通过非线性采样实现强大的360度视频流

米贾努尔·帕拉什、沃伊库·波佩斯库、阿米特·谢奥兰、索尼娅·法赫米

普渡大学

fahmy}@purdue.edu，阿舍兰，波佩斯库

***摘要—*我们提出了CoRE，一种360视频流方法，与传输整个360度视频相比，它降低了带宽要求。CoRE在空间和时间域中使用非线性采样来实现对视图的鲁棒性**

**方向预测误差和瞬时无线网络带宽波动。每个CoRE帧从各个方向对环境进行采样，在预测的视野范围内具有全分辨率，在外围分辨率逐渐降低，因此无论视图预测误差大小如何，都会避免丢失像素。CoRE视频块chunk有一个全帧速率的主要部分和一个逐渐降低的帧速率的扩展部分，这可以避免在等待延迟传输时停顿。我们通过基于轨迹的实验和用户研究评估了CoRE的原型实现，发现与使用低分辨率填充的切片相比，CoRE减少了数据传输量、停顿和H.264解码开销，提高了帧速率，并消除了丢失的像素。**

1. 介绍

360视频让用户生动体验复杂环境。用户可以主动改变观看方向，这与被动观看为用户选择观看方向的传统视频相比，在环境中提供了更大的存在感。

**挑战。**一个360的视频要存储大量的像素才能捕捉到四面八方的环境。为了降低无线链路的带宽要求，仅传输用户实际看到的360视频部分。例如，对于90°\*90°的用户视野(FoV)，用户只看到大约八分之一的360°\*180°等矩形帧。在这种情况下，360视频流必须预测用户的观看方向[[17].](#_bookmark51)

×

×

无线360视频流的第二个挑战是网络带宽的波动，这是由无线电塔之间的切换或稀疏无线电覆盖引起的[[1],](#_bookmark35) [[23].](#_bookmark57) 例如，高速列车上的移动设备每150秒就会出现2到15秒的连接故障[[22].](#_bookmark56) 随着蜂窝运营商努力争取更高的带宽，5G无线电最初稀疏的覆盖范围[[19]](#_bookmark53) 会加剧带宽波动，使带宽预测变得困难。在传统的视频流中，带宽波动可以通过预取和速率自适应来掩盖[[2],](#_bookmark36)[[27].](#_bookmark61)然而，360视频流中的预取受限于人们能够可靠地预测用户观看方向的前方有多远。延迟预取会导致播放暂停和观众流失[[10].](#_bookmark44)

这项工作得到了国家自然科学基金资助CNS-1717493的部分支持。

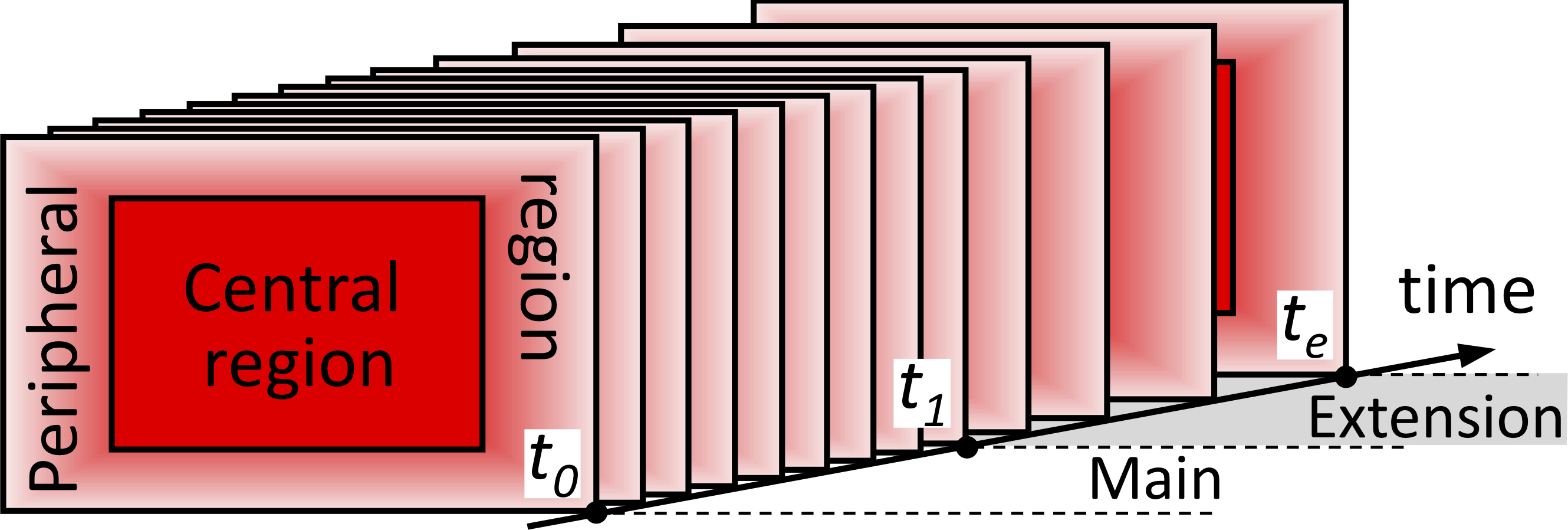


图1:CoRe 360视频概念。

**最先进的State of the Art。**大多数360视频流系统[[8],](#_bookmark42) [[17]](#_bookmark51)将输入视频均匀地分割成矩形块，并传输与预测的用户FoV重叠的块，加上安全边缘，可能以较低的分辨率。例如，在等矩形视频的4\*6平铺中，一个图块覆盖大约60°\*45°,因此90°\*90°FoV至少需要2\*2个图块，这对应于最多24/4 = 6的传输大小缩减。只有10°的安全边缘使FoV达到110°110°,传输大小减少24/(3\*3)= 2.66。在FoV外围使用分辨率较低的图块会减小传输尺寸，但会引入突然的分辨率变化。

为了提高视图预测的准确性，360视频流系统将视频块的持续时间从4秒[9]减少到1秒[[8],](#_bookmark42) [[17].](#_bookmark51) 这意味着延迟预取和更可能的停顿。此外，更短的块产生更多的文件，例如，331个瓦片中的4个的36个文件。这降低了压缩效率，因为编解码器不能跨越空间和时间块边界利用数据冗余，并且增加了客户端的解压缩成本。这可以通过可变大小的分离来缓解[[6]](#_bookmark40) 或者重叠[[29]](#_bookmark63) 图块，允许以统一的复杂性进行分区[[6]（Pano的做法）,](#_bookmark40)并允许管理服务器的存储需求[[29],](#_bookmark63)同时限制瓦片的总数，尽管约为30个瓦片。最先进的系统[[6],](#_bookmark40)[[8],](#_bookmark42)[[17]](#_bookmark51)需要一个GPU，这会耗尽电池，并排除弱客户端。

×

**提出的方法。**我们提出CoRE(Compressed Rotated Equirectangular压缩旋转等矩形投影)360视频流，旨在满足五个主要要求：

1. *对视图预测误差的鲁棒性：*无论视图预测误差大小如何，都不会丢失像素。
2. *对瞬时带宽波动的鲁棒性：*当下一个块被延迟时，最小化停顿。
3. *单个文件：*从一个视频文件重建每个帧。
4. *与当前流的兼容性：*利用标准编解码器和协议，例如H.264和DASH[[9].](#_bookmark43)
5. *计算效率：*支持没有GPU的客户端，计算成本低。

CoRE基于为流式传输设计的360°视频参数化。***在空间上，***一个CoRE帧覆盖所有视图方向，因此不存在丢失的像素，无论视图预测误差大小如何。一个CoRE帧有一个全分辨率的中心区域和一个分辨率逐渐降低的外围区域[(图](#_bookmark0)1).中心区域与用户视图方向对齐，因此预测FoV内的输出帧像素是根据全分辨率数据计算的。预测FoV之外的像素是从较低分辨率的外围区域计算的。外围区域的非线性分辨率为更小的预测误差提供了更高的质量，而预测误差更频繁。CoRE帧分辨率的连续性避免了突然的质量变化（分辨率是连续的，而不是离散的1080p~360p）。

