一种适用于帧式和事件视频源的异步强度表示

Andrew C. Freeman
acfreeman@cs.unc.edu 北卡罗来纳大学教堂山分校，美国北卡罗来纳州

Montek Singh
montek@cs.unc.edu 北卡罗来纳大学教堂山分校，美国北卡罗来纳州

Ketan Mayer-Patel
kmp@cs.unc.edu 北卡罗来纳大学教堂山分校，美国北卡罗来纳州

arXiv:2301.08783v1 [cs.CV] 20 Jan 2023

摘要

仿生“事件”相机旨在通过异步感知模拟人类视觉系统，为高速和高动态范围应用开辟了新的领域。然而，研究人员在应用中往往要么将事件数据重新转换为帧式表示，要么针对特定相机的事件数据类型开发专用应用。为了开启视频系统的新时代、适应新型事件相机设计，并探索异步视频在传统应用中的优势，我们认为亟需一种与数据源无关的异步视频表示。本文提出一种适用于帧式和非帧式数据源的新型异步强度表示。我们展示了，与基于网格的表示相比，该表示能够提高强度精度，并大幅减少每像素采样数。对于帧式数据源，我们证明，通过对相似像素值进行时间平均而允许少量损失，可以将表示的采样率降低一半以上，同时 VMAF 质量评分仅下降 4.5。我们还展示了，在保持2000× 时间分辨率的同时，该方法融合帧式数据与事件相机数据并将其转码为强度表示时，延迟低于最先进的方法。我们认为，该方法具备构建面向事件相机的实时强度应用所需的计算效率和时间粒度。

1 引言

本文介绍了一种将帧式视频和现代事件相机数据转码为新型异步视频表示的方法。该表示能够根据各个像素的时间稳定性，系统地探索率失真权衡。这种方法计算效率高、可进行有损压缩，并适用于实时视觉应用。

对于**帧式视频** 表示，我们的方法旨在解决以下不足。首先，采样率是固定的，并且在整个帧中保持一致，这意味着视频中动态（变化显著）的空间区域，与相对静态（不变）的空间区域具有相同数量的强度样本。因此，原始表示中存在大量数据冗余，目前只能在压缩器层面加以处理。

允许免费为个人或课堂使用而制作本作品部分或全部内容的数字或纸质副本，前提是这些副本不用于营利或商业利益，也不进行分发，并且副本首页须附上本声明及完整引文。本作品中第三方组件的版权必须得到尊重

如需用于其他用途，请联系所有者/作者。，
，©2023 版权所有归所有者 / 作者所有。
ACM ISBN 。

其次，帧式视频的精度受单个图像帧所能表示的最大值限制，通常每个颜色通道为 8 至 12 位。然而，目前没有方案能够以高于强度随时间动态变化的像素的精度，来表示跨越多个帧的时间稳定的强度。我们认为，若将帧式视频转换为异步表示，其精度和时间速率控制都将得到改善。

此外，我们还解决了**事件传感器**在表示方面的不足。这些传感器不会记录传统图像帧；相反，每个像素都会独立记录一系列事件，当像素的对数强度增加或减少一定量时，就会触发一种特定的基于时间的事件。这种技术可实现极高的时间精度（约为微秒量级）和高动态范围，但无法捕捉强度不变或缓慢变化的数据。因此，许多此类相机的用户会通过帧式图像采集来补足事件传感器的信息缺失（例如，DAVIS相机 [17, 27]）。然而，这些帧图像往往会出现运动模糊，而且由于跨模态融合信息的计算速度较慢，通常会与事件数据分开处理。此外，每个事件表示的强度变化相对于前一个事件而言，因此这种原始表示不支持有损压缩，也无法体现率失真权衡。这些传感器在视觉任务和高速机器人技术领域迅速普及，但其惊人的数据速率（每秒高达数百万个事件）以及不断提高的空间分辨率（现已达到1080p），尚未从系统视角得到充分解决。相反，多数研究应用只是将事件数据转换为若干种基于帧的表示。这些表示大幅降低了数据的时间分辨率，削弱了事件感知的一项主要优势。

我们提出如下类比。设想不同厂商的传统相机各自使用专有的原始数据格式，彼此之间没有通用的表示方式。再设想，许多视频应用只能处理特定厂商相机的原始数据，而其他所有视频应用只能处理由相机原始表示生成的 PNG图像序列。这实际上就是当今事件相机生态系统的现状：事件视频没有类似 H.26X [48] 的编解码器，这意味着事件相机系统没有采用任何重要的有损压缩，而且是针对特定相机的原始数据格式定制构建的。我们认为，要构建最高效的实时事件视频系统，应用层的视频表示也应当是异步的，同时 支持速率驱动的有损压缩。

我们提出了一种统一视频表示，其基本数据类型是异步的**强度事件**。我们的事件并不

表示强度变化，这与事件传感器的表示不同；相反，每个事件直接编码一个强度值。因此，我们可以将帧视频数据和事件视频数据转码为同一种表示，从而解决上述数据源的缺点。我们的贡献总结如下。

- 为帧式和非帧式数据源提出一种新颖的异步视频表示和转码器。
- 评估该表示在帧式彩色视频源上的速度和视觉质量，结果显示其精度极高（最高可达 14 位，而源视频为 8 位），且率失真曲线具有可预测性。只需允许少量损失，我们就能将该表示的中位事件率降低 54.3%，而 VMAF 质量的中位数仅下降 4.5。
- 评估各种转码器参数对码率和质量的影响。
- 评估该表示在有无额外帧式数据的事件相机源上的速度和码率特性，并展示不同码率下的视觉差异。当目标时间精度达到每秒 500 个样本或更高时，与最先进的帧／事件融合方法相比，我们的方法延迟更低。事实上，对于不断变化的强度，我们的方法可保持微秒级 时间分辨率。我们还发布了首个完全开源的这一最先进融合方法实现，并进行了大量改进和优化，以实现高性能。
- 发布我们的开源转码器框架，其中包括 用于可视化回放我们的视频表示的实用工具。

2 相关工作

2.1 帧式视频

在帧式视频采集范式下，相机在相同的时间内曝光整个成像平面，从而生成一系列离散的二维图像，如图 1b 所示。尽管帧式视频表示得益于数十年的压缩研究，可满足各种存储需求和流媒体应用，但仍存在诸多局限。

首先，动态范围与源的捕获灵敏度密不可分，这为该表示的强度范围设定了实际上限 [44]。通常，人们会以较高位深度（最高 12 位）录制视频，随后制作母版，以便在标准动态范围和高动态范围显示器上观看 [23]。母版制作过程通常会调整亮度级别，以呈现所需的细节。在这种情况下，精度更高的表示能让电影制作人在后期处理图像暗部和亮部时拥有更大的灵活性。此外，高精度还有助于减弱强度梯度区域内显而易见的边缘。

其次，由于像素采样的帧率固定，稳定视频区域不会在多帧之间自然进行像素平均，导致高对比度和高速场景的视频暗部受到噪声的严重影响。即使超高帧率相机问世，也未能带来视频表示的新范式 [28]。相反，这类相机生成的是体积庞大、时空冗余的帧式视频文件，并使用标准编解码器。尽管某些视频容器格式允许视频采用可变帧率，但像素采样在时间上仍按帧组织 [9],，而且对这类视频的软件支持有限。

最后，大多数计算机视觉算法都依赖于对世界的帧式解读，将视频处理为一系列帧，并进行卷积 [37]。尽管这类系统在目标识别、目标跟踪和场景重建等任务上的效能已取得巨大进展，但声称其结构模拟了人眼连续、动态的感知方式并不诚实 [17]。

2.2 事件视频

近来，我们看到异步事件相机得到应用，旨在解决上述基于帧的视频采集与表示方式的不足。这些传感器不输出帧，而是采用地址事件表示（AER）输出基于时间的事件，其中每个事件表示特定像素位置和时刻的一个离散数据点。本文将考察几种事件相机及其对应的AER数据格式。

2.2.1 DVS。动态视觉传感器（DVS）优先考虑局部采集速率，而非全局精度。DVS相机（图 1a）会持续~测量入射到像素 x, y 上的对数强度 $L = \ln(I)$，直到亮度发生变化，使得 $L(x, y, t) - L(x, y, t - \Delta t) = p\theta$，其中 $p \in \{-1, +1\}$ 表示亮度变化的极性，$\theta > 0$ 表示阈值，Δt 表示自该像素上次记录到此类亮度变化以来经过的时间 [17]。当亮度变化超过阈值 $p\theta$ 时，像素会输出一个异步事件，以元组 x, y, p, t 表示。参数 θ 由多种相机增益设置以及场景照明的影响共同决定。因此，相机用户难以确定该参数，而且在视频录制过程中它可能发生显著变化。

