

文章编号: 1672-8785(2023)02-0041-08

基于降码率设计的图像编码 处理技术研究

李 耿 刘 怡 占 力 杨家斌 张 翔 马蒙蒙

(西南技术物理研究所, 四川 成都 610046)

摘 要: 为了降低老化的图像采集设备引入的加性噪声所带来的图像数据冗余, 研究了加性噪声对图像和视频的影响, 提出用三维块匹配(Block Matching 3D, BM3D)算法处理受噪的静态图像, 验证了通过降噪来降低图像信息熵和数据量的方法, 进而提出降噪-编码方案。首先采用四维视频块匹配(Video Block Matching 4D, VBM4D)算法对受噪视频进行降噪处理, 随后进行 H.264 编码。经过此方案处理的图像信息熵降低 16%, 数据量减少 38%; 在视频质量未显著降低的前提下, 编码码流减少 59.5%。数据表明该方法能在保证视频质量的前提下显著提升 H.264 的编码压缩率, 促进视频在小带宽通信系统中的传输。与 H.264 以及神经网络等编码方案相比, 该方案复杂度低, 能相对实时地处理视频。

关键词: 信息熵; 加性噪声; 视频; VBM4D; H.264 编码

中图分类号: TN919.81 **文献标志码:** A **DOI:** 10.3969/j.issn.1672-8785.2023.02.007

Research on Image Coding Processing Technology Based on Reduced Bit Rate Design

LI Geng, LIU Yi, ZHAN Li, YANG Jia-bin, ZHANG Xiang, MA Meng-meng

(Southwest Institute of Technical Physics, Chengdu 610046, China)

Abstract: In order to reduce the redundancy of image data caused by additive noise introduced by aging image acquisition equipment, the effect of additive noise on image and video is studied, and the BM3D (block matching 3D) algorithm to process noisy static images is proposed in this paper. It is verified that the image information entropy and data volume can be reduced by reducing noise. Then the noise reduction-coding scheme is proposed. In this scheme, the noisy video is firstly denoised by VBM4D algorithm, and then encoded by H.264. After the scheme processing, the image information entropy is reduced by 16%, the data volume is reduced by 38%, and the encoded code stream is reduced by 59.5% on the premise that the video quality is not significantly reduced. The data show that the encoding compression rate of H.264 can be significantly improved and the transmission of video in small bandwidth communication systems is promoted by this method under the premise of guaranteeing video quality. Compared with H.264 and neural networks, this scheme has low complexity and can process video relatively in real time.

Key words: information entropy; additive noise; video; VBM4D; H.264 encoding

收稿日期: 2022-10-27

作者简介: 李耿(1995-), 甘肃平凉人, 硕士研究生, 主要研究方向为图像处理与传输。

E-mail: 2399902214@qq.com

0 引言

随着信息通信行业的高速发展, 视频传输逐渐进入通信领域。视频作为一种表达信息的综合媒介, 传输时拥有更加全面的信息^[1]。随着传输的视频清晰度变高, 视频的数据量更大, 对通信带宽资源的需求将会更高。为了实现视频传输, 首先要降低其对优良的通信带宽的高度依赖, 因此以图像压缩为核心的数字视频压缩编码技术^[2]对传输视频至关重要。显而易见, 在保证一定的视频画面质量的前提下, 传输视频最关键的是降低视频的数据冗余(统计冗余、空间冗余和视觉冗余)。降低视频数据冗余的方式主要是以有损编码为主的图像压缩技术。

当下广泛应用的 H.264 标准采用整数离散余弦变换(DCT)和熵编码两种方式减少视频中的冗余信息。众多学者在一定程度上提高了 H.264 视频编码技术的应用。

王琰等人提出了一种通过纠错控制机制来提高 H.264 编解码适应性的码率控制方法^[3], 解决了在不稳定的空间信道中 H.264 码流易丢失的问题。该方法依赖于硬件实现, 不便于软件实现。马军和王洪亮等人在华为海思 Hi3516A 平台上实现了 H.264 视频压缩, 并提出了一种基于 FPGA 和以太网接口的 H.264 码流实时传输方案^[4]。虽然该方法实现了 H.264 码流实时传输, 但是对后续码流的观察以及解码等步骤依赖于其他软件, 导致系统集成度不够。张佳将无人机遥测图像通过 H.264 压缩后实现了无人机与地面之间的无线传输^[5], 但是未在 H.264 压缩率提升等方面加以改进。以上方法均在硬件平台上实现了 H.264 标准以及传输, 并未分析噪声对 H.264 编码的影响, 更没有提出相应的措施。