***在时间上，***一个CoRE块chunk在全帧速率下有一个主要部分(图1中，从t0到t1)，随后是帧速率逐渐降低的扩展部分(从t1到te)。在没有带宽波动的情况下，以全帧速率使用主要部分重构输出视频。如果下一个块被延迟，通过从扩展部分重建输出视频，可以避免延迟达te-t1秒。扩展的非线性帧速率处理更可能的短延迟，而不会显著降低帧速率。当使用扩展时，视频捕获的场景更新不太频繁，但是系统保持对用户视角方向变化的完全响应，而没有传统比特率降低所暗示的视觉降级。用户可以无延迟地旋转视图方向，唯一的区别是动态场景对象每秒更新10次，而不是30次。

−

CoRE非线性空间和时间采样为防止视图预测误差和停滞导致的退化提供了保障。保险费是附加数据在外围区域和延伸部分的转移。基于应用、场景和网络条件，采样方案提供了最小化额外传输和最大化输出质量所需的灵活性。

使用标准低级视频编解码器(例如H.264)压缩CoRE块，并使用标准视频流协议(例如DASH)将其传输到客户端。服务器为所有可能的视图方向的统一离散化预先计算块，这在存储与客户端数量的可伸缩性之间进行了权衡，符合DASH范式。单个CoRE帧覆盖整个FoV，最大限度地提高了视频压缩效率，并最大限度地降低了客户端视频解码成本。最后，CoRE优化了从输出帧到CoRE帧的映射，在纯串行CPU实现中，在平板电脑上实现了30fps，而大多数现有系统只能与GPU一起工作。这为最薄客户端上的360°视频流打开了大门。

我们评估了几个360°视频、用户视角方向轨迹和无线网络轨迹上的CoRE，将其与五种基线切片变体进行了比较，并进行了用户研究。我们的结果表明，与使用填充的切片相比，CoRE显著减少了传输的数据量、停顿和H.264解码开销，消除了丢失的像素和突然的分辨率变化，并提高了帧速率。

1. 核心流

[Figure 2](#_bookmark1)显示了CoRE 360视频流系统的架构。360视频内容离线编码(1-2)成CoRE块[(§II-A)](#_bookmark2)用于所有可能的视图方向的离散化。然后，在线流式传输时，在自适应确定的时间[(§II-E),](#_bookmark17) 客户端基于预测视图(5)从服务器(6-9)请求CoRE块，预测视图(5)基于视图用户界面(3)捕获的当前用户视图(4)。客户端解码核心块(A，[§II-C)](#_bookmark15)使用当前用户视图(B)计算当前输出帧(C)。



*计算机网络服务器*

*客户*

3

查看用户界面

一

CoRE编码

四

B

C

2

查看预测核心解码

5 A

流式控制

8

七

6

9

CoRE组块

360o视频

*网络*

图2: CoRE架构。

1. *CoRE编码*

CoRE块是一个视频立方体，它以全分辨率覆盖中心区域，以递减的空间和时间分辨率覆盖外围区域[(Figure](#_bookmark0) 1).CoRE块是用[algorithm 1.](#_bookmark3)输入视频V∫采用标准视频编解码器编码，如H.264，支持任意全向参数化，如等矩形、立方体贴图、鱼眼或其他球形投影；所需要的只是一个将方向投射到输入360视频帧上的功能，如下所述。选择w h是为了匹配输出客户端帧分辨率，但不超过输入视频中可用的分辨率V∞。we、te和ne平衡了健壮性/开销的权衡。

×

算法1:核心块编码

输入:输入360°视频V∫，预测用户观看方向dp，用户FoV∏φ∏θ，CoRE组块开始时间t0，主时长∏t，中心区域分辨率w h，周边

×

×

区域横向厚度we、延伸持续时间te和延伸帧数ne。

**输出:CoRE视频块CoRE。**

1 (V，ω) =低电平解码(V∫，t0，t0+∏t+∏te)

2表示i = 0至ω∏t do

3*VCoRE [i] = CoREFrameEncd(V [i]，dp，∏φ，∏θ，w，h，we)*

4 (a0，a1，a2)= setup frame rate reduce(ω，te，ne)

**5代表j = 1到ne do**

6 TJ = a0j 2+a1j+a2；k =ω(⇼t+TJ)

[ ♩

七*VCoRE [i++] =*

*CoREFrameEncd(V [k]，dp，∑φ，∑θ，w，h，we)*

8返回VC∫ORe = low level encode(VC ORe)

该算法(从输入视频中)提取主时间间隔和扩展时间间隔(第1行)内的帧。这

算法计算主要部分的ω∏t帧，其中ω是输入视频的帧速率(第2-3行)。每个CoRE帧都是根据一个输入视频帧计算的，如中所述[§II-B.](#_bookmark9) 主要部分具有恒定的帧速率ω。

扩展部分帧在时间轴上的位置越来越远[(Figure](#_bookmark0) 1).让扩展的帧j的时间线位置是t(j)，相对于扩展的开始。那么t(j)必须满足三个条件[(Equation](#_bookmark4) 1):第一帧必须放在扩展的开头，最后一帧必须放在扩展的结尾，扩展部分开头的帧率必须与主体部分的帧率匹配，即ω。

*t(0) = 0，t(ne)= te，t′(0)= 1/ω(1)*

满足三个条件需要三个系数，所以t(j)的最简单表达式是二次的[(Equation](#_bookmark5) 2).

*t(j) = a0j2 + a1j + a2，t′(j)= 2a0j+a1* (2)

算法2:核心帧编码

输入:360♀视频帧F，预测用户视角方向dp，用户FoV∏φ∏θ，中心区域分辨率w h，

× ×

和外围区域横向厚度we。

**输出:CoRE框架FCoRE**

1他=我们(180o∏θ)/(360 o∆φ)

— −

2 u0 = w(360 o/∏φ1)/2；v 0 = h(180 o/∏θ1)/2

— −

3 R.x = dpR.y = dp (0，1，0)；R.z = R.x R.y

× ×

**4表示v = 0到h + 2he do**

**5表示u = 0至w + 2we do**

如果(u，v)[我们，我们+w][他，他+ h]那么

∈ ×

7 (ue，ve) = (u，v) (we，he) + (u0，v0)

−

**其他8个**

9 (ue，ve) =展开(u，v)

10 de =取消项目(ue，ve)

**11 dr = RT dT**

*e*

12 (ui，vi) =项目(dr)

13 FCoRE (u，v)= LookOut(F，ui，vi)

**14返回FCoRE**

系数ai

*q r*

被发现[(Equation 3,](#_bookmark8) 第4行)通过求解

*行政帐户(account executive)*

*体育课(physical education)*

*a*

*q*

*c*

*v*

*0*

*h*

*e*

*2*

*r*

*2*

*q*

*r*

*q0 pc b*

*中部*

*r0*

*一*

*一*

由方程定义的线性方程组[1](#_bookmark4) 和[2.](#_bookmark5)CoRE组块的扩展部分上的最低帧速率出现在扩展的末尾，并且它等于1/t′(ne)。扩展上的平均帧速率为ne/te。

*A0 =(ω∏te ne)/(ωN2)，a1 = 1/ω，a2 = 0 (3)*

*e*

该算法从其相应的输入视频帧k(第5-7行)计算每个扩展帧j。通过计算帧j的扩展时间线位置tj，然后通过在时间点∏t+TJ(第6行)找到输入帧，可以找到k。最后，使用编解码器(例如H.264)对CoRE块进行编码，以通过数据压缩来减小文件大小(第8行)。

1. *核心帧编码*

CoRE帧的计算如下[(Figure 3](#_bookmark10) a-d)。输入的等矩形框架(a)被旋转(b)以使预测的用户FoV居中(黑色)。然后，外围区域被压缩。最终的CoRE框架以c比例显示，以d比例放大，中心区域显示为红色。