DVS 方法具有显著优势。首先，像素可以快速触发变化事件，不会产生同步传感器的帧延迟。其次，由于强度经过对数压缩，因此能够实现高动态范围。第三，当像素处的强度相对稳定时，触发的事件很少，从而节省带宽。DVS 的主要弱点是像素噪声特性较差，且难以重建实际强度值。像素值会受到噪声影响，因为对数转换由亚阈值MOS晶体管完成，而且传感器采集的是瞬时电压采样，而不是在积分区间内累积得到的值。由于传感方式基于差分，噪声累积会导致漂移，因此难以高质量地重建实际强度值。鉴于这一缺点，DVS 传感器通常用于捕捉运动物体的边缘至关重要的场景。这类应用主要包括需要快速响应、且不重视人类视觉质量的机器人应用，例如目标检测与跟踪、手势识别以及 SLAM（同步定位与地图构建）。另一方面，文献中提出了大量基于学习的 [34, 35, 39] 和直接 [36, 38, 46] 方法，用于对 DVS 数据进行帧式重建，但即使使用最先进的硬件和低分辨率（240×180 像素）传感器，这些方法也仅能以每秒 100–1000 帧的速度进行重建 [35]。

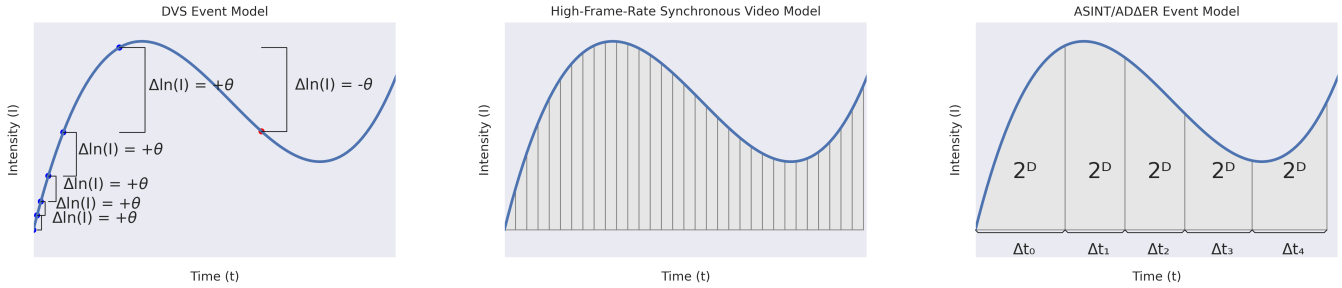


图 1：视频模型比较。(a) 每当 ln(I) 的变化达到阈值 θ时，就会触发一个新的 DVS 事件。由于传感器感知的是差值，因此强度重建颇具挑战。(b) 帧式视频在同步的时间点为每个像素提供一个样本，样本之间的间隔固定。(c) 一个 ADΔER 事件表示累积强度单位达到2^D。 D 控制每个像素的灵敏度；为便于说明，此处将其显示为恒定值。然而，在我们的 ADΔER 转码器中， D 会根据正在积分的强度变化进行调整，以降低事件率。

*2.2.2 DAVIS 和 ATIS。*动态与主动像素视觉传感器（DAVIS）旨在通过在传感器芯片上将主动像素（APS）与 DVS 像素并置，提升 DVS 技术的实用性 [17]。这些主动像素以固定速率采集离散帧，就像传统相机一样。APS 数据可以防止 DVS 准确度发生漂移，从而提高强度重建的质量 [8, 29, 30, 38]。由此得到的强度重建结果可以具有极高的动态范围，适用于重视更全面场景理解的视觉应用。

异步时间型图像传感器（ATIS）采用了不同于 DAVIS 的方法，在捕获 DVS 事件的同时获取强度。这些传感器上的积分器子像素通过输出两个 AER 式事件来表示绝对强度：两个事件之间的时间较短表示像素较亮，而间隔较长则表示像素较暗 [17]。尽管这些强度测量十分准确，但只有在 DVS 子像素记录到强度变化事件时才会触发 [32]。因此，ATIS 不会在给定时间点为每个像素提供绝对强度，而只会在高对比度运动边缘处提供。

ATIS 和 DAVIS 通过在基于对比度的事件中加入绝对强度测量，助力目标检测等视觉任务。然而，它们会产生两种截然不同的数据流，因此需要开发复杂的配套软件基础设施，将两种数据流融合起来，才能充分发挥其能力。尤其对于 DAVIS，强度帧有助于对焦相机镜头、结合经典视觉算法与基于事件的算法进行比较或增强，并通过融合事件数据与帧式数据，提升强度重建的效果。DAVIS 已成为研究领域中一种重要的基于事件的感知技术，而 ATIS 尚未获得广泛的商业应用。本文将重点关注 DAVIS 传感器。

2.2.3 事件视频表示。虽然一些应用会直接使用脉冲神经网络（SNNs）处理 DVS 事件[4, 11, 12], ，但许多应用会采用基于网格的事件表示，因为这类表示更适用于经典视觉处理流程，且无需专用的神经形态网络硬件[1, 5, 18, 26, 33, 47]。我们在表 1 中概述了其中几种基于网格的表示，并重点介绍研究人员在时间精度与处理便捷性之间所作的权衡。

尽管研究人员已探索 DVS 流的速率控制方案，但所提出的系统仅在应用层 [19], 或在相机端通过调整偏置来进行速率自适应 θ [6, 10]，而没有探索针对事件表示与传输的率失真控制方案。在 DAVIS 相机中，APS 帧流与 DVS 事件流是彼此独立的表示。研究人员可以如上所述融合主动与动态信息，但往往仍需回退到帧表示，以整合这些不同的数据流。这种做法会对事件进行时间量化，舍弃 DVS 捕获的许多高速率优势。虽然 Banerjee等人独特地提出了一种面向联网客户端的应用驱动的DAVIS速率控制系统 [2, 3], ，但该方案并未将事件信息与帧式信息统一为单一表示，这意味着这些数据流仍需各自独立的应用层逻辑。

此外，事件视觉的应用生态系统被众多原始数据表示割裂开来。数据集以多种格式发布：人类可读文本格式 [21, 27], 、适用于机器人操作系统的.bag格式 [21, 22, 27, 35], 、用于基于学习的Python任务的.hdf5格式 [20, 21],、为兼容jAER软件而设计的.aedat2格式 [20, 21], ，或用于MATLAB应用程序的.mat格式 [30, 46]。事件相机制造商iniVation和Prophesee各自提供了包含各种应用程序的软件套件，但这些套件都只能处理各自专有的相机数据格式。虽然已有一些工具可将原始相机数据转换为上述某种中间表示，但没有一种工具从根本上改变底层数据类型。我们认为，既然必须转换原始数据以供应用程序使用，不如将其转码为一种简单、快速、可压缩、非专有的数据表示，同时做到应用无关和相机无关。这样，应用程序便可读取这一统一表示，而对未来事件感知模态的支持也可通过数据转码器实现，而不必在各个应用程序中分别实现。

表 1：事件视频表示的比较。AD∆ER 表示在转码模式 (ii) 下为基于事件的应用提供了最大的灵活性，同时也可以轻松转换为基于帧的强度表示，以用于传统应用。

Name	表示	连续?	全传感器强度 强度?	可转换 为动态视觉传感器吗?	兼容 卷积神经网络吗?	兼容 脉冲神经网络吗?
DVS	$\langle x, y, p, t\rangle$	✓		–		✓
DAVIS	$\langle x, y, p, t\rangle$ $[w, h, I]$	仅动态视觉传感器	仅主动像素传感器	–	仅限主动像素传感器	仅使用动态视觉传感器
DAVIS 图像融合	$[w, h, I]$		✓		✓	
DVS 事件帧	$[w, h, \Delta L]$				✓	
DVS 时间表面	$[w, h, t]$				✓	
DVS 体素网格	$[w, h, n, \sum p$ 或 $\frac{\partial L}{\partial t}]$				✓	
ATIS	$\langle x, y, p, t\rangle, \langle x, y, I, t\rangle$	✓		–		✓
AD∆ER 事件帧	$[w, h, \frac{2^D}{\Delta t}]$		✓		✓	
DAVIS (i) → AD∆ER	$\langle x, y, D, \Delta t\rangle$	✓	✓			✓
DAVIS (ii) → AD∆ER	$\langle x, y, D, \Delta t\rangle$	✓	✓	✓		✓
动态视觉传感器 (iii) → AD∆ER	$\langle x, y, D, \Delta t\rangle$	✓		✓		✓

3 提出的表示方法

3.1 动机

虽然我们认可DVS风格表示对于基于对比度的事件很有用，但这种格式并不适合表示绝对强度。采用DVS风格的方法时，只有通过减轻传感器噪声，以及允许应用程序忽略其认为不重要的事件，才能实现有损处理。也就是说，移除某个像素事件流中的一个DVS事件，必然会影响对该像素事件流中后续事件的解读，因为每个事件传达的强度信息相对于先前的事件而言。出于同样的原因，即使是能够编码强度信息的DVS风格的ATIS表示，也不适合作为一种通用、可适配且可压缩的表示。具体而言，这些表示不支持由接收方驱动的事件有损压缩。