本文在传统的 H.264 视频编码的基础上, 提出了编码前降噪处理的方案, 在保证视频质量的同时降低数据冗余, 进一步提高了编码压缩比。

1 提高压缩比的方法

在 H.264 压缩标准中, 改变相关参数会使得压缩率变化。通过试验发现, 增大量化步长能够明显降低视频的数据量, 从而提高压缩比。除此之外, 增加 H.264 编码 GOP(画面组)中 P 帧(参考帧)的数量也能在一定程度上降低所压缩视频的数据量, 提高压缩比。然而, 增大量化步长是以大幅度舍弃视频的有效信息熵为代价来降低视频数据量的, 因此重构视频的视觉效果很差; 增加 P 帧的数量会造成传输过程中的误码传递, 导致部分帧重构失败。显然, 上述两种方式无法在不降低或者小幅度降低目标图像有效信息熵的前提下大幅度减少视频中无效的数据冗余。

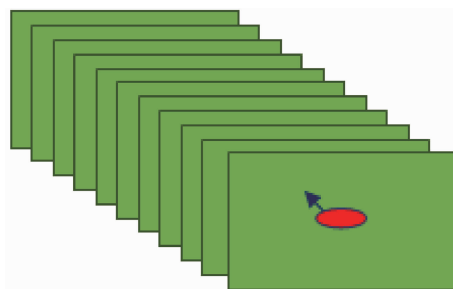


图 1 H.264 的 GOP 结构

1.1 H.264 编码关键技术介绍

H.264 中能有效压缩视频数据的关键是熵编码、帧内压缩以及帧间压缩。利用信源的统计特性进行码率压缩的编码称为熵编码。由于压缩率比较低, 一般在变换编码后再进一步压缩。H.264 中的熵编码技术主要有自适应可变长编码(Context-based Adaptive Variable Length Coding, CAVLC)和自适应二进制算术编码(Context-based Adaptive Binary Arithmetic Coding, CABAC)。其中, CAVLC 编码提高了 H.264 的编码效率, 适合实时视频传输场所(如视频会议和直播等方面); CABAC 从宏块类型开始, 包括帧内运动向量模式、各种编码标志以及图像的残留数据在内的所有图像信息熵都使用 CABAC 编码, 大大提高了数据压缩率, 但编码速率较慢。H.264 标准中的熵编码旨在代替或者舍弃图像中不易

编码的信息以降低传输视频的信息熵, 从而降低传输的视频码流。

1.2 无效噪声信息对 H.264 编码的影响

图 1 展示了 H.264 中的一个 GOP 组。当关键帧 I 帧(图 1 中带红色的帧)受噪声污染后, 由于 H.264 中关键帧的编码为无损压缩, 所以 I 帧中的噪声将会一直保留, 经过编码进入传输信道, 在解码端会影响后续 P 帧的重构。在 H.264 中进行帧内预测分析后发现, 消除视频的空间冗余是根据视频帧中区域像素的相似性和重复性来实现的^[6]。显然, 被污染图像的区域和周围像素有较大的差异。所以帧内预测并不能有效消除噪声带来的影响。帧间预测基于视频相邻帧内容的相关性^[6], 而视频采集设备中的电路容易产生随机噪声, 使得视频相邻帧之间的相关性变弱。因此, 帧间预测也无法有效消除噪声的影响。

从以上的分析得知, H.264 并不能有效去除视频中噪声带来的数据冗余。因此, 在 H.264 编码之前消除噪声带来的数据冗余对 H.264 编码有益。

目前, 应用于各行业的图像传输普遍存在一个问题: 采集设备的老化会使得采集端采集的原始视频携带大量噪声; 噪声带来的数据冗余造成视频的传输码流增大, 使得图像传输速率太低^[7]以致接收端视频显示出现卡顿现象。因此, 从降噪角度分析, 进一步降低视频压缩码流是有必要的。