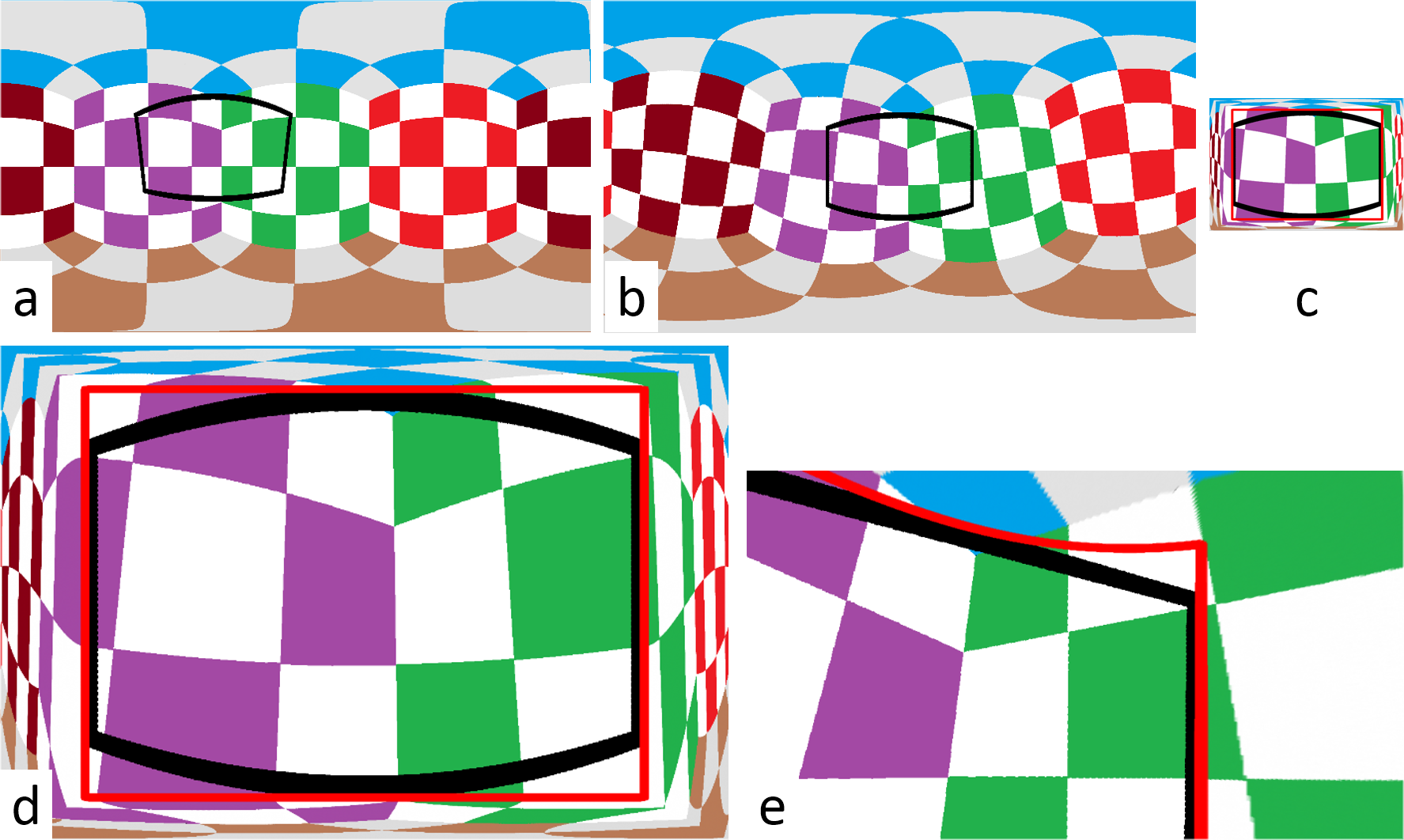


图3: CoRE帧构建阶段(a-d)和输出帧

(e)从CoRE帧(d)解码。

一个CoRE帧是根据输入360帧F计算的[algorithm 2.](#_bookmark6)使用由输入参数we(第1行)定义的每度像素数来计算外围区域he的垂直厚度。为

图CoRE外围区域的扩展。

例如，对于90°○90°用户FoV，外围区域的垂直厚度为we/3。扩展的等矩形框架中中心区域左上角的坐标(u0，v0)计算为扩展框架总宽度和高度的一半(第2行)。在我们的90°○90° FoV示例中，对于1，000°的w，扩展框架的总宽度为1，000(360○/90○) = 4，000，u0为1，500。用户FoV的局部坐标系是根据预测的视图方向(第3行)构建的。

×

×

CoRE帧一次计算一个像素(第4-13行)。如果当前像素(u，v)在核心区域内，它在扩展的等矩形框中的坐标是通过添加一个考虑到不同外围区域厚度的偏移来计算的(第7行)。如果当前像素在外围区域(第9行)，则如下所述计算其扩展坐标。

到当前像素(u，v)中心的方向de是通过将等矩形框2D点(ue，ve)不投影到单位球面上的3D点de(线10)来计算的。de旋转回到世界坐标系以获得dr(第11行)。最后，dr被投影到输入360帧F(第12行)，在此查找颜色(第13行)。例如，如果F是等矩形，投影计算单位球面上给定点的纬度/经度值。

[Figure 4](#_bookmark7) 说明了压缩外围区域(阴影部分)顶部内的点pc扩展到扩展的等矩形框架点pe。扩展过程如下。在线段q1q0和r1r0与穿过pc的水平线的交点处找到点p和q。找到点ac，使得q2c/AC R2 = qpc/PCR。通过将acpc分别与q1r1和q0r0相交，找到点ae和b。这

pc的扩展位置pe属于aeb。通过计算pe的垂直坐标ve将pe放置在aeb上。ve是函数ve(vc)，它必须满足三个条件[(Equation](#_bookmark11) 4).

*ve(he) = v0，ve(0) = 0，ve′(he)= 1* (4)

前两个条件定义了外围区域的内边缘和外边缘的扩展(例如，b的坐标从he变为v0，而ac的坐标保持不变

当扩展到ae时为0)。第三个条件确保CoRE采样速率从中心区域到外围区域是连续的。二次型满足所有三个条件[(Equation](#_bookmark12) 5).

*v (v ) = a v2 + a v + a，v′(v)= 2a v+a(5)*

算法3:核心帧解码

输入:CoRE视频帧FCoRE，中心区域分辨率

*w h，外围区域横向和垂直厚度we和he，从世界到CoRE局部坐标系的旋转矩阵R，当前用户视图摄像机矩阵Cf，以及输出帧分辨率wf 高频。*

×

×

输出:客户端为用户显示的帧Fout。

**1表示v = 0至hf do**

**2表示u = 0到wf do**

**3 d = Cf (u，v，1)T**

**4 dr = Rd**

5 (ue，ve) =项目(dr)

6 (uc，vc) =压缩(ue，ve)

7 Fout(u，v) = LookUp(FCoRE，uc，vc)

**8返回Fout**

*e c 0 c*

1 c 2 e c

0°c一

系数的计算方法是[Equation 4,](#_bookmark11) 如所示[Equation 6.](#_bookmark14)沿着bac段，扩张缓慢开始，没有接近b的缩放，然后加速，将ac一直推到ae。最差的采样速率出现在CoRE帧的外边缘，其中一个像素步长对应于扩展帧中的2v0/he 1像素步长。外围区域的其他三个部分类似地扩展。

−

*A0 =(he v0)/H2，a1 = 2v 0/he 1，a2 = 0 (6)*

*e*

1. *核心帧解码*

客户端从CoRE块计算输出帧。首先对块进行低级解码，以提取CoRE帧。然后，从相应的CoRE帧计算每个输出帧。CoRE块的主要部分被回放。然后，如果下一个数据块被延迟，则使用等式回放扩展部分[2](#_bookmark5) 和[3](#_bookmark8) 找出给定输出帧要使用的CoRE帧的索引j。