为此，我们提出了**地**址、**抽**取、 Δt **事**件**表**示法（**AD∆ER**，读作“adder”）格式，作为异步视频数据的“窄腰”表示。这种表示基于一种提出的异步积分（ASINT）相机模型 [15, 40, 41]。如图 1c 所示，像素 x， y， c持续对光进行积分；当累积了 2^D 个强度单位（例如光子）时，便触发一个 AD∆ER 事件 x， y， c， D， Δt。其中， D 是抽取阈值， Δt 是该像素上次触发事件以来经过的时间。我们以时钟“周期”计量 t，时钟周期的粒度可由用户调整。与 ATIS 事件不同，单个 AD∆ER 事件可直接通过 $I = \frac{2^D}{\Delta t}$ 指定强度 I。该模型的关键见解是，在转码过程中**动态地按像素控制** D。提高某个像素的 D 会降低其事件率，而降低 D 则会提高其事件率。借助这种多方面的 D 控制，我们可以确保像素灵敏度与场景动态相匹配，从而避免 ATIS 模型中的事件率和压缩问题，同时保留强度事件的优势。

尽管相关工作侧重于模拟所提出的 ASINT 传感器，以及开发早期的压缩方案和应用 [14–16]，但在这里，我们将传感器的功能与其表示区分开来。

也就是说，我们借鉴缓冲处理技术，为任意一组输入编码出最优的 AD∆ER 表示，并将 AD∆ER 作为非 ASINT 源的转码表示。借助这种表示，我们获得了一种统一格式，可据此构建面向各种相机技术的压缩方案和应用。通过针对特定源的转码器模块和抽象的 AD∆ER 库，我们可以将任意相机端（包括帧式和非帧式）的输入视频转换为单一、统一的 AD∆ER 数据表示。

3.2 AD∆ER 转码器基础

为了说明如何将数据转换为 AD∆ER，我们必须先介绍一种新颖的像素模型，以及所有转码类型共用的参数。

3.2.1 事件像素列表结构。 我们首先考察单个 AD∆ER 像素模型的积分过程。我们用链表表示 AD∆ER 像素状态。列表中的每个节点都包含一个抽取因子 D、一个强度积分测量值 I 和一个时间间隔测量值 Δt。节点之间的连接携带一个 AD∆ER 事件，该事件表示父节点在创建子节点之前所积分的强度。初始化新列表时，我们将 D 设为可用于表示我们打算积分的首个强度的最大可能值。为保持精度，列表中的值 I 和 Δt 均为浮点数。由于 AD∆ER 与数据源无关，我们将传入的值解释为通用的“强度单位”，而不将其绑定到光子等现实世界的强度度量；此外，我们可以根据需要缩放各个数据源的输入值。

让我们通过图 2 中的可视化示例来了解这种列表结构。假设我们要积分的第一个强度是 101。在图 2a 中，我们将头节点初始化为 $D = \lfloor \log_2(101) \rfloor = 6$ 。

现在，假设我们对跨越 20 个时间单位（时钟周期）的强度 101 进行积分。头节点在其 $2^D = 64$ 积分饱和前，只累积了 64/101 个强度单位。因此，这次部分积分所跨越的时间为 $(64/101) \cdot 20 = 12.67$ 时钟周期。我们可以将一个 AD∆ER 事件（不包含其空间坐标）表示为，

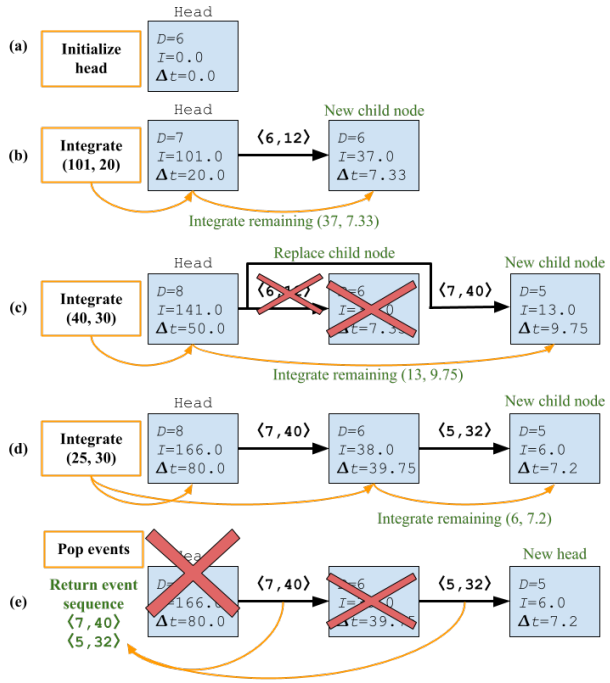


图2：连续积分下事件像素列表的结构。

如图 2b 所示，以尖括号表示的节点连接。这些事件由整数构成，因此我们使用 $\lfloor \Delta t \rfloor = 12$。此时，我们创建一个子节点来表示剩余的积分。该子节点取头节点的 D（即 $D = 6$）。我们对该子节点剩余的 $101 - 64 = 37$ 个强度单位进行积分，跨越 $(37/101) \cdot 20 = 7.33$ 时钟周期。最后，我们递增头节点的 D 值，并对剩余的 $I = 37$，$\Delta t = 7.33$ 强度进行积分。

现在，如图2c所示，让我们在30个时钟周期内积分40个强度单位。头节点的 $2^D = 128$ 积分达到饱和，因此我们替换子节点，并用一个新节点取代它，该节点由新事件 $D = 7$，$\Delta t = \lfloor 20 + 27 \cdot 40 \cdot 30 \rfloor = 40$，$(/) \cdot$ 连接。这样，我们就能尽量减少表示强度序列所需的 ADΔER 事件数，同时最大化事件的 D 和 Δt 值。与之前一样，生成子节点后，我们将剩余强度同时积分到头节点和子节点中。

最后，如图2d所示，让我们在30个时钟周期内积分25个强度单位。我们先积分头事件。它未 达到其 2^D =256的积分阈值，因此我们既不替换其事件，也不替换子节点。然后，我们以相同的强度积分子节点，使其在19个强度单位处达到 $2^D = 32$ 阈值，并额外经过 $(19/25) \cdot 30 = 22.8$ 个时钟周期。因此，我们递增 D，为子节点创建一个事件 $D = 5$，$\Delta t = \lfloor 9.75 + (19/25) \cdot 30 \rfloor = 32$，并在列表中生成另一个节点，以便在 $(6/25) * 30 = 7.2$ 个时钟周期内积分剩余的6个强度单位。

节点连接构成一个队列，当正在积分的强度发生显著变化时，转码器会输出该队列。满足此条件后，我们可以将事件从列表中出队，并将尾节点设为新的头节点，如图 2e 所示。这种链表结构最大限度地减少了表示一系列强度积分所需的 ADΔER 事件数量，并消除了 Freeman 等人 [14–16] 的工作中出现的“斑点”伪影。

3.2.2 用户可调参数。用户可以根据需要设置若干转码参数。

Δt_s：每秒刻度数。此参数定义 ADΔER 流的时间分辨率。

Δt_{ref}：标准时长积分所需的时钟周期数（例如，当源为帧式视频时，即输入帧曝光时间）。此参数与 Δt_s 一起决定准确度和数据速率。

M：ADΔER 对比度阈值。当像素接收到的归一化强度相对于其基线强度的变化超过此阈值（无论为正向还是负向）时，该像素的事件队列将被输出，其积分状态将重置，基线也将设为新的强度。与检测对数强度变化的 DVS 模型不同，我们关注的是绝对强度的变化。在本文中，M 对所有像素均为统一值，不过未来的应用可以动态调整每个像素的 M，而不会增加复杂度。下文将评估 M 对转码质量和事件率的影响。

Δt_{max}：任何事件能够跨越的最大 Δt。假设场景是静态的；此参数与事件率直接相关。也就是说，将 Δt_{max} 减半会使事件率翻倍。然而在实践中，用户只需确保 Δt_{max} 足够大，以便在视频的时间稳定区域内实现所需程度的强度平均，同时又足够小，从而使给定应用能够以足够快的速率接收像素更新。

3.2.3 二进制表示。原始 ADΔER 文件以一个文件头开头，其中包含描述视频分辨率、文件规范版本、字节序、Δt_s、Δt_{ref} 和 Δt_{max} 的元数据。紧接着文件头之后，是一系列原始 ADΔER 事件，格式如下。

x：无符号 16 位 x 地址。
y：无符号 16 位 y 地址。
c：可选的无符号 8 位颜色通道地址。如果视频为单色，则不包含 c 值。
D：无符号 8 位抽取值。
Δt：自该像素上一次事件以来经过的时间。

我们采用打包表示，因此每个事件占 9–10 字节，具体取决于是否包含颜色通道。不过，我们强调，这种事件表示具有很高的可压缩性，正如 [14, 16] 所示。我们注意到，D 的自然值范围是 $[0, 127]$,，因为在我们所用的编程语言中，2_{127} 是可表示的最大无符号整数，尽管 D 值通常处于 $[0, 30]$ 范围内。此外，我们将 $D = 254$ 保留为特殊符号，用于表示该事件的强度为零；仅凭 $D \in [0, 127]$ 无法传达这一信息。

3.2.4 实现细节。我们使用 Rust 语言实现了整个框架。非 Rust 依赖项包括用于图像导入、处理和可视化的 OpenCV[7]，以及用于帧式视频编码的 FFmpeg [13]。该框架速度快、并行度高，并经过优化，可充分利用 CPU 资源。本文所有速度评估均使用配备 8 核 AMD Ryzen 2700x CPU 的计算机，输出文件写入 RAM 磁盘。