图像信息具有一种人类视觉难以感知的性质。根据这种性质, 可以舍弃图像所携带的人类视觉难以观察的信息。因此, 以舍弃这部分

信息熵为代价来减少图像中的噪声污染, 能够在基本不降低图像视觉效果的同时大幅度减少图像的数据, 提高图像的压缩比^[8], 从而降低图像的信息熵。在图像处理阶段有效地应用滤波处理对于确保图像达到所需的质量至关重要^[9]。因此, 在压缩之前对图像进行滤波去噪, 可以提高图像的压缩比和传输效率。

通过对 H.264 编码阐述可知, 帧间压缩技术保证了 H.264 较大的压缩比。帧间压缩只保留了相邻帧之间的差值信息。如果噪声污染了不同帧的不同图像区域, 帧间压缩会保留大部分噪声信息, 无法将其有效滤除。仅靠帧内压缩是很难大幅度滤除视频中的噪声信息的, 所以 H.264 并不能有效减少视频中噪声带来的信息熵冗余。

1.3 H.264 在视频传输中的应用分析

从以上对 H.264 压缩方法的分析可以得出, H.264 压缩编码能够通过减小视频信息熵来达到降低视频传输码率的目的。该方式通过增大量化步长和增加参考帧来满足传输带宽的限制。但是增大量化步长会使得视频的信息严重丢失, 增加参考帧会导致误码的传递, 影响参考帧的恢复。

这两种缺陷对传输视频极其不利。如何在不带来这两种缺陷的同时从视频原本的信息熵中剔除视频采集和传输过程带来的噪声信息十分关键。根据 H.264 中熵编码技术压缩视频的原理, 从信息熵的角度探索, 如何在视频的有效信息熵中剔除噪声信息是本文降噪-编码方案的关键思路。

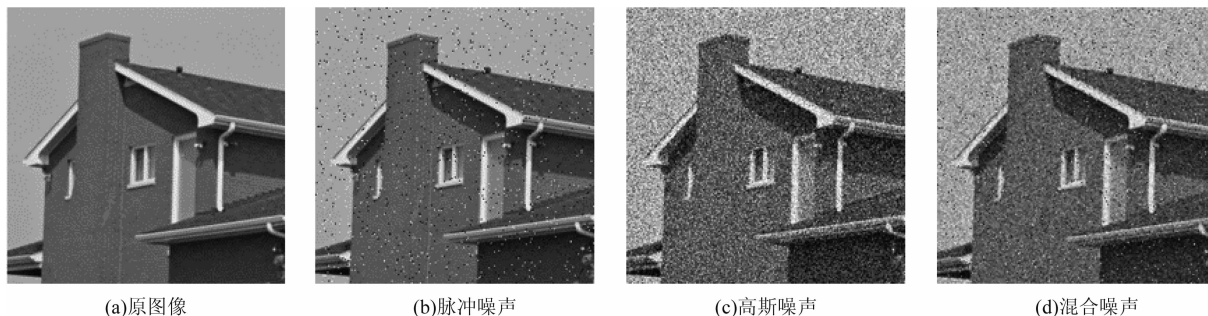


图 2 原图像与受噪图像

1.4 噪声对图像的影响

在图像的采集端和传输过程中,传感器材料属性和传输信道的影响易产生高斯噪声和脉冲噪声等加性噪声。受噪后图像的信息熵增大,导致信息量增大,进而使得表征信息量的数据量增加(见表1)。

采用小带宽通信系统传输图像的关键是图像数据量的大小,所以研究噪声对图像数据量的影响是重要的。原图像的数据量是 34.1 kB,加入脉冲噪声的图像数据量是 66.7 kB,而加入高斯噪声的图像数据量是 94.8 kB。加入这两种噪声的混合噪声后数据量增大至 95.5 kB。如图2所示,在脉冲噪声、高斯噪声以及混合噪声的污染下,原图像的细节受到了很大程度的影响。通过观察该算法对单帧图像中混合噪声的去除效果,考虑通过滤除这种混合噪声来降低视频的信息熵。

1.5 降噪设计

1.5.1 BM3D 算法去除加性噪声的效果

为了分析噪声的降低对图像信息熵和数据量的影响,首先对单帧图像进行降噪处理。BM3D是目前去除加性噪声效果最好的算法之一。该算法去噪后图像的数据量和信息熵大小如表1所示。与原图相比,受噪声污染后的图像数据量和图像信息熵都呈现上升趋势。经BM3D算法去噪后,受混合噪声干扰的图像数据量由 95.5 kB下降至 59.2 kB,信息熵由

