢复质量^[14]。PSNR 或 SSIM 的评分越高,则图像的恢复质量越好。通过在每种丢失模式下分别测试所有图像的平均性能,表 1 和表 2 比较了不同算法的恢复质量。从表中可以看出,NSM 与最好质量算法之间的平均值差距分别为 0.85 dB 的 PSNR、0.022 1 的 SSIM。相比于较低复杂度的 AVC、CAD、BBF 和 EGI 算法,NSM 算法分别取得了 0.625 dB、3.461 dB、1.952 dB 和 0.506 dB 的平均 PSNR 增益,同时取得了 0.027 6、0.051、0.035 3 和 0.013 6 的平均 SSIM 增益。

就计算复杂度而言,表 3 显示了不同算法的运行时间比较。对于每种丢失模式,各算法的运行时间是恢复每一图像后的平均值。从表中可以看出,AVC、CAD、BBF、EGI 和 NSM 可以被归为低复杂度的错误隐藏算法,而其它迭代类算法的运行时间至少增长了一个数量级。AVC 算法的实时性最好,但其平均 PSNR 和平均 SSIM 仅高于 CAD 算法,恢复质量相较于其他低复杂度算法处于劣势。BBF 算法比 NSM 算法稍快,但由于没有引入局部自适应的梯度特征学习机制,导致恢复质量较差。除了 AVC 和 BBF 算法,所提 NSM 算法相比于其他错误隐藏算法始终需要更少的运行时间,计算复杂度比迭代类算法低了一个数量级以上。NSM 算法无需实测数据进行模型训练,对不同的丢失模式具有良好的通用性,其平均 PSNR 和平均 SSIM 在低复杂度算法中表现最好。

表 1 在各种丢失模式下,各种错误隐藏算法的平均 PSNR 比较										
Tab. 1 Comparison of average PSNR of spatial error concealment algorithms with various loss patterns										dB
丢失模式	算法									
	HII	MMK	SLP	DSR	JSL	AVC	CAD	BBF	EGI	NSM
规则间隔 8×8	31.008	30.566	30.715	30.366	30.868	29.060	29.615	28.002	29.476	29.887
规则连续 8×8	27.665	24.142	24.181	26.885	27.552	26.047	24.457	23.303	26.399	26.804
随机突发 8×8	34.476	34.106	34.185	33.805	34.495	32.796	31.857	31.382	33.079	33.156
随机行 8×8	25.063	24.852	24.917	23.641	25.455	23.735	18.642	22.831	24.389	24.997
规则间隔 16×16	27.934	28.722	28.796	28.672	28.425	27.119	26.646	26.157	26.953	27.705
规则连续 16×16	25.142	25.328	25.379	25.375	25.404	24.092	20.390	22.675	23.683	24.690
随机突发 16×16	31.857	33.074	33.039	33.018	32.738	31.240	27.593	30.219	31.213	31.601
随机行 16×16	22.368	23.548	23.625	22.995	23.278	22.329	14.526	21.233	22.180	22.574
平均值	28.189	28.042	28.104	28.095	28.527	27.052	24.216	25.725	27.171	27.677

表 2 在各种丢失模式下,各种错误隐藏算法的平均 SSIM 比较

Tab.2 Comparison of average SSIM of spatial error concealment algorithms with various loss patterns

丢失模式	算法									
	HII	MMK	SLP	DSR	JSL	AVC	CAD	BBF	EGI	NSM
规则间隔 8×8	0.896 7	0.892 6	0.894 1	0.893 8	0.898 5	0.863 9	0.874 9	0.855 8	0.871 8	0.879 6
规则连续 8×8	0.785 5	0.677 3	0.679 2	0.770 4	0.787 5	0.727 1	0.712 6	0.657 1	0.737 4	0.755 4
随机突发 8×8	0.955 6	0.954 6	0.955 2	0.954 6	0.957 5	0.944 1	0.945 4	0.939 5	0.946 9	0.949 7
随机行 8×8	0.688 1	0.702 1	0.703 4	0.694 8	0.721 1	0.626 3	0.568 1	0.640 6	0.676 6	0.693 8
规则间隔 16×16	0.861 6	0.873 5	0.874 3	0.870 7	0.871 1	0.833 3	0.834 0	0.835 3	0.838 9	0.852 0
规则连续 16×16	0.725 2	0.740 3	0.739 6	0.735 3	0.742 6	0.670 0	0.624 0	0.660 2	0.676 9	0.708 7
随机突发 16×16	0.944 9	0.950 1	0.950 1	0.949 6	0.950 2	0.936 6	0.932 3	0.937 3	0.938 3	0.943 4
随机行 16×16	0.639 3	0.681 6	0.681 8	0.675 8	0.680 8	0.609 7	0.532 2	0.623 9	0.636 4	0.649 2
平均值	0.812 1	0.809 0	0.809 7	0.818 1	0.826 1	0.776 4	0.753 0	0.768 7	0.790 4	0.804 0