4 帧视频源

我们首先介绍如何将 ADΔER 像素模型应用于帧式视频源。

4.1 转码器细节

我们将视频帧概念化为在固定时间段内积分得到的强度矩阵，并假设每个像素的强度仅会在帧与帧之间的瞬时发生变化（图 1b）。Δt_{ref} 定义每帧的积分时间，我们为源视频帧率 F 设置 $\Delta t_s = \Delta t_{ref} F$ 个时钟周期。由于每个 ADΔER 像素对每个输入帧恰好积分一次强度，因此我们可以轻松并行化这一过程：在多个 CPU 线程上对像素分组进行积分，然后将生成的 ADΔER 事件收集到单个向量中，在每帧积分完成后写出。由于 ADΔER 模型中的每个像素都是独立处理的，因此无需按时间交错不同像素的事件。例如，像素 A 可能依次触发事件 A_1、A_2，而像素 B 可能在 A_1 和 A_2 时刻之间产生一个事件 B_1。这种情况下，在 ADΔER 流中，A_1 和 A_2 出现在 B_1 之前是有效的，但我们绝不会遇到 A_2 出现在 A_1之前。

4.1.1 帧式重建与时间连贯性的保持。为了便于使用传统方法查看并评估我们的 ADΔER 转码效果，我们可以对数据进行帧式重建。为此，我们只需为 ADΔER 流中的每个像素维护一个运行时间戳计数器 T。读取给定像素的事件 x，y，D，$\Delta t'$ 时，我们将每帧的强度归一化至 $I_{frame} = \frac{2^D \Delta t_{ref}}{\Delta t'}$，然后将其设为 $[T/\Delta t_{ref}, (T + \Delta t')/\Delta t_{ref}]$ 中所有帧的像素值。

然而，我们必须确保强度变化只发生在帧时间边界上。否则，某些像素会比其他像素稍早触发，从而造成时间上的不连贯，除非源视频是以极高帧率录制的。为此，在某个像素列表状态生成新的 ADΔERevent 后，我们会忽略该状态中剩余的任何强度，从而确保每个事件在时间上的起始点对应于源视频中一帧的开始。处理事件时，我们只需将时间间隔 T 向上取整到 Δt_{ref}的下一个整数倍，即可推断事件之间的时间。我们的帧式重建程序会自动处理这种情况，因为源数据类型和 Δt_{ref} 都编码在 ADΔER 文件的文件头中。

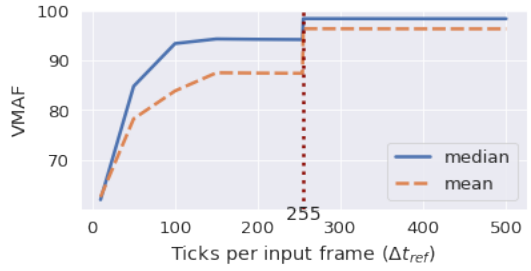


图 3：Δt_{ref} 对 ADΔER 帧式重建感知质量（VMAF）的影响。源视频为 8 位 PCC。

4.2 优化 Δt_{ref}

由于我们将 Δt_s 定义为相对于 Δt_{ref}的量，因此理解 Δt_{ref}对转码表示质量的影响至关重要。我们选取了 10 段具有多样性的帧式视频，在不同的 Δt_{ref}取值下将其转码为 ADΔER，并采用 $M = 0$（以获得最准确的表示）。随后，我们对 ADΔER 流进行帧式重建，并使用 VMAF [25]指标计算重建结果相对于源视频的感知质量，如图 3 所示。我们发现，为获得良好的重建质量，每个输入帧至少需要 255 个时钟周期；这是由源视频的位深度（即每个通道 8 位）决定的。在 $\Delta t_{ref} = 255$中，源帧的最大强度为每个时钟周期 $\frac{255}{255} = 1$ 个强度单位，我们可以用 $D = 7$，$\Delta t = 128$ 表示。最小强度为

$$\frac{0}{254}, \quad \Delta t = 255$$

可以用 $\frac{1}{255}$ 表示。然而，如果 $\Delta t_{ref} = 254$，我们会发现最大强度 $\frac{255}{254} = 1.004$ 无法用整数 Δt 值表示。因此，我们必须将 Δt_{ref}设得足够大，以便表示任意源强度而不损失准确度，即 $\frac{I_{max}}{\Delta t_{ref}} \leq 1.0$。

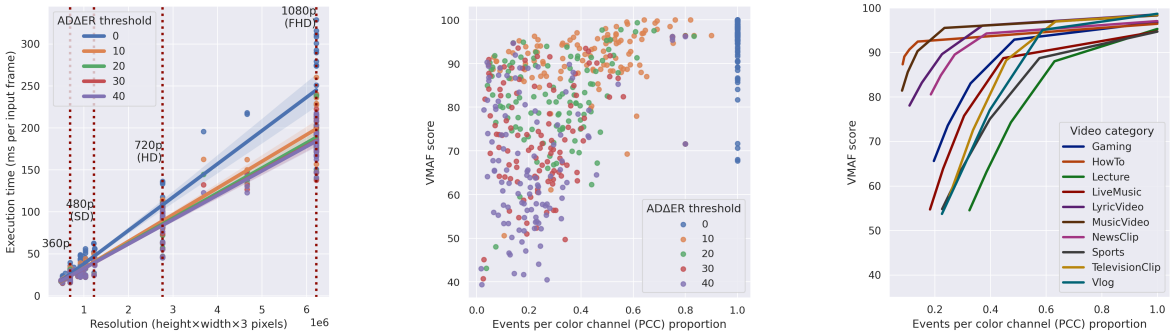
4.3 评估

为了评估我们的 ADΔER 表示在帧式视频源上的表现，我们从 YT-UGC 视频压缩数据集 [45] 的一个子集中选取了 112 个视频。这些 20 秒的视频涵盖 10 个类别（见图 5），分辨率分别为 360p、480p、720p 和 1080p。我们使用了采用 H.264 压缩、质量级别为 CRF-10 的视频版本，每个视频均为 24 位色彩。

我们使用参数 $\Delta t_{ref} = \frac{\Delta t}{120} = \frac{30600}{255}$ 和 $\Delta t_{max} = \Delta t_{ref} \cdot 255$。将每个视频转码为 ADΔER，使得任意生成的 ADΔER 事件所能跨越的最长时长为 120 个输入帧。如第 4.1 节所述，我们为每个视频设置了 $\Delta t_s = \Delta t_{ref} F$。例如，帧率为 24 帧/秒的视频具有 $\Delta t_s = 6120$。对于每个视频，我们改变了 ADΔER 的阈值参数 $M \in \{0, 10, 20, 30, 40\}$。如图 4a 所示，我们记录了每次转码在测试机器（第 3.2.4 节）上的执行时间。可以看到，M 值越大，转码操作会略快；处理每个输入帧所需的时间则随视频分辨率线性增长。

最后，我们按照第 4.1.1 节所述，从 ADΔER 表示中重建帧式视频，以便考察该表示的质量特性。我们计算了 M 范围内每个 ADΔER 视频相对于其参考视频的 VMAF 分数。图 5 显示，各视频类别的 VMAF 分数中位数极高，处于 95–99 之间；随着 M 增大，感知质量会下降。

如果我们考察不同 M 水平下，每个像素颜色通道的 ADΔER 事件数相对于 $M = 0$ 的比例（图 4b 和 4c），就会发现明显的率失真曲线，其形状取决于 M 的选择。仅通过设置 $M = 10$，我们就能看到，在整个数据集中，ADΔER 码率的中位数降低了 54%，而 VMAF 分数的中位数仅降低了 4.5。在测试中，我们观察到 ADΔER 的最高精度为 44 dB，即 14 位，显著高于源视频的 8 位精度。随着像素能够以更高的 M对更长的强度序列进行积分（从而求平均），精度往往会提高，进而扩大表示值的范围。



(a) (b) (c)

图 4：ADΔER 在帧式视频源上的定量结果。(a) 不同 M下分辨率对执行时间的影响。性能随分辨率线性变化。(b) 不同 M下所有视频的率失真散点图。(c) 数据集中每个类别各选取一个视频的特定率失真曲线。增大 M 会降低 ADΔER 视频的质量，同时降低其码率。

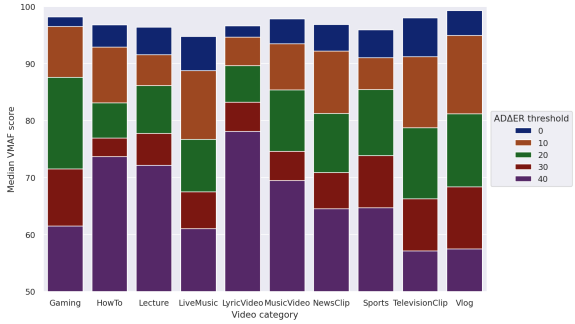


图 5：帧式视频各类别转码为不同 M下的 ADΔER 后的 VMAF 分数中位数。对于所有类别，随着 M 增大，质量都会下降。LyricVideo 等动态性较低的视频类别在 M 增大时质量几乎没有下降，而 TelevisionClip 和视频博客等动态性较高的类别则会出现大幅下降。