7.7730下降至 6.5948。数据表明,降噪能有效降低噪声导致的图像数据及信息熵的增量。

1.5.2 VBM4D 算法

三维视频块匹配 (Video Block Matching 3D, VBM3D) 算法的视频去噪是基于 BM3D 算法的,主要应用于视频超分辨率、图像锐化和图像去模糊。VBM3D 没有明确以时间维度和自相关性进行块分组,而视频的时间相似性会使得数据冗余大幅度增长。针对 VBM3D 这一缺点,基于 VBM4D 算法提出的 3D 结构的时空体包含了一系列遵循特定运动轨迹的像素块。在 VBM3D 基础上, VBM4D 没有对像素块简单分组,而是对 3D 结构的时空体进行 4D 堆栈,然后进行匹配滤波。算法主要包括构建目标体、构建相似体集合以及匹配滤波。构建目标体主要有搜寻相似块、获取目标块预估位置、验证目标块与计算块的大小变化、构建块的运动轨迹以及转场控制等步骤。将相似的目标体组合后形成集合。目标体集合是匹配滤波处理的对象。匹配滤波分为硬阈值过滤聚合以及维纳滤波与聚合。文献[11]阐述了 VBM4D 算法,本文对其简要概括如下:

(1) 目标体构建

$$z(x, t) = y(x, t) + n(x, t) \quad (1)$$

式中, $y(x, t)$ 是原视频, $z(x, t)$ 是受噪视频, $n(x, t)$ 是加性噪声。VBM4D 算法包括 BM3D

表 1 图像数据

	原图	脉冲噪声图	高斯噪声图	混合噪声图	降噪图像
图像数据量	34.1 kB	66.7 kB	94.8 kB	95.5 kB	59.2 kB
图像信息熵	6.4971	6.5302	7.7770	7.7730	6.5948

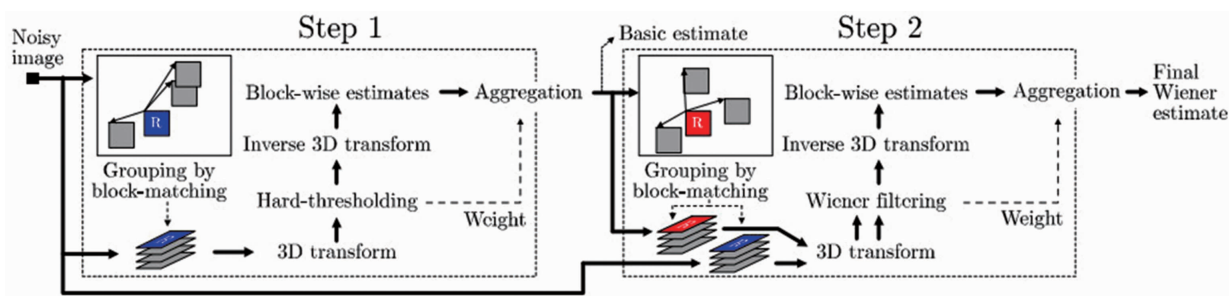


图 3 BM3D 结构^[12]

的三个基本步骤, 分别是分组、协同滤波和聚合(见图 3)。视频中的每个时间以及空间维度上的量将经过图 3 所示的算法处理流程。

(2) 时空体轨迹

时空体是按照视频场景中的特定运动轨迹沿时间维度构建的 3D 块序列。与 (x_0, t_0) 相关的轨迹定义为

$$Traj(x_0, t_0) = \{x_j, t_0 + j\}_{j=h^-}^{h^+} \quad (2)$$

式中, 元素 $(x_j, t_0 + j)$ 是时间的连续坐标, 代表参块 $B_z(x_0, t_0)$ 相邻帧 $z(X, t_0 + j)$ 内的位置; $j = -h^-, \dots, h^+$ 。在下文中, 假设对于每个 $(x, t) \in X \times T$, 轨迹 $Traj(x_0, t_0)$ 是给定的, 因此与 (x_0, t_0) 相关联的 3D 时空体积可以确定为

$$V_z(x_0, t_0) = \{B_z(x_i, t_i) : (x_i, t_i) \in Traj(x_0, t_0)\} \quad (3)$$

(3) 构建相似体集合

相似的时空堆栈体经过分组后构成 VBM4D 的元素:

$$G_z(x_0, t_0) = \{V_z(x_i, t_i) : (x_i, t_i) \in Ind(x_0, t_0)\} \quad (4)$$

式中, 假设 3D 块是沿着第四维度堆叠的。因此, 这些组是 4D 数据结构。

(4) 匹配滤波

匹配滤波通常包括以下步骤: 首先是一

个 $d+1$ 维可分线性变换应用于组, 然后利用硬阈值处理或维纳滤波缩小变换系数, 最后经过 $d+1$ 维反变换获得每个分组的估计值元素。

匹配滤波应用 4D 可分离线性变换 Γ_{4D} 在每个 4D 组 $G_z(x_0, t_0)$ 上变换, 从而提供群体估计。过程如下:

$$\hat{G}_y(x_0, t_0) = \Gamma_{4D}^{-1}(\gamma(\Gamma_{4D}(G_z(x_0, t_0)))) \quad (5)$$

式中, γ 表示通用收缩算子。过滤后的 4D 组由体积 $\hat{G}_x(x_0, t_0)$ 组成。 $\hat{G}_y(x_0, t_0)$ 的计算公式如下:

$$\hat{G}_y(x_0, t_0) = \{\hat{V}_y(x_i, t_i) : (x_i, t_i) \in Ind(x_0, t_0)\} \quad (6)$$

式中, $\hat{V}_y$ 是对原始视频 y 中的未知体积 V_y 的估计。

(5) 聚合

$\hat{G}_y$ 表示冗余视频。因为 $\hat{V}_y$ 是重叠体积, 所以匹配滤波时从不同的重叠局部估计中获得全局估计。估计中的自适应凸组合聚合权重定义为

$$\hat{y} = \frac{\sum_{(x_0, t_0) \in X \times T} (\sum_{(x_i, t_i) \in Ind(x_0, t_0)} w(x_0, t_0) \hat{V}_y(x_i, t_i))}{\sum_{(x_0, t_0) \in X \times T} (\sum_{(x_i, t_i) \in Ind(x_0, t_0)} w(x_0, t_0) X(x_i, t_i))} \quad (7)$$

体积 $\hat{V}_y(x_i, t_i)$ 支持的特征函数和聚合权重 $w(x_0, t_0)$ 对于不同的组是不同的。直观上来说, 缩小的 4D 谱 $\hat{G}_y(x_0, t_0)$ 越稀疏, 相应的权重 $w(x_0, t_0)$ 越大。

2 试验验证

本文用 VBM4D 算法实现对视频的预处理, 减少视频中的噪声信息(见图 4); 然后采用 H.264 标准压缩视频, 将编码码流在解码端进行视频重构; 接着将预处理编解码视频与原视频压缩解压后的结果进行对比, 并形成 (Peak Signal-to-Noise Ratio, PSNR) 曲线以观



(a) 原图像 (b) 去噪图像

图 4 原图像与去噪图像

察视频质量的损失; 同时计算压缩比, 比较预处理-压缩编码与压缩编码前后的码流大小, 分析该方法降低码流的效果。系统的设计流程如图 5 所示。整个处理过程是在 MATLAB 平

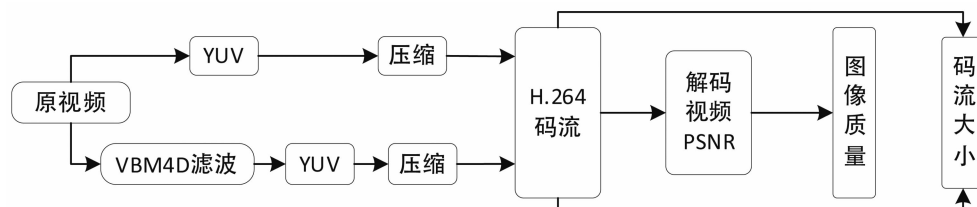


图 5 实现流程



(a)原噪声视频



(b)原视频压缩编解码



(c)用本文方法处理的视频

图 6 视频处理效果对比

台和 Visual Studio 平台上实现的。由于 MATLAB 具有强大的图像处理功能，可以直观地观察在 MATLAB 上实现 VBM4D 算法的处理结果。压缩处理是在 Visual Studio 上实现的，调用 Opencv 来实现 H.264 算法。在 Visual Studio 中对每帧视频求取 MSE（均方根误差）文件，在 MATLAB 中形成 PSNR 曲线。