通过解码CoRE计算输出帧Fout

用...框住[algorithm 3.](#_bookmark13) 矩阵Cf表示当前用户视图。Fout的每个像素(u，v)是通过在FCoRE(第1-7行)中找到其对应的像素(uc，vc)来计算的。该映射取消当前像素(u，v)到3D方向d(第3行)的投影，在CoRE坐标系中将d旋转到dr(第4行)，将dr投影到(ue，ve)处的等矩形帧上(第5行)，并通过压缩外围区域(第6行)将(ue，ve)移动到CoRE坐标(uc，vc)。压缩与中描述的扩展相反[§II-B.](#_bookmark9) 像素(u，v)由FCoRE在(uc，vc)处的双线性插值设置(第7行)。

[Figure](#_bookmark10)3e显示了从CoRE帧(d)解码的输出帧。由于视图预测误差，实际FoV超出了预测视图(e中的黑线)和CoRE中心区域(e中的红线)。输出帧(e)在使用核心区域像素的地方(红线的左下方)具有全分辨率，在使用外围区域像素的地方(红线的右上方)具有平滑降低的分辨率。

1. *核心帧解码优化*

我们优化了从输出帧到CoRE帧的映射，如下所示。回头参考[algorithm 3,](#_bookmark13)相机矩阵Cf通过与R相乘进行预旋转(M = RCf)。那么dr是递增计算的，代价是

三个浮点相加，即dr(u，v) = dr(u 1，v) + Mi2，其中Mi2是M的第三列。等矩形投影(第5行)归一化dr，找到仅取决于dr.y的纬度ve，并找到仅取决于dr.x和dr.z的经度ue。纬度和经度分别在预先计算的1D和2D查找表中找到。压缩(第6行)也在预先计算的2D表中查找。因此，计算输出帧像素的成本是3次加法、一次3D矢量归一化、3次表格查找和一次双线性插值。

1. *自适应预取*

−

受TCP重传超时(RTO)计算算法的启发，我们自适应地确定预取每个CoRE文件的时间[[14].](#_bookmark48) 我们将预取时间更新为:平滑下载时间= (1 α)平滑下载时间+α上次下载时间；下载时间变化= (1 β)下载时间变化+β(平滑下载时间上次下载时间)；预取时间= γ平滑下载时间+ρ下载时间变化。

×

— ×

— × ×

×

— ×

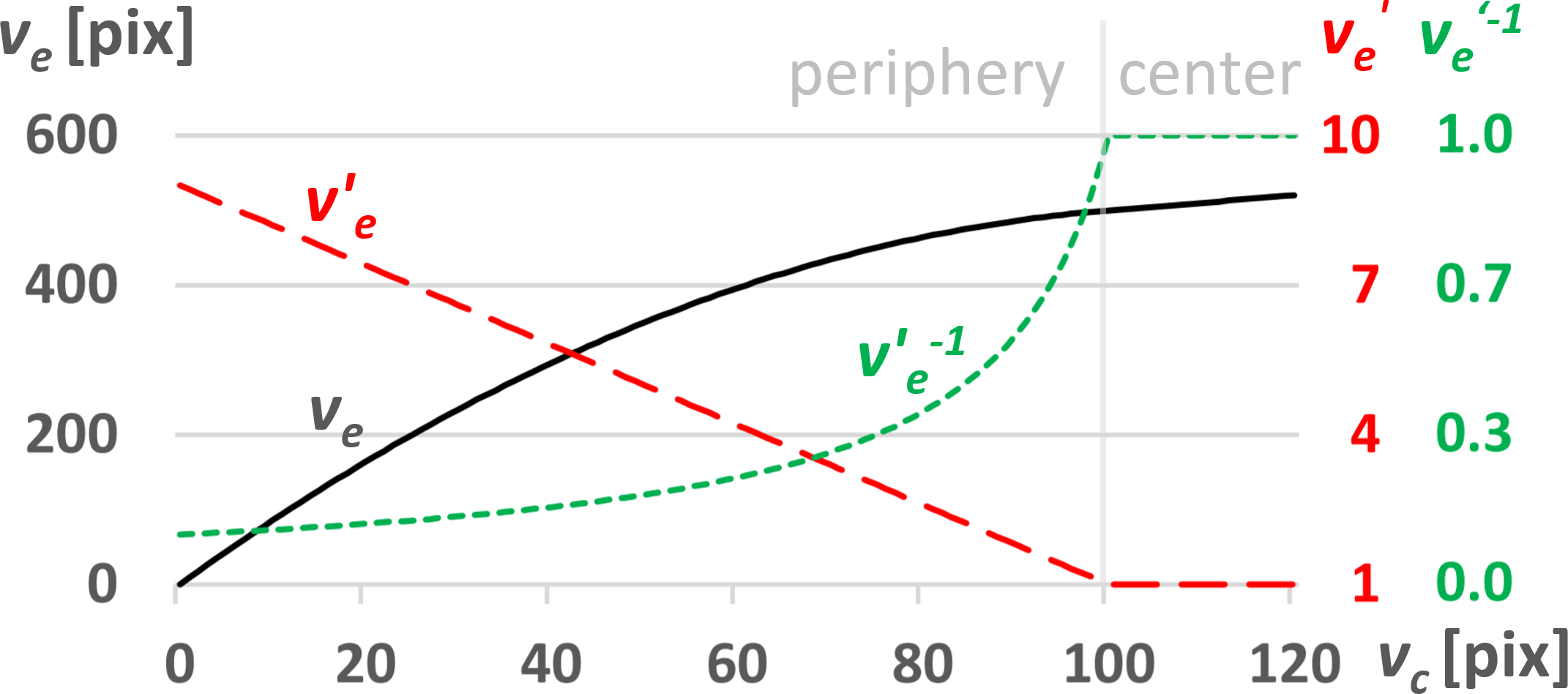
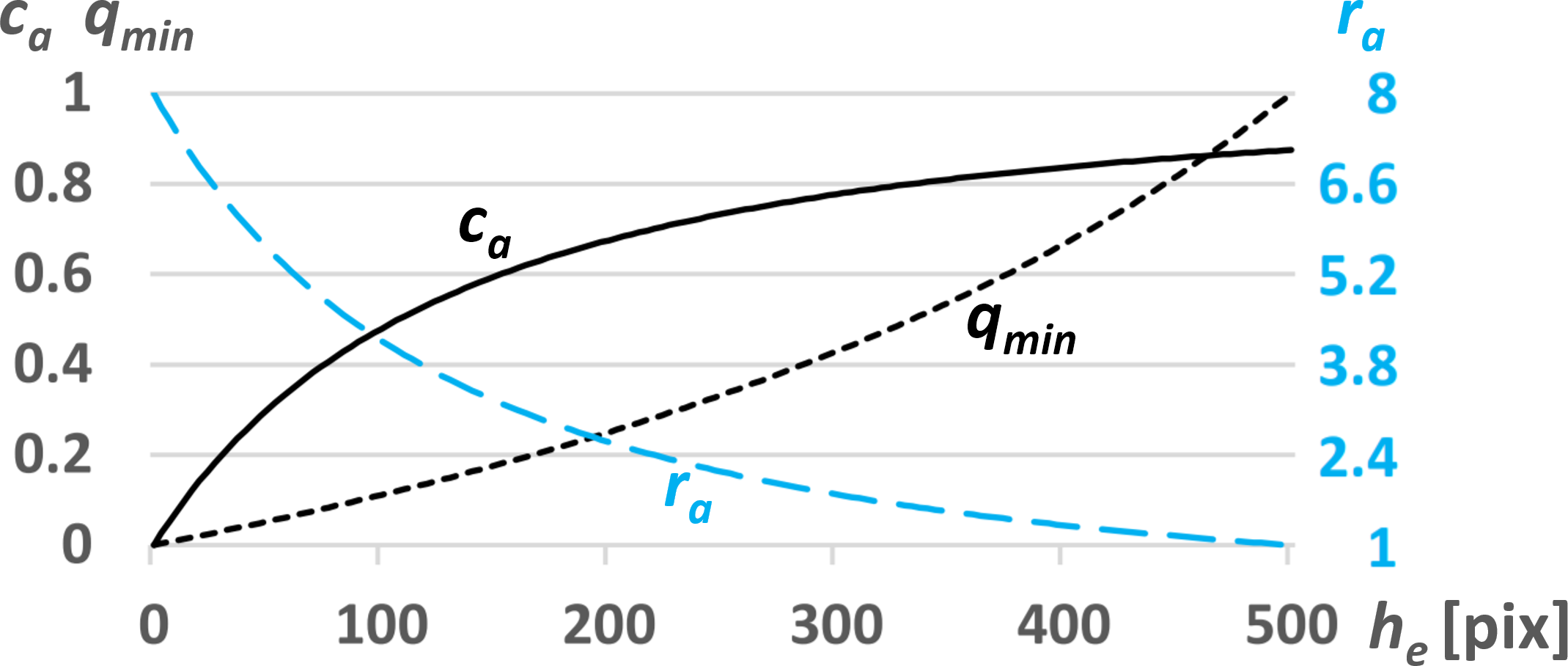
我们用不同的α、β、γ和ρ值进行实验。