表 3 在各种丢失模式下,各种错误隐藏算法的平均运行时间比较

Tab.3 Comparison of average runtime of spatial error concealment algorithms with various loss patterns

s

丢失模式	算法									
	HII	MMK	SLP	DSR	JSL	AVC	CAD	BBF	EGI	NSM
规则间隔 8×8	138.75	131.39	181.19	217.88	161.38	0.05	4.41	1.35	4.66	3.04
规则连续 8×8	285.89	353.32	560.18	1004.93	589.33	0.16	10.18	3.40	17.03	10.88
随机突发 8×8	127.55	114.98	148.61	205.71	123.33	0.05	5.11	1.02	4.17	2.62
随机行 8×8	591.74	524.82	679.46	1011.13	590.35	0.17	12.78	4.45	18.66	11.47
规则间隔 16×16	275.59	1 040.03	1 349.70	785.20	283.78	0.03	20.64	11.99	11.08	6.98
规则连续 16×16	547.89	1 529.72	1 981.96	1 275.21	558.27	0.05	27.27	15.62	18.24	11.24
随机突发 16×16	101.50	376.45	487.79	288.13	104.13	0.02	17.59	4.27	4.70	2.90
随机行 16×16	537.02	1 816.28	1 417.88	638.29	246.78	0.02	17.41	18.74	22.14	12.42
平均值	325.74	735.87	850.85	678.31	332.17	0.07	14.42	7.61	12.58	7.70

4 结 论

在现有的空域错误隐藏算法中,运行快速的多方向插值算法在处理多种丢失模式时效果不甚理想,而恢复质量高的迭代类算法又难以满足实时性需求. 为了改进当前空域错误隐藏的综合性能,本文提出了一种高效的非迭代收缩多方向预测算法,通过各向同性梯度检测器学习每个受损块邻域的梯度特征,随后根据精细的收缩填充次序逐一对丢失像素进行非迭代重构. 所提算法在各种丢失模式下均能够获得良好的综合性能,在通用性、计算复杂度和恢复质量之间取得了一种具有竞争力的性能折衷.

参考文献

[1] USMAN M, HE Xiangjian, XU Min, et al. Survey of Error Concealment techniques: research directions and open issues[C]// Proceedings of 2015 Picture Coding Symposium. Cairns, Australia: IEEE, 2015: 233. DOI: 10.1109/PCS.2015.7170081

[2] CARREIRA J F M, ASSUN O P A, DE FARIA S M M, et al. Error

concealment-aware encoding for robust video transmission[J]. IEEE Transactions on Broadcasting, 2019, 65(2): 282. DOI: 10.1109/TBC.2018.2865644

[3] KORHONEN J. Study of the subjective visibility of packet loss artifacts in decoded video sequences[J]. IEEE Transactions on Broadcasting, 2018, 64(2): 354. DOI: 10.1109/TBC.2018.2832465

[4] KUMAR S, XU Liyang, MANDAL M K, et al. Error resiliency schemes in H. 264/AVC standard[J]. Journal of Visual Communication and Image Representation, 2006, 17(2): 425. DOI: 10.1016/j.jvcir.2005.04.006

[5] ZHANG Rongfu, ZHOU Yuanhua, HUANG Xiaodong. Content-adaptive spatial error concealment for video communication[J]. IEEE Transactions on Consumer Electronics, 2004, 50(1): 335. DOI: 10.1109/TCE.2004.1277882

[6] CHUNG B J, YIM C H. Hybrid error concealment method combining exemplar-based image inpainting and spatial interpolation[J]. Signal Processing: Image Communication, 2014, 29(10): 1121. DOI: 10.1016/j.image.2014.09.009

[7] HWANG B, JO J, RI C. An improved multi-directional interpolation for spatial error concealment[J]. Multimedia Tools and Applications, 2019, 78(2): 2587. DOI: 10.1007/s11042-018-6362-1