作为定性示例，图 8a 展示了 ADΔER 转码结果中，在我们设定的 M 值范围[1]内抽取的瞬时帧样本。随着 M提高，我们会观察到低对比度区域的“涂抹”效应加剧、整体细节减少，以及由场景转换或快速相机运动造成的多色重影伪影。出乎意料的是，图 4b 中展示的 LiveMusic 和 Sports 示例，尽管在普通观察者看来画质很高，但对于采用 $M = 0$ 的情况，它们的 VMAF 分数都是离群值。这是因为在高对比度转换点附近，某些事件的$\Delta t > \Delta t_{ref}$不够精确；由于我们将 Δt四舍五入为整数以构成一个 ADΔER 事件，因此会产生这种情况。此外，由于我们使用独立的事件像素列表表示每个颜色通道，像素某个颜色通道上的微小误差会对该像素的 VMAF感知质量产生叠加影响。这两段视频都呈现出抖动的相机运动，这表明快速的强度变化可能会使此类量化误差更频繁地出现。

[1]我们在此提供这些视频的帧式重建，作为补充材料。

5 事件视频源

在第4节中，我们展示了 ADΔER 能够以异步方式有效表示帧式视频源；这里我们将讨论 ADΔER 在表示本身已是异步的视频源方面的效用。具体而言，我们将探讨如何表示 DAVIS346 相机的动态视觉传感器（DVS）事件数据和主动像素传感器（APS）帧数据，这些数据采用 .aedat4 文件格式。

5.1 基于事件的双重积分

正如我们在第2.2节中所讨论的，许多使用混合模态源（帧和事件）的应用，要么以固定帧率重建图像序列，要么将帧数据与事件分开处理。此外，尽管 APS 帧能够测量整个传感器上的绝对强度，但在具有挑战性的曝光场景下，它们往往会出现模糊，例如场景照明较弱、快速运动或光圈较窄。与此同时，DVS 像素具有更高的灵敏度，且响应迅速，因此能够以亚帧精度捕捉事件。因此，基于 DAVIS 的重建方法必须解决 APS 帧数据去模糊这一关键问题。然而，这些方法大多采用速度较慢的机器学习技术，严重限制了实际重建帧率，从而削弱了使用事件相机的一项主要优势 [42, 43]。不过，有一种无需学习的帧式重建方法，即基于事件的双重积分（EDI） [30]。作者提出了 EDI 模型，通过对 APS 图像曝光时间内发生的 DVS 事件进行积分，为模糊的 APS 图像求得清晰的“潜在”图像 L。至关重要的是，本文采用一种优化技术来确定给定时刻 DVS 传感器的 θ 值；也就是说，该方法通过选取不同的 DVS灵敏度 θ对 APS 图像进行去模糊，而生成最清晰潜在图像时所用的 θ即为最优选择。APS 去模糊和 θ 优化技术都是我们面向事件视频源的 ADΔER 转码器处理流程中的关键步骤。

5.1.1 我们的实现。 遗憾的是，与许多关于基于事件的系统的文献一样，EDI 模型的实现并未考虑实用的实时性能。相反，作者

发布了经过混淆处理的 MATLAB 程序作为其实现，每秒只能重建几帧视频。此外，他们的实现使用了 DAVIS 数据专用的 MATLAB 格式，这印证了我们在第2.2.3节中关于特定领域的事件数据表示十分普遍的论点。因此，我们从头开始用 Rust 实现了 EDI 模型，并设计程序以解码 DAVIS 346 相机直接生成的数据包格式。对于这一分辨率，我们的实现能够以 1000+ 帧/秒的速度重建帧序列，比基准 MATLAB 实现快两个数量级。为了进行真实且可重复的性能评估，我们模拟预录制视频的数据包延迟，使其反映实际世界中的时序。用户还可以选择为每个 APS 帧生成一幅去模糊图像，而不是生成高帧率帧序列。我们的实现支持连接的 DAVIS 346 相机作为输入，为实用的实时 DAVIS 应用开辟了可能。

5.2 转码器细节

我们确定了将 DAVIS 数据转码为 ADΔER 的三种方式。每种模式都采用第 3.2 节所述的相同逐像素积分方案，但我们积分的实际强度取决于所选模式。每种情况下，我们都会从与转码器并行运行的 EDI 实现接收输入数据。与帧式源转码器不同，后者中我们根据 Δt_{ref} 的选择和视频帧率来确定 Δt_s；在这里，我们将 $\Delta t_s = 1 \times 10^6$ 设为与 DAVIS 346 的时间分辨率相匹配。然后，在下文的模式（i）和（ii）中，Δt_{ref} 用于定义所需的去模糊帧长度；在模式（iii）中，则用于定义强度刻度。图 6 展示了我们的处理流程。

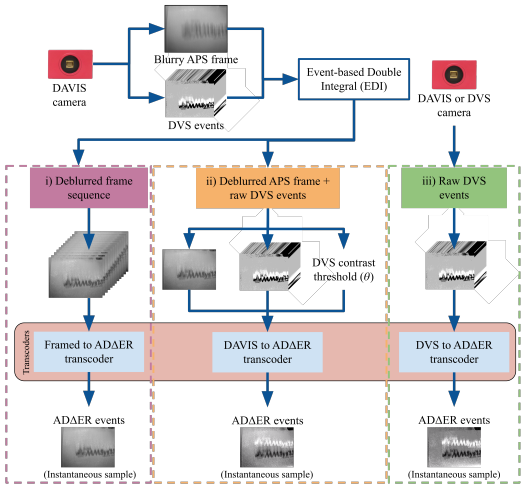


图 6：将事件相机数据转码为 ADΔER 的处理流程。我们的转码器支持三种模式。模式（i）和（ii）会纳入来自 DAVIS传感器的去模糊 APS 帧，而模式（iii）仅使用 DVS 事件。

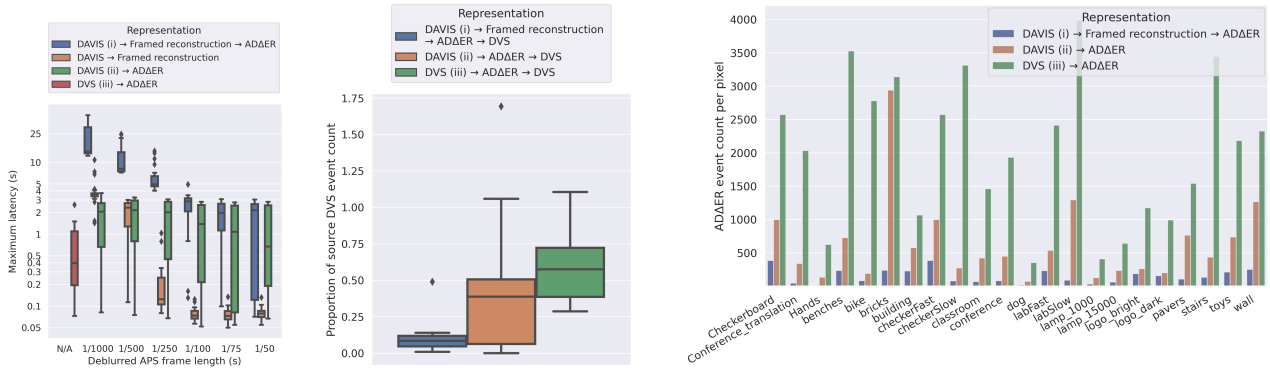
5.2.1 模式 *(i)*：去模糊帧序列 → 更Δ多。 如果我们使用 EDI 重建高帧率帧序列，就可以直接采用第 4.1 节所述的帧源转码器技术。从定性角度来看，这种方法产生的结果观感最佳，因为 EDI 模型天然能在对数空间中统一正极性和负极性事件。也就是说，我们可以累积给定帧间隔内某个像素发生的多个事件，从而得到该帧的最终强度值。只有在此时，超出 APS 帧强度范围的强度才会被钳制 $[0, 255]$。

5.2.2 模式 (ii)：去模糊 APS 帧 + DVS 事件 → ADΔER。我们也可以选择使用 EDI，仅对每个 APS 帧进行去模糊处理，并将发生在该帧曝光时间之外的 DVS 事件逐个输入我们的 ADΔER 转码器。由于 DVS 事件表示相对于先前潜在值的对数强度，我们为每个像素维护一个记录最新对数强度的矩阵，并据此计算传入 DVS 事件所表示的强度。当我们输入去模糊帧时，会将帧强度I缩放到范围 $L \in [0, 1]$，并通过 $\widetilde{L} = \ln(1 + L)$ 设置每个像素的潜在对数强度。我们将 DVS 事件解释为指定强度发生变化的确切时刻。假设对于某个像素，我们输入的去模糊帧强度为 $I_0 = 20$，其时间跨度为 $[t' - \Delta t_{ref}, t']$，，其中该帧具有 8 位值，且 $\Delta t_{ref} = 1000$。那么，我们得到潜在值 $\widetilde{L_0} = 0.0755$。假设在时间 t' 之后发生了一系列 DVS 事件 p, t，即 $\{E_1 = 0, t' + 500 \; E_2 = 1, t' + 800 \; E_3 = 1, t' + 1200\}$。当我们遇到 E_1 时，首先重复先前的积分以填补经过的时间。也就是说，我们在 500 个时钟周期内积分 $\frac{L_0}{\Delta t_{ref}} \cdot 500 = 10$ 个强度单位。