3 结果分析

原视频为 avi 格式的灰度视频，画面分辨率为 320×240 ，时长为 10 s，共计 150 帧。将原视频经 VBM4D 算法处理后转换为 YUV 文件格式，然后经过 H.264 压缩编码为 .h264 文件，并在解码端进行视频重构。原视频大小为 12M，转换为 YUV 文件后大小变为 16.4M。量化步长为 28 时，原视频经压缩后得到的 .h264 文件大小为 4.0M，H.264 标准的压缩比为 4.1。经 VBM4D 算法滤波处理后的 YUV 文件大小依旧为 16.4M。经 H.264 压缩后，得到的 .h264 文件大小为 1.62M，压缩比为 10.1。相比于直接压缩原视频的压缩比，滤波后压缩视频的压缩比提高了 6.0。图 6(a)与图 6(b)分别展示了原视频和对原视频进行 H.264 编解码后的同一帧画面效果。可以看出，原视频的视觉质量并没有因 H.264 编解码而大幅下降。图

6(c)是对预处理的视频进行 H.264 编解码后的视频画面。对比图 6(b)与图 6(c)可以清楚地看到，预处理后的视频显著滤除了原视频中的噪声干扰，视频的视觉效果有较大提升。相比于原视频的 H.264 编码码流，滤波后的 H.264 编码码流显著下降（见表 2），进一步验证该方法能够有效降低 H.264 编码码流。

4 视频质量评估

为了客观评估图像处理后的图像质量，普遍的做法是计算其 PSNR 值—— R_{PSNR} 。PSNR 是指信号的最大功率与破坏性噪声功率的比值。对视频进行有损压缩编码后，由于舍弃了一些数据，所以解码端的视频恢复会有一定程度的失真。通过计算视频中每一帧图像的 PSNR 值，可以定量地评价视频帧的失真度，进而评价压缩和解压缩的功能实现情况。其算法如下：

$$R_{PSNR} = 10 \lg \left(\frac{I_{max}^2}{\sigma} \right) \quad (8)$$

$$\sigma = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (I(i,j) - K(i,j))^2 \quad (9)$$

式中， m 和 n 为图像的尺寸参数， σ 为帧内各像素点的 MSE 值之和， I_{max} 为图像像素的最大值， $I(i,j)$ 为原始图像上 i 行 j 列的像素点， $K(i,j)$ 是经过编解码后图像上 i 行 j 列的像

素点。

为了直观地观察图像质量的变化情况, 求取了处理后视频的 PSNR 曲线(见图 7 和图 8)。其中, 横轴是视频帧, 总计 150 帧; 纵轴是视频每帧画面的 PSNR 值。图 7 所示为对携带噪声的视频进行压缩编码后的 PSNR 曲线。图 8 所示为用本文提及的降噪-编码方法处理携带噪声视频后的 PSNR 曲线。在 H.264 编码过程中, 分别采用 16、22、28、34、40、46、52 的量化步长对预处理后的视频进行压缩。从图 8 中可以看出, 通过降低噪声信息熵来提高压缩比时, 视频质量并没有因量化步长的增大而急剧变差, 说明该方法适用于不同量化步长的 H.264 压缩编码图。对比图 7 与图 8 可以发现, 采用预处理-压缩编码的方法能在保留图像有效信息的同时以较大的压缩比压缩视频。

表 2 视频数据

	原视频	本文方法处理结果
视频 YUV	16.4 MB	16.4 MB
H.264 编码	4.0 MB	1.62 MB

表 2 列出了编码量化步长为 28 时采用两种不同方法处理视频的数据。可以看出, 采用预处理-压缩编码的方法将压缩比从 4.1 提高至 10.1。经压缩编码后, 视频码率由原视频的 9830.4 kbps 减小至 3276.8 kbps。经预处理-压缩编码后, 视频码率减小至 1327.1 kbps。预处理-压缩编码能显著降低视频的数据冗余, 提高视频压缩比, 有利于视频传输。根据表 2 所列的实验数据可知, 本文方法的 H.264 编码码流为 1.62 MB, 比传统的 H.264 编码码流 4.0 MB 降低了 2.38 MB, 码流数据量减少了 59.5%。从 H.264 的编码帧结构来看, 经本文方法编码视频的 I 帧和 P 帧的码流降低了近 50%。