我们以0.1的增量在0和1之间变化α和β，以0.5的增量在0.5和2之间变化γ和ρ。我们还比较了在当前区块结束之前使用1s、2s、3s或4s的固定预取时间。我们发现α = 0.9，β = 0.9，γ = 0.5，ρ = 1的自适应预取效果最好。因此，我们在本文的所有实验中使用这些值。

1. 核心流分析

在本节中，我们将分析CoRE的成本/收益权衡。

对视图预测误差的鲁棒性。CoRE帧是一个360°的帧，因此不存在丢失的像素，无论视图预测误差的大小如何。然而，视图预测误差越大，采样的CoRE帧离中心区域越远，图像质量越低。给定在(uc，vc)处的CoRE帧中查找的输出帧中的一个像素(u，v)，我们将(u，v)处的图像质量量化为(uc，vc)处的采样率。采样率q(vc)是沿着压缩方向的采样步长的倒数，即q(VC)= ve′(VC)1 = 1/ve′(VC)。

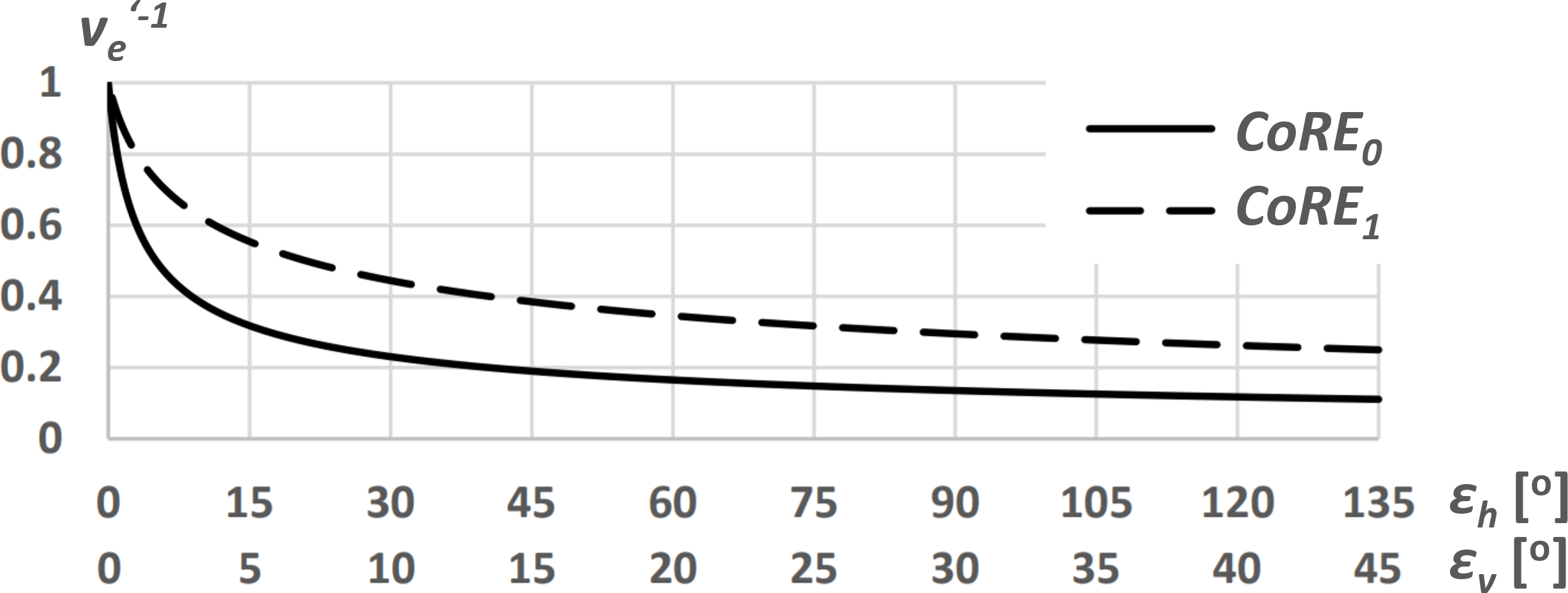
* 1. (二)(三)

图5: CoRE采样速率与外围坐标(a)和厚度(b)的关系，以及与视图预测误差(c)的关系。

假设中心区域的FoV为∏φ∏θ= 90°○90 ○,分辨率为w h = 1，000±1，000。如果外围区域的横向厚度是we = 300，基于的线1-2[algorithm 2,](#_bookmark6) 垂直厚度为he = 100，扩展外围区域的厚度为u0 = 1，500和v0 = 500。[Figure 5a](#_bookmark19) 显示了扩展垂直坐标ve、采样步长ve′和采样速率ve′1在外围区域(例如，从ae到b in)上对CoRE垂直坐标vc的依赖性[Figure](#_bookmark7) 4).ve具有大的初始速度，并均匀减速，直到外围区域和中心区域之间的边界，之后它与vc一对一地变化。采样步长ve′从ve′(0)= 9开始，并在边界处线性减小至ve′(100)= 1。采样速率ve′1从1/9开始，在边界处增加到1。平均采样步长(速率)为v0/u0 = 5 (1/5 = 0.2)。

× × ×

×

成本/收益的权衡是通过外围区域we的厚度来控制的，这也决定了he。[Figure 5b](#_bookmark19)显示了最低质量qmin，定义为CoRE帧边界处的采样速率ve′(0)1，CoRE开销ca，定义为整个CoRE帧区域上的外围区域面积，以及CoRE大小缩减ra，定义为扩展帧在CoRE帧区域上的面积。qmin对应于最大可能的视图预测误差，即，当用户朝与预测方向相反的方向看时。对于he = 100pix的薄边框，qmin为1/9，外围区域为CoRE帧的48%(ca = 0.48)，CoRE帧比传统的360帧小4倍以上(ra = 4.16)。he = 200pix的边界将qmin提高到0.25，成本ca = 0.67，同时仍然实现ra = 2.6的降低。

我们现在分析采样率对视图预测误差大小的依赖性。对于我们的示例，中心区域外的270°水平FoV左右均匀分布，因此最大可能的水平视图预测误差(ϵh)为135○。类似地，最大可能的垂直视图预测误差(ϵv)是45○。在…里[Figure 5c,](#_bookmark19)CoRE0配置的he = 100，ca = 0.48，ra =

4.16.对于高达15○ (5○)及以上的水平(垂直)视图方向预测误差，采样率高于0.25

误差高达45○ (15○)时为0.2。CoRE1配置的he = 200，ca = 0.67，ra = 2.6，采样速率高于0.4，误差高达45○ (15○)。

对链路带宽波动的鲁棒性。成本/收益

权衡通过两个参数来控制:持续时间

te和CoRE扩展中的帧数ne。应选择te以避免最长预期带宽降级的停顿，例如蜂窝拖车之间的切换。CoRE块可以避免高达t+te的停顿，其中

≠t是主要的持续时间。ne的选择应基于系统或应用程序认为可接受的开销，以及扩展期间所需的平均ωavg和最小ωmin帧速率。我们将开销ce定义为ne与主要部分帧数ω∏t之比，其中ω是主要部分帧速率。