然后，我们只需按如下方式设置新的潜在强度。假设 $\theta = 0.15$，我们的潜在值 L_1 变为 $L_1 = \exp(\widetilde{L_0} - 0.15) - 1.0 = -0.0718$。然而，由于无法对负强度进行积分，我们必须将 L_1 和 $\widetilde{L_1}$ 限制为 0.0。在此阶段，我们不知道像素会维持该强度多久，因此不会立即积分额外的值。相反，当遇到 E_2时，我们会在 300 个时钟周期内积分 $\frac{L_1}{\Delta t_{ref}} \cdot 255 \; 300 = 0$ 强度单位。和之前一样，我们设置新的潜在对数值 $\widetilde{L_2} = \widetilde{L_1} + 0.15 = 0.15$，使得 $L_2 = \exp(\widetilde{L_2}) - 1.0 = 0.1618$。接着，当遇到 E_3时，我们会在 400 个时钟周期内积分 $\frac{L_2}{\Delta t_{ref}} \cdot 255 \; 400 = 41.27$ 强度单位。

尽管钳位机制的视觉效果不如模式 (i)（第 5.2.1 节），如图 8b 中运动边缘处的白色和黑色像素所示，但这种方法保留了源事件的时间分辨率。相比之下，模式 (i) 中的强度时间被量化到 EDI 重建帧率。

5.2.3 模式（*iii*）：动态视觉传感器事件 → *ADΔER.* 最后，我们可以选择忽略 APS 数据，或者只使用动态视觉传感器，并直接整合动态视觉传感器事件。在这种情况下，我们的 EDI 程序仅充当相机驱动程序或事件文件解码器。每隔亚秒级间隔，我们就将 ADΔER 转码器中的潜在对数图像 $\widetilde{L}$ 重置为 ln(0.5)，以表示中等强度。然后，我们计算后续动态视觉传感器事件相对于这一中等强度的绝对强度。由于在此模式下我们不会对 APS 帧进行去模糊处理，因此无法找到用于解读这些动态视觉传感器事件的最佳 θ值。因此，我们使用一个常数 $\theta = 0.15$ 来计算对数强度的变化。如图 6 所示，我们只能获取变化足够大、从而触发动态视觉传感器事件的像素的强度信息；未发生变化的像素则会呈现中等灰度值。



(a) (b) (c)

图 7：ADΔER 在事件相机源上的定量结果。(a) Δt_{ref}对 ADΔER 转码延迟的影响。(b) ADΔER 转码后恢复的 DVS 事件所占比例，与源 .aedat4 文件中的 DVS 事件率相比。(c) 数据集中每种转码模式和每段视频对应的 ADΔER 事件率，其中 $\Delta t_{ref} = \Delta t_s/500$ 和 $M = 40$。尽管模式（iii）未涵盖场景中缓慢变化或静止区域的绝对强度，但其产生的 ADΔER 事件率仍显著更高。

5.3 评估

我们的首要目标是推动快速、实用的基于事件的系统的开发，因此需要在真实世界的源表示上测试我们的方法。然而，正如第2.2.3节所述，大多数事件视觉数据集使用的中间表示读取速度较慢。因此，我们采用 DVSNOISE20 数据集，它提供了与 DAVIS 346 相机生成格式相同的原始 .aedat4 文件。该数据集为 16 个场景中的每个场景提供了 3 段录制数据，不过在评估中，我们每个场景只使用了一段录制数据。此外，我们还向数据集中添加了自己录制的 6 段数据，这些数据采用了不同的 APS 曝光时长和更极端的光照条件。

我们使用模式（ii），并将 ADΔER 阈值参数设为 $M \in \{0, 10, 20, 30, 40\}$、$\Delta t_{ref} \in \{100, 1000\}$、$\Delta t_s = 1\times10^6$ 和 $\Delta t_{max} = \Delta t_s \cdot 4 = 4\times10^6$，将每段 DAVIS 视频转码为 ADΔER。当 $M = 10$ 时，对于 Δt_{ref} 的两种取值，我们观察到 ADΔER 事件率的中位数相比 $M = 0$ 下降了 49–50%。在 $M = 40$ 时，我们观察到 ADΔER 事件率的中位数相比 $M = 0$ 下降了 66%，这表明我们的时间平均方案具有实用价值。

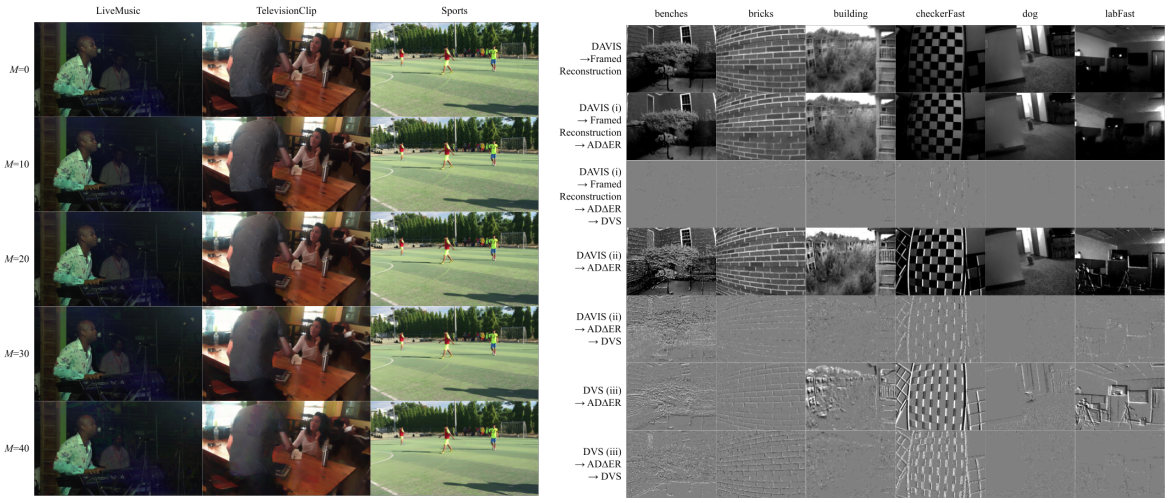
其次，我们在固定 $M = 40$ 的情况下，分别采用模式 (i) 和 (ii)，将每个 DAVIS 视频转码为 ADΔER，并使用 $\Delta t_{ref} \in \{\Delta t_s/1000, \Delta t_s\ 500\ \ \Delta t_s\ 250\ \ \Delta t_s\ 100\ \ \Delta t_s\ 75\ \ \Delta t_s\ 50\ \ /,/,/,/,/\}$ 个时钟周期。此外，我们还使用 EDI 对 DAVIS 进行了直接帧式重建，并采用模式（iii）对每个视频转码一次。我们记录了在 EDI 程序中从 .aedat4 源视频读取一个数据包到完成该数据包数据转码之间的最大延迟。在图 7a 中可以看到，随着有效帧率提高，帧式重建和转码模式 (i) 的延迟都显著增加，这印证了我们在第 4.3 节中的发现。相比之下，模式 (ii) 在 Δt_{ref} 的所有取值下延迟都相对恒定，为 2.1 秒；但随着 Δt_{ref} 增大，其方差更小。模式（iii）的速度不受 Δt_{ref} 影响，而且其中位转码延迟仅为 0.4 秒。这些结果支持了我们的观点：采用高帧率帧式强度表示处理事件数据在计算上代价高昂。虽然这类方法不可避免地会对事件源的时间分量进行量化，但我们方案的模式 (ii) 和 (iii) 可以将 DAVIS/DVS 事件数据转码为强度表示 ADΔER，其时间分辨率与源视频相匹配。

我们在图 7b 中看到了这一主张的直接证据。该图展示了使用 $M = 40$ 和 $\Delta t_{ref} = \Delta t_s/500$ 个时钟周期转码后，从 ADΔER 表示中重建出的 DVS 事件比例，并将其与源 DAVIS 视频中的 DVS 事件数量进行了比较。我们可以快速从 ADΔER 重建 DVS 流，因为触发 DVS 事件的强度变化只会发生在稀疏 ADΔER 事件的时间边界处。不过，我们必须假设强度恒定，如转码模式 (iii)（第 5.2.3 节）所示。模式 (i) 将事件量化为固定时长帧，只能恢复源视频中一小部分 DVS 事件。相反，即使在较高的 M 下，模式 (ii) 和 (iii) 也能恢复很大比例的 DVS 事件。图 7c 显示，在相同设置下，转码模式 (ii) 通常比模式 (iii) 所需的 ADΔER 事件少得多。因此，我们采用模式 (ii) 的 ADΔER 编解码器可以同时利用 APS 帧压缩 DVS 流，并利用 DVS 流对 APS 帧去模糊，同时将这两种流融合为统一的异步强度表示。