5 结束语

基于以上视频处理结果分析以及视频传输难点研究可知, 本文提出的降噪-编码处理方法通过降噪减少图像信息熵进而降低 H.264 压

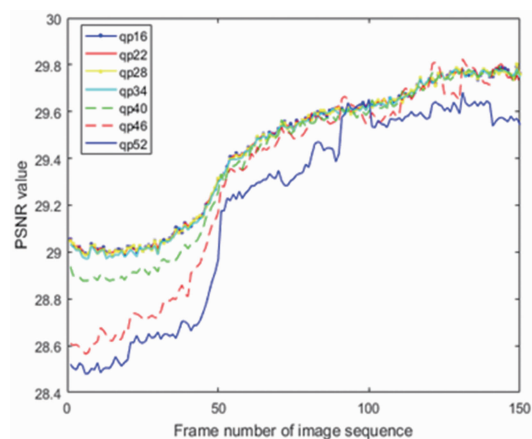


图 7 压缩编码后的 PSNR 曲线

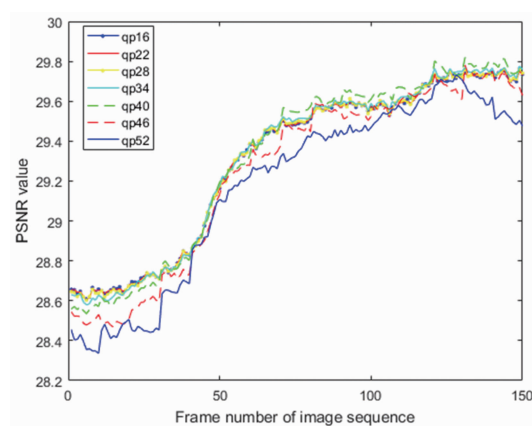


图 8 预处理-压缩编码后的 PSNR 曲线

缩码流, 能够有效地减小视频的编码数据量, 从而降低传输码流。该方法适于采用不同的量化步长进行压缩编码, 且适用于不同分辨率的视频编码处理, 可广泛应用在监控、安防等场景中, 在不增加其他成本的同时大大提高了视频传输效率。这种方法在现阶段视频传输编码方案的基础上加入了降噪处理, 进一步提高了视频的压缩率, 具有一定的应用前景。然而本文只是软件层面的实验仿真, 仍需在硬件实现方面进行研究和探索, 以促进该研究的大规模应用。

参考文献

- [1] 洪铭. 基于窄带通信的视频的压缩编码研究[J]. 福建电脑, 2016, 32(5): 14-16.
- [2] 周红, 蔡莲红, 徐光祐. 数字视频图像技术综述[J]. 多媒体世界, 1994, 1(4): 4-7.

- [3] 王琰, 于大海, 沈霁, 等. 一种非可靠信道下 H.264 码流控制方法 [J]. *数字技术与应用*, 2019, **37**(4): 62–63.
- [4] 马军, 王红亮. 基于 FPGA 的 H.264 码流实时传输系统设计 [J]. *计算机测量与控制*, 2021, **29**(5): 225–229.
- [5] 张佳. 基于 H.264 标准的无人机遥测图像传输系统设计 [J]. *自动化与仪器仪表*, 2017, **24**(6): 130–132.
- [6] 朱新山, 杨璐. 基于 H.264 与新一代视频编码标准的压缩域视频水印综述 [J]. *信息安全学报*, 2018, **3**(6): 28–41.
- [7] 李明凡, 李孟阳. 北斗多媒体信息传输系统 [M]// 中国卫星导航定位协会. *卫星导航定位与北斗系统应用*. 北京: 测绘出版社, 2017.
- [8] Burnik U. Image transmission for narrowband network connections [C]. Melbourne: International Conference on Computational Intelligence and Multimedia Applications, 1998.
- [9] 桂邦豪, 莫玮, 殷玉喆, 等. H.264 帧内预测模式的研究 [J]. *国外电子测量技术*, 2011, **30**(3): 9–12.
- [10] Coady J, Newe T, Riordan A O, et al. An overview of popular digital image processing filtering operations [C]. Sydney: 13th International Conference on Sensing Technology, 2019.
- [11] Maggioni M, Boracchi G, Foi A, et al. Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms [J]. *IEEE Transactions on Image Processing*, 2012, **21**(9): 3952–3966.
- [12] Lebrun M. An analysis and implementation of the BM3D image denoising method [J]. *Image Processing on Line*, 2012, **2**: 175–213.