*ωavg在ne中是线性的(即ωavg = ne/∏te)，ce也是如此(即ce = ne/(ω∏t))。ωmin是ne(ωmin = t′(ne)1)中的一个过线性，其中t由下式定义*[Equation 2](#_bookmark5) 和[Equation](#_bookmark8) 3).对于30个额外帧和6s扩展，帧速率从30fps降低到2.72fps，平均帧速率为5fps，开销为25%。对于有60个额外帧的CoRE配置，开销增加到50%，平均和最小帧速率分别增加到10fps和6fps。

1. 实验评估

我们首先给出我们的评估方法，然后给出我们的结果。

1. *评价方法*

我们进行了基于跟踪的实验，比较了在搭载英特尔酷睿i5 3320M 2.6 GHz双核处理器和8gb RAM的惠普EliteBook 8470p笔记本电脑的CPU上运行的CoRE和tiling的C++客户端和服务器实现。我们的C++实现由5，000多个LoC组成。我们还在搭载高通snapdragon 855八核(1x2.84GHz + 3x2.41GHz + 4x1.78GHz)处理器和安卓版本9的三星galaxy Tab S6上使用安卓客户端原型进行了用户研究。我们的安卓实现通过JNI重用了我们的C++代码，并增加了1000多个Java LoC。

* 1. *流式传输方法:我们将CoRE流式传输与以下切片基线算法进行比较，以流式传输存储的视频:(1)仅FoV(仅获取与用户视野重叠的切片)，(2) FoV+ 1QL(也获取视野外的填充切片，所有切片的质量级别相同)，(3) FoV+ 2QL(填充切片的质量级别低于FoV切片)，(4) FoV 360(获取所有切片，具有更高质量的FoV切片)和(5) FoV+ 360(获取所有切片，具有更高质量的FoV和填充切片)*

在我们的切片实现中，当停顿发生时，我们不会跳过任何视频帧；相反，我们回放整个

视频，由于停顿，回放可能需要比视频持续时间更长的时间。我们也停止头部运动。因此，我们在平铺帧质量方面报告的结果有利于平铺。我们还结合了我们为CoRE开发的解码优化[(§II-D)](#_bookmark16)以进行公平的比较。

FoV+ 2QL、FoV 360和FoV+ 360遵循最先进的360视频流系统采用的多分辨率方法[[6],](#_bookmark40) [[8],](#_bookmark42) [[17].](#_bookmark51) 由于我们无法获得这些系统的实现，我们的比较仍停留在算法层面，没有这些系统采用的优化。我们注意到CoRE与这些优化兼容，如中所述[§VI.](#_bookmark34)CoRE只是缓解了视角方向预测和瞬时带宽波动的挑战性问题。

对于切片，除非另有说明，否则我们使用Flare中使用的以下参数[[17]:](#_bookmark51)(1)每个视频块的持续时间为1s，(2)用户FoV为90○ 48○，(3)视频被分割成的片数为4 ^ 6，(4)填充量为20%，以及(5)所有高质量片的采样率为1，低质量片的采样率为0.5。

×

×

对于CoRE，除非另有规定，否则我们使用以下参数:(1)用户输出帧分辨率wf hf为960 512，(2)用户FoV∏φ∏θ为90○48 ○( 3)CoRE w h的编码高质量中心区域的分辨率为1，152 614，(4)完整CoRe帧的分辨率为

×

×

× × ×

×

高质量的中心和外围区域是16881068，

×

(5)在一个CoRE帧中，最大采样率为1(中心区域)，最小采样率为0.11，(6)在当前CoRE组块的主要部分的回放将完成之前，我们在1s-4s内自适应地请求下一个CoRE，(7)每个CoRE的实时时长为10s:一个4s的主要部分加上一个6s的扩展部分，(8)主要部分的帧率ω为30fps，扩展部分的ne = 30帧。

* 1. *视频:我们使用从YouTube上获得的六个视频进行实验:(1)水下潜水(diving)，(2)大象从湖中饮水(大象)，(3)纽约市全景(NY)，(4)巴黎全景(Paris)，(5)犀牛(Rhino)，以及(6)过山车之旅(roller)。所有视频的分辨率约为4k (3，840±2，048)。我们对每个视频进行大约一分钟的拍摄(但我们使用两分钟进行用户研究)，转换为。avi格式，并使用恒定速率因子(CRF) 30重新编码。产生的文件大小列于*[Table](#_bookmark22) 爱达荷（Idaho的缩写）

×

表一:实验中使用的视频

视频视频视频文件大小(兆字节)

原装4s 1s

跳水2OzlksZBTiA 69.5 69.6 75.15

大象2bpICIClAIg 39.6 40.0 43.8

纽约ciw 8r 8 th M8 46.7 48.5 55.3

巴黎sJxiPiAaB4k 15.2 18.5 36.2

犀牛7IWp875pCxQ 13.2 15.2 26.75

滚轮8 LSb-P8Gsm 49.3 49.7 55.22

* 1. *头部运动:我们使用一分钟的头部运动轨迹*[[4]](#_bookmark38) 基于Corbillon等人的工作。[[3].](#_bookmark37) 我们列出

中每个视频的跟踪总数[Table II.](#_bookmark23) 除了

注入静脉的[-B6,](#_bookmark28)我们在提取时使用当前视图方向信息。这不代表头部运动预测，对于平铺和CoRE来说都是一个具有挑战性的案例。

表二:实验中使用的头部运动轨迹

视频潜水大象纽约巴黎犀牛滚筒

痕迹60 38 57 58 25 59

* 1. *网络条件:对于我们基于轨迹的实验，我们使用从Mahimahi项目存储库中获得的带宽轨迹*[[12],](#_bookmark46) 这些数据是由温斯坦等人使用饱和器工具收集的。[[23].](#_bookmark57)我们选择这些轨迹是因为据我们所知，它们在所有公开可用的轨迹中具有最高的精度，即毫秒级。中列出了我们使用的四种跟踪[Table III.](#_bookmark24)请注意，表中列出的平均带宽以KBps为单位，以便与以字节为单位的传输量进行比较。由于这是头部运动轨迹的长度，所以在大约前60秒计算平均值[[4].](#_bookmark38)

表三:实验中使用的带宽轨迹

迹线标识平均带宽(KBps)行驶速度下降121

ATT-LTE-驾驶-2016.down 576

t移动电话-长期演进-行驶中。下降1593

威瑞森-长期演进-行驶。下降1899年

* 1. *性能指标:我们在质量/成本权衡方面比较了CoRE和tiling。质量通过三个帧内(a-c)和两个帧间(d-e)度量来衡量，成本通过两个度量(f-g)来衡量。*

1. 缺少像素:输出帧像素映射到从服务器传输的360帧的一小部分之外。
2. 采样率:采样步长的倒数[(§III),](#_bookmark18)100，覆盖所有输出帧的所有像素。

×

1. 采样率的突然变化:至少有一个相邻像素质量较低的输出图像像素的数量，为使用多个离散分辨率级别的方法定义。
2. 帧率:播放的总帧数除以播放时长。对于切片基线，如果至少FoV切片已准备好，即使填充切片尚未准备好，我们也认为下一帧已准备好。
3. 暂停:当输出帧无法更新，因为播放时间已达到接收数据覆盖的最后一个时间步长时，就会出现暂停。对于CoRE，当扩展的最后一帧被使用时，暂停开始。对于切片，当使用当前块的最后一帧时，暂停开始；同样，我们不会等待填充块的到来。
4. 传输量:在整个实验期间传输的数据总量。
5. 低级解码开销:视频编解码器在客户端为低级解码消耗的时间和精力。与我们的详细质量指标(a-c)不同，PSNR等一次总付指标无法检测和量化在少量像素上出现的误差，如采样率不连续或少数缺失像素，因为它对这些误差进行了平均

像素与帧中的其他像素。此外，PSNR取决于每个视频的内容；例如，PSNR只惩罚低分辨率，如果它是在一个有精细细节的区域；使用像PSNR这样的指标对流媒体进行可靠评级需要对大量视频和用户跟踪进行测试。