图 8b 展示了我们数据集[2]中六个场景的定性结果。图中所示的 AD Δ $M = 40$ $\Delta t_{ref} = \Delta t_s$ 500。ER 转码采用了与延迟实验中相同的 和 /。由于事件转码器的模式 (i) 只是在使用 EDI 重建帧序列后，调用我们的帧式源转码器（第 4.1 节），因此我们指出，这里的 ADΔER 表示中的伪影源于 EDI 的帧式重建伪影，或 ADΔER 中的时间平均。特别是，我们指出这些建筑图像：它们展现了 EDI 在快速运动和长曝光条件下对 APS 帧进行去模糊的局限性。我们发现，从模式 (i) 中恢复出的 DVS 事件很少，这是由于源事件在时间上被量化所致。

[2]我们在此提供这些视频的帧式重建，作为补充材料。

,
,



(a) (b)

图 8：ADΔER 转码器的定性结果。(a) 将帧式视频转码为 ADΔER 后，在不同 M 下的瞬时采样。较高的 M 会导致低对比度区域出现更多拖影和重影伪影。(b) 将事件视频转码为帧式视频（第一行），以及在我们的三种转码模式下转码为 ADΔER（第 2、4、6 行），$\Delta t_{ref} = \Delta t_s/500$ 和 $M = 40$。我们将 ADΔER 表示转回 DVS（第 3、5、7 行），可以看到模式 (ii) 和 (iii) 保留了更多 DVS 流的时间对比度细节。

然而，尽管模式 (ii) 还传达了没有 DVS 源事件的像素的绝对强度，但其恢复的 DVS 事件质量与模式（iii）相近。

6 未来工作

6.1 扩展本文

我们注意到，针对 DAVIS/DVS 源，对转码表示的质量进行实证评估存在困难。由于 APS 帧存在模糊，因此无法作为帧式质量指标（例如峰值信噪比、结构相似性和 VMAF）的可靠参照。此外，现有的 DVS 流质量指标侧重于噪声滤波任务，每秒只能处理数百个事件 [24],，而真实世界的 DVS 序列每秒包含数百万个事件。因此，待端到端系统搭建完成后，我们将以应用性能作为衡量从基于事件的源转码而来的流的表示质量的指标。最后，我们还将扩展框架，以支持更多事件相机数据类型，例如 Prophesee [31] 的数据。

6.2 更宏观的前景

我们的视频采集与表示处理流程，是迈向构建实时、与源无关的事件视频生态系统的必要第一步。接下来的工作是对 ADΔER 数据进行压缩、流传输并开发相关应用。

如前所述，ADΔER 适用于有损压缩，但与传统视频编解码器不兼容。因此，需要采用新技术，根据数据失真、连接带宽和应用性能等指标来引入损失。ADΔER 的稀疏表示自然会产生较低的数据速率，因为变化最小的像素所需的数据速率较低；但压缩技术必须辨别如何引入强度精度损失，同时避免独立像素流之间出现时间去相干（即事件的 Δt 值损失不同程度的精度）。

最后，互联客户端应用可以采用双管齐下的方式处理 ADΔER 数据：一类是专门的基于事件的应用，直接接收 ADΔER 流；另一类是传统的基于帧的应用，希望利用我们稀疏的表示和高时间精度。对于后一类，客户端可以使用我们的开源库，根据接收到的 ADΔER 数据重建帧图像；不过，在实时流传输的情况下，重建所产生的最大延迟将随 Δt_{max} 的选择呈线性增长。计算开销也会限制可实时处理的时空分辨率，不过，高效的压缩方案有助于缓解这一障碍。

7 结论

本文提出了适用于帧源和事件源的 ADΔER 视频表示。我们表明，通过对相似强度进行时间平均，与基于网格的表示相比，可以提高强度精度，并大幅减少每像素采样数。我们展示了延迟极低的 DAVIS 融合，并指出，我们的方法具备构建面向事件相机的实时强度应用所需的计算效率和时间粒度。

参考文献

[1] Raymond Baldwin、Ruixu Liu、Mohammed Mutlaq Almatrafi、VijayanK Asari 和 Keigo Hirakawa。2022。面向事件相机的时间排序近期事件（TORE）体积。*IEEE Transactions on Pattern Analysis and Machine Intelligence*（2022），1–1。https://doi.org/10.1109/TPAMI.2022.3172212