1. *基于痕迹的实验结果*

我们首先使用中列出的美国电话电报公司网络跟踪评估核心和切片[Table III.](#_bookmark24)该轨迹显示带宽波动，尤其是在20至30秒内，带宽显著下降。因此，对于切片和CoRE来说，这都是一个具有挑战性的案例。我们用IV中剩余的三个网络轨迹进行实验 [-B5.](#_bookmark27)

* 1. *缺失像素和质量变化:*[Figure 6](#_bookmark29)显示缺少至少一个像素(1MP)的帧的百分比[Figure 7](#_bookmark29)描绘了在使用不发送一帧的所有切片的切片变体(即FoV、FoV+ 1QL和FoV+ 2QL)时，任何帧中丢失像素的最大百分比(MMP)。由于FoV+ 1QL和FoV+ 2QL使用同一组图块(尽管质量水平不同)，因此我们只显示FoV+ 1QL的结果。此处以及整篇论文其余部分的误差条代表95%的置信区间(基于正态分布计算)。

对于图形中的仅视场平铺[6](#_bookmark29) 和[7,](#_bookmark29) Rhino视频的1MP和MMP值最低:40%的播放帧至少有一个像素丢失，至少有一个帧丢失了高达81%的像素。NY视频最高1MP为60.3%，MMP为95%。正如预期的那样，填充减少了丢失的像素数量，但并没有消除它们。对于FoV+ 1QL和FoV+ 2QL平铺，我们在图中看到了NY (Rhino)视频的14(6)% 1MP和51(33)% MMP[6](#_bookmark29)和[7.](#_bookmark29)因此，对于仅FoV平铺，用户应该每两帧注意到丢失的像素，并且至少有一帧丢失了许多像素。对于FoV+ 1QL和FoV+ 2QL平铺，用户每隔五到十帧就会遇到丢失像素的情况，并且至少有一帧会丢失大约一半的像素。有了CoRE，永远不会有任何像素丢失。现在，我们用两个质量级别量化切片的采样率变化。[Figure 8](#_bookmark29) 显示了质量级别更改边界(1AP，蓝色条)上至少有一个像素的帧的百分比，以及一帧中边界像素的最大数量(乘以100，标记为MAP并以红色显示)。将近一半的帧在质量变化边界上有像素，在一些帧中有多达1100个像素。质量水平的突然变化会对用户体验产生负面影响[[2].](#_bookmark36)

∼

对于CoRE，采样速率变化平稳。

* 1. *转账金额:*[Figure 9](#_bookmark29)比较使用切片和CoRE传输的数据量。我们观察到，在大多数情况下，CoRE传输的数据比FoV+切片变体少

优质瓦片，使得FoV+ 1QL比FoV 360更消耗带宽。不出所料，FoV+ 360传输的数据最多。

* 1. *帧速率和停顿:六个视频的帧速率和停顿的平均值和置信区间如所示*[Figure 10.](#_bookmark30)从图中可以明显看出，CoRE几乎没有停顿，接近30fps，而所有切片方法都存在停顿持续时间长、帧速率低的问题。在切片方法中，仅FoV切片传输最少，因此在平均帧速率和停滞持续时间方面都获得了更好的性能。但是，仅FoV平铺会丢失像素(四 [-B1).](#_bookmark25)
  2. *低级视频解码开销:一旦客户端从服务器接收到视频文件，客户端就必须使用标准视频编解码器(例如H.264)对其进行解压缩。当视频被分割成单独解码的较小视频时，这种低级视频解码的累积开销会增加。*[Figure 11](#_bookmark32)显示了时间和能耗方面的视频解码开销。计时数据是在我们的笔记本电脑上收集的，而能耗则是在我们的平板电脑上通过计算所有轨迹上电池电量从充满到下降的平均值来测量的[(Table](#_bookmark23)二)。CoRE需要每4s解码一个文件，它的开销比tiling小得多，tiling需要对FoV+方法解码大约50个文件，对360个方法解码6 4 4 = 96个。我们得出结论，CoRE提供了显著的解码节省。

× ×

* 1. *网络条件的影响:我们现在使用中给出的所有网络跟踪*[Table III.](#_bookmark24) 由于篇幅的限制，我们只讨论了潜水视频的结果，但我们对其他视频进行了实验，发现结果是相似的。我们观察到，对于四个网络带宽轨迹，CoRE的平均采样速率分别为62.18%、62.19%、65.33%和66.69%。[Figure 12](#_bookmark33) 比较所有切片方法和CoRE的平均帧速率和停顿持续时间。我们观察到CoRE优于所有平铺变体。

我们还实验了不同的中心和外围区域大小(1152 614和1323 1053)，以及不同的延伸长度(6s和9s)。我们发现，在缩小外围区域的同时扩大中心区域，保持CoRE帧尺寸不变，会提高平均采样率，但代价是增加文件大小。增加扩展部分的持续时间有助于掩盖更长的带宽下降，同样是以更大的文件为代价。注意，由于不同的预测视图和降低的帧速率，扩展对于下一个块只有有限的冗余。

× ×

* 1. *视角方向错误预测的影响:我们将视角预测的准确度从100%变化到60%。根据预测的和实际的用户视图方向之间的程度差异来计算预测精度。x%的预测精度意味着预测的视图方向是*

经度和纬度为180 ×( 100 x)和90 ×( 100 x)

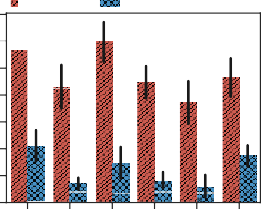
100 100

案例，从而节省网络带宽和设备能源。不出所料，仅FoV平铺在平铺变体中转移的数量最少，其次是FoV+ 2QL、FoV 360、FoV+ 1QL和FoV+ 360。传输的数据取决于传输的切片数量和切片质量。FoV 360发送的瓦片比FoV+ 1QL多，但后者的更高

远离用户实际观看方向。

我们使用了威瑞森带宽追踪[(Table III)](#_bookmark24) 因为它的高带宽降低了网络对性能的影响。我们观察到采样率与预测精度相关，例如，将预测精度从100%降低到60%，将采样率降低38%、42%和43%

70



仅FoV

FoV+ 1QL

60

50

1MP (%)

40

30

20

10

0

潜水大象纽约巴黎犀牛滚筒

图6: 1MP (%)。

100

80

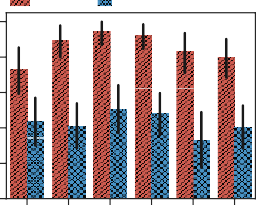
60

基质金属蛋白酶(%)

40

20

0



仅FoV

FoV+ 1QL

潜水大象纽约巴黎犀牛滚筒

图7: MMP (%)。

120

100

80

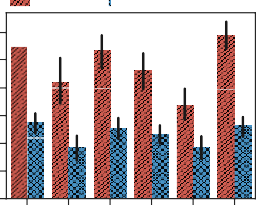
价值

60

40

20

0



地图(x100)

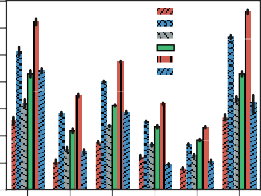
1AP(%)

潜水大象纽约巴黎犀牛滚筒

图8: 1AP和MAP。

70

60



仅FoV FoV+1QL FoV+2QL FoV 360

FoV+ 360 CoRE

传输的数据(兆字节)

50

40

30

20

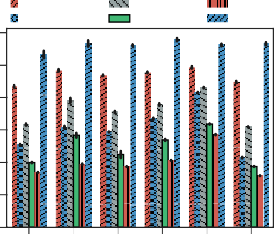
10

0

潜水大象纽约巴黎犀牛滚筒

图9:传输的数据。

30



仅FoV FoV+2QL FoV+360

FoV+ 1QL FoV 360 CoRE

25 150

帧速率(fps)

125

20

失速

100

15

75

10

50

5 25

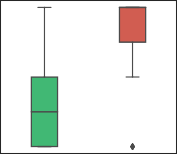
仅FoV FoV+1QL

FoV+ 2QL FoV 360

FoV+ 360 CoRE

缺少像素(p < 0.001)