[2] Srutarshi Banerjee、Henry H. Chopp、Jianping Zhang、Zihao W. Wang、Oliver Cossairt 和 Aggelos Katsaggelos。2021。面向资源受限设备的联合强度-神经形态事件成像系统。arXiv:
2105 14164 [ ] 。eess.IV
[3] Srutarshi Banerjee、Zihao W. Wang、Henry H. Chopp、Oliver Cossairt 和 Aggelos K. Katsaggelos。2021。基于图像派生四叉树和泊松盘采样的有损事件压缩。载于2021年IEEE国际图像处理会议（ICIP）。IEEE。https://doi.org/
10 1109 42928 2021 9506546 。/icip。。2021
[4] Thomas Barbier、Celine Teuliere 和 Jochen Triesch。。脉冲神经网络中方向、视差和运动表示的基于脉冲时序的无监督学习。载于IEEE/CVF计算机视觉与模式识别会议研讨会论文集。1377–1386。[5] Ryad Benosman、Charles Clercq、Xavier Lagorce、Sio-Hoi Ieng 和 Chiara Bartolozzi。2014。基于事件的视觉流。IEEE Transactions on Neural Networks and Learning Systems 25，2（2014），407–417。https://doi.org/10.1109/TNNLS.2013.2273537[6] Raphael Berner、Christian Brändli 和 Massimo Zannoni。2018。基于事件的视觉传感器的数据速率控制。[7] G. Bradski。2000。OpenCV库。Dr. Dobb's Journal of Software Tools（2000）。[8] Christian Brandli、Lorenz Muller 和 Tobi Delbruck。2014。使用基于帧与事件的DAVIS传感器实时高速视频解压缩。载于2014年IEEE国际电路与系统研讨会（ISCAS）。686–689。https://doi.org/
10 1109 2014 6865228 。/ISCAS。。Benjamin Bross、Ye-Kui Wang、Yan Ye、Shan Liu、Jianle Chen、Gary J. Sullivan 和 Jens-Rainer Ohm。2021。通用视频编码（VVC）标准及其应用概述。IEEE Transactions on Circuits and Systems for Video Technology
31, 10 2021 3736 3764 10 1109 2021 3101953 （），–。
[9] https://doi.org/。/TCSVT。。T. Delbruck、R. Graca 和 M. Paluch。。事件相机的反馈控制。载于2021年IEEE/CVF计算机视觉与模式识别会议研讨会（CVPRW）。IEEE计算机学会，美国加利福尼亚州洛斯阿拉米托斯，1324–1332。https://doi.org/10.1109/CVPRW53098.2021.00146[11] Georgi Dikov、Mohsen Firouzi、Florian Röhrbein、Jörg Conradt 和 Christoph Richter。2017。借助神经形态硬件实现延迟为2毫秒的脉冲协同立体匹配。载于《仿生与生物混合系统》，Michael Mangan、Mark Cutkosky、Anna Mura、Paul F.M.J. Verschure、Tony Prescott 和 Nathan Lepora（编）。Springer International Publishing，Cham，119–137。[12] Hadar Cohen Duwek、Albert Shalumov 和 Elishai Ezra Tsur。2021。利用拉普拉斯预测和泊松积分以及脉冲神经网络与人工神经网络，从神经形态事件相机重建图像。载于IEEE/CVF计算机视觉与模式识别会议研讨会论文集。1333–1341。[13] FFmpeg项目。2021。FFmpeg。https://ffmpeg.org/ [14] Andrew C. Freeman、Chris Burgess 和 Ketan Mayer-Patel。2021。面向事件相机集成的运动分割与跟踪。载于第12届ACM多媒体系统会议论文集（土耳其伊斯坦布尔）（MMSys '21）。计算机协会，美国纽约州纽约，1–11。https://doi.org/
10 1145 3458305 3463373 。/。Andrew C. Freeman 和 Ketan Mayer-Patel。2020。集成事件相机传感器模拟器。载于第28届ACM国际多媒体会议论文集（美国华盛顿州西雅图）（MM '20）。计算机协会，美国纽约州纽约，
4503 4505 10 1145 3394171 3414394 –。
[15] https://doi.org/。/。Andrew C. Freeman 和 Ketan Mayer-Patel。。用于集成事件相机的有损压缩。载于2021年数据压缩会议（DCC）。53–62。https://doi.org/
10 1109 50243 2021 00013 。/DCC。。G. Gallego、T. Delbruck、G. M. Orchard、C. Bartolozzi、B. Taba、A. Censi、S. Leutenegger、A. Davison、J. Conradt、K. Daniilidis 和 D. Scaramuzza。2020。基于事件的视觉：综述。IEEE Transactions on Pattern Analysis and Machine Intelligence（
2020 1 1 10 1109 2020 3008413 ），–。
[16] https://doi.org/。/TPAMI。。Daniel Gehrig、Antonio Loquercio、Konstantinos G. Derpanis 和 Davide Scaramuzza。2019。异步事件数据表示的端到端学习。载于IEEE/CVF 国际计算机视觉大会（ICCV）论文集。[19] Arren Glover、Valentina Vasco 和 Chiara Bartolozzi。2018。面向在线机器人技术的受控延迟事件相机框架。载于2018年IEEE国际机器人与自动化会议（ICRA）。2178–2183。https://doi.org/
10 1109 2018 8460541 。/ICRA。。Yuhuang Hu、Jonathan Binas、Daniel Neil、Shih-Chii Liu 和 Tobi Delbruck。2020。DDD20端到端事件相机驾驶数据集：通过深度学习融合帧与事件，提升转向预测效果。载于2020年IEEE第23届智能交通系统国际会议（ITSC）。1–6。https://doi.org/ 10 1109 45102 2020 9294515 。/ITSC。。
[20] Yuhuang Hu、Shih-Chii Liu 和 Tobi Delbruck。2021。v2e：从视频帧生成逼真的DVS事件。载于2021年IEEE/CVF计算机视觉与模式识别会议研讨会（CVPRW）。1312–1321。https://doi.org/10.1109/CVPRW53098.2021.00144[22] Alex Junho Lee、Younggun Cho、Young-sik Shin、Ayoung Kim 和 Hyun Myung。2022。ViViD++：可见性视觉数据集。IEEE机器人与自动化快报 7，3（2022），6282–6289。
[21] https://doi.org/。/。Lucien Lenzen、Rolf Hedtke 和 Mike Christmann。2018。《色调映射如何影响编码序列的比特率和位深度》。SMPTE Motion Imaging Journal
127, 5 2018 38 43 10 5594 2018 2810021 ( ), – 。
[23] https://doi.org/ 。/JMI. 。Jianing Li、Yihua Fu、Siwei Dong、Zhaofei Yu、Tiejun Huang 和 Yonghong Tian。2021。《面向事件相机的异步时空脉冲度量》。IEEE Transactions on Neural Networks and Learning Systems (2021)，1–12。https: // doi.org/10.1109/TNNLS.2021.3061122[25] Zhi Li、Anne Aaron、Ioannis Katsavounidis、Anush Moorthy 和 Megha Manohara。2016。《迈向实用的感知视频质量度量》。The Netflix Tech Blog 6, 2 (2016)。[26] Ana Maqueda、Antonio Loquercio、Guillermo Gallego、Narciso Garcia 和 Davide Scaramuzza。2018。《基于事件的视觉与深度学习相结合，用于自动驾驶汽车的转向预测》。5419–5427。https://doi.org/10.1109/CVPR.2018.00568[27] Elias Mueggler、Henri Rebecq、Guillermo Gallego、Tobi Delbruck 和 Davide Scaramuzza。2017。《事件相机数据集与模拟器：用于位姿估计、视觉里程计和同步定位与地图构建的基于事件的数据》。The International Journal of Robotics Research 36, 2 (2017)，142–149。https://doi.org/10.1177/0278364917691115 arXiv:https://doi.org/
10 1177 0278364917691115 。/ Yuya Omori、Takayuki Onishi、Hiroe Iwasaki 和 Atsushi Shimizu。2017。《采用可扩展至4K并行配置的120帧/秒高帧率实时HEVC视频编码器》。载于2017 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS)。1–3。https://doi.org/10.1109/CoolChips.2017.7946382[29] Liyuan Pan、Richard Hartley、Cedric Scheerlinck、Miaomiao Liu、Xin Yu 和 Yuchao Dai。2020。《基于事件相机的高帧率视频重建》。IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)，1–1。https://doi.org/10.1109/TPAMI.2020.3036667 [30] Liyuan Pan、Cedric Scheerlinck、Xin Yu、Richard Hartley、Miaomiao Liu 和 Yuchao Dai。2019。《借助事件相机让模糊帧焕发生机，实现高帧率视频》。载于IEEE/CVF计算机视觉与模式识别会议论文集 (CVPR)。[31] Etienne Perot、Pierre de Tournemire、Davide Nitti、Jonathan Masci 和 Amos Sironi。2020。《使用百万像素事件相机学习目标检测》。载于第34届神经信息处理系统国际会议论文集（加拿大温哥华）(NIPS'20)。Curran Associates Inc.，美国纽约州 Red Hook，文章1396，共14页。[32] Christoph Posch、Daniel Matolin 和 Rainer Wohlgenannt。2010。《具有无损像素级视频压缩功能的QVGA 143dB动态范围异步地址事件PWM动态成像传感器》。载于2010 IEEE International Solid-State Circuits Conference - (ISSCC)。400–401。https://doi.org/10.1109/ISSCC.2010. 5433973[33] Henri Rebecq、Timo Horstschaefer 和 Davide Scaramuzza。2017。《使用基于关键帧的非线性优化实现事件相机的实时视觉惯性里程计》。https://doi.org/
10 5244 31 16 。/C. 。Henri Rebecq、René Ranftl、Vladlen Koltun 和 Davide Scaramuzza。2019。《从事件到视频：将现代计算机视觉带给事件相机》。IEEE Conf. Comput. Vis. Pattern Recog. (CVPR) (2019)。[35] Henri Rebecq、René Ranftl、Vladlen Koltun 和 Davide Scaramuzza。2019。《利用事件相机实现高速、高动态范围视频》。IEEE Trans. Pattern Anal. Mach. Intell. (T-PAMI) (2019)。http://rpg.ifi.uzh.ch/docs/TPAMI19_Rebecq. pdf [36] Christian Reinbacher、Gottfried Munda 和 Thomas Pock。2016。《使用流形正则化实现事件相机的实时强度图像重建》。 9 1 9 12 10 5244 30 9 。– 。
[34] Sourav Dey Roy 和 Mrinal Kanti Bhowmik。2020。《基于计算机视觉的运动目标检测方法综述》。载于2020 IEEE Region 10 Symposium (TENSYMP)。1531–1534。https://doi.org/10.1109/ TENSYMP50017.2020.9230869[38] Cedric Scheerlinck、Nick Barnes 和 Robert Mahony。2019。《使用事件相机进行连续时间强度估计》。载于《计算机视觉—ACCV 2018》，C.V. Jawahar、Hongdong Li、Greg Mori 和 Konrad Schindler（编）。Springer International Publishing，Cham，308–324。[39] Cedric Scheerlinck、Henri Rebecq、Daniel Gehrig、Nick Barnes、Robert Mahony 和 Davide Scaramuzza。2020。《使用事件相机快速重建图像》。载于IEEE Winter Conf. Appl. Comput. Vis. (WACV)。156–163。https://doi.org/10.1109/WACV45572.2020.9093366[40] M. Singh、P. Zhang、A. Vitkus、K. Mayer-Patel 和 L. Vicci。2017。《采用异步像素的无帧成像传感器：架构评估》。载于2017 23rd IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)。 110 117 10 1109 2017 19 – 。
[37] https://doi.org/ 。/ASYNC. 。Aaron J. Smith、Montek Singh 和 Ketan Mayer-Patel。2017。《无帧异步高动态范围传感器的系统模型》。载于NOSSDAV'17。[42] Chen Song、Qixing Huang 和 Chandrajit Bajaj。2022。《E-CIR：事件增强连续强度恢复》。载于2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)。7793–7802。https://doi.org/10.1109/CVPR52688. 2022.00765 [43] Bishan Wang、Jingwei He、Lei Yu、Gui-Song Xia 和 Wen Yang。2020。《事件增强高质量图像恢复》。载于欧洲计算机视觉会议。Springer。

[44] Hao Wang、Youhui Huo 和 Hongbo Zhang。2021。《成像设备动态范围分析与改进研究》。载于*2021*年算法与大数据研讨会（中国福州）*(WABD 2021)*。计算机协会，美国纽约州纽约，40 44 10 1145 3456389 3456392 – 。[45] https://doi.org/ 。/ 。Yilin Wang、Sasi Inguva 和 Balu Adsumilli。 。《用于视频压缩研究的YouTube 用户生成内容数据集》。载于*2019 IEEE*第*21*届多媒体信号处理国际研讨会*(MMSP)*。[46] Ziwei Wang、Yonhon Ng、Cedric Scheerlinck 和 Robert Mahony。2019。2021。《混合事件相机的异步卡尔曼滤波器》。载于*IEEE/CVF* 国际计算机视觉大会论文集*(ICCV)*。448–457。[47] Alex Zhu、Liangzhe Yuan、Kenneth Chaney 和 Kostas Daniilidis。2018。《EV-FlowNet：面向事件相机的自监督光流估计》。https://doi.org/10.15607/RSS.2018.XIV.062[48] Bin Zhu、Shan Liu、Yuan Liu、Yi Luo、Jing Ye、Haiyan Xu、Ying Huang、HualongJiao、Xiaozhong Xu、Xianguo Zhang 和 Chenchen Gu。2021。《实时H.266/VVC软件解码器》。载于*2021 IEEE*多媒体与博览会国际会议 *(ICME)*。1–6。https://doi.org/10.1109/ICME51207.2021.9428470