5



四

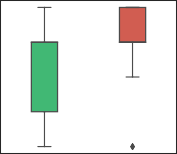
3

2

一

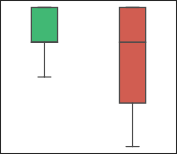
平铺CoRE

低帧率(p < 0.001)



平铺CoRE

模糊度(p < 0.001)



平铺CoRE

0

潜水大象纽约巴黎犀牛滚筒

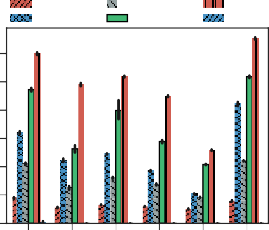
1. 平均帧速率

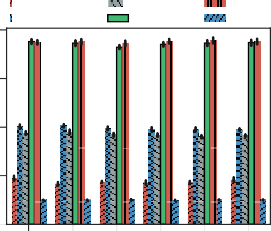
0

潜水大象纽约巴黎犀牛滚筒

1. 失速持续时间

图13:用户研究的结果。

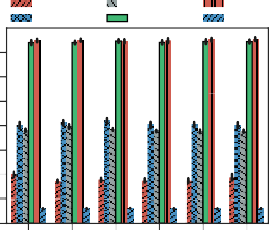
图10:美国电话电报公司带宽跟踪的帧速率和停顿。



仅FoV FoV+2QL FoV+360

FoV+ 1QL FoV 360 CoRE

0.8



仅FoV FoV+2QL FoV+360

FoV+ 1QL FoV 360 CoRE

350

观看视频时，可以滑动客户端平板电脑屏幕，向上、向下、向左或向右更改观看方向。平板电脑和笔记本电脑通过手机创建的热点与美国电话电报公司4G LTE服务连接。无线网络平均值

300

解码时间(秒)

250

200

150

100

50

0

潜水大象纽约巴黎犀牛滚筒

时间

0.6

0.4

电池耗尽(%)

0.2

0.0

潜水大象纽约巴黎犀牛滚筒

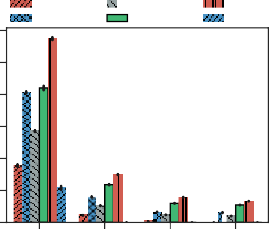
能源

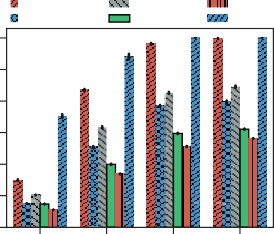
下载速度约为每秒1.03 MB(为了便于比较，我们再次使用了Bytes)。

30名参与者，年龄从18岁到52岁不等，包括男性和女性，观看了6个视频并对其进行了评级(第四部分中的潜水、犀牛和滚轴视频[-A2](#_bookmark21) 两种流式传输方法中的每一种)。对于每个参与者，六个视频的顺序被打乱。我们回放了三个滑动运动轨迹

图11: H.264视频解码开销。

捕获(三个视频各一个)以生成参与者评价的六个视频。这有给予的好处

30 600



仅FoV FoV+2QL FoV+360

FoV+ 1QL FoV 360 CoRE

25 500

帧速率(fps)

20 400

失速

15 300

10 200

5 100

0 0

仅FoV FoV+1QL

FoV+ 2QL FoV 360

FoV+ 360 CoRE

所有参与者都有同样的经历[[6].](#_bookmark40)

[Figure 13](#_bookmark31)显示了参与者对三个问题的评分:缺失像素、低帧速率(不连续)和模糊度(块状)，评分范围为1(强烈不同意)到5(强烈同意)。如图所示，CoRE在丢失像素和低帧率方面优于(平均值低于)平铺，但在模糊度较低方面优于CoRE。模糊结果是预期的，因为

美国电话电报公司移动威瑞森移动

1. 平均帧速率

美国电话电报公司移动威瑞森移动

1. 失速持续时间

CoRE通过降低外设采样率来避免丢失像素。对于平铺，我们使用的是FoV+ 1QL平铺基线，

图12:四条网络带宽轨迹的潜水视频。

潜水、犀牛和滚轴视频。我们使用当前视图方向而没有任何预测的实验对应于大约80%的预测精度。

1. *用户研究*

我们在第四章描述的三星平板电脑上使用我们的CoRE Android客户端原型[-A](#_bookmark20)使用CoRE和FoV+ 1QL平铺进行IRB批准的用户研究。我们用大约两分钟的时间观看三个视频，这些视频由运行在上述实验中使用的同一台笔记本电脑上的服务器进行流式传输。这

它使用单一分辨率。

我们现在测试一个假设，即CoRE和tiling的响应分布在三个指标上是相等的。给定两个种群P1和P2，我们评估零假设:H0 : P1 = P2，以及替代假设，H1 : P1 = P2，其中P1和P2分别代表CoRE和tiling的观测值。由于分布是正态的，我们使用t检验[[7]](#_bookmark41) 用p值阈值α = 0.05来验证零假设。我们发现，对于(1)缺失像素的存在(p < 0.001)、(2)低帧速率(p < 0.001)和(3)模糊度(p < 0.001)，差异具有统计学意义[(Figure](#_bookmark31) 13).

/

1. 相关著作

与我们最密切相关的工作是Flare[[17],](#_bookmark51)Rubiks[[8],](#_bookmark42)Pano[[6],](#_bookmark40)ClusTile[[29],](#_bookmark63)POI360[[26],](#_bookmark60)360prob-DASH[[25],](#_bookmark59)Oculus360[[28],](#_bookmark62)和5G网络的流媒体[[19].](#_bookmark53)Flare[[17]](#_bookmark51)将机器学习技术应用于视图方向预测。由于预测不完善，Flare会在FoV周围发送额外的切片，并开发调度算法和系统优化来提高输出视频质量。Rubiks[[8]](#_bookmark42) 通过将时间分段成层来增强切片，从而平衡解码时间、视频质量和带宽消耗之间的权衡。Pano[[6]](#_bookmark40)使用可变大小的瓷砖——这是ClusTile提出的一个想法[[29].](#_bookmark63)在POI360[[26],](#_bookmark60)每个视频帧在空间上被分割成96个切片，根据它们与FoV中心的距离按比例缩小。在一个图块中，质量等级是恒定的。360probDASH[[25]](#_bookmark59) 通过解决QoE驱动的优化问题，确定要预取的切片及其质量级别。孙等[[19]](#_bookmark53)通过5G发送多层编码视频。CoRE受到这些论文中的想法的启发，但采取了一种替代平铺的方法，即逐渐降低分辨率，逐渐降低帧速率，并将整个区块存储在一个文件中。正如我们在中所讨论的，CoRE可以结合上述工作中的一些优化[§VI.](#_bookmark34)

Oculus流使用基于修改的立方体投影的360°视频的非线性重映射，将更多像素分配给正面的中心区域[[28].](#_bookmark62) 与CoRE不同的是，Oculus框架在中心线处是不连续的，它从左-前-右切换到上-后-底面，分辨率固定在三个可能的级别之一，缺乏灵活性。

CoRE旨在通过降低帧速率扩展来提高对瞬时带宽波动的鲁棒性。可以与CoRE集成的一个补充思想是预测可用带宽，并使用它来决定下一个块的比特率。先前关于比特率适配的工作包括MPC[[27],](#_bookmark61) BOLA[[18],](#_bookmark52) 和Pensieve[[13]](#_bookmark47) 用于存储视频，和Jigsaw[[1]](#_bookmark35)和MPC-Live[[20]](#_bookmark54)视频直播。这些研究使用控制理论、优化、机器学习或分层编码技术来使用户体验质量适应可用带宽和可用缓冲空间。几种360°视频流方法[[6],](#_bookmark40)[[8],](#_bookmark42)[[17]](#_bookmark51)已经采用了MPC。最后，非线性采样早已在可视化中得到研究[[5],](#_bookmark39)[[11],](#_bookmark45)[[15],](#_bookmark49)[[16],](#_bookmark50)[[24].](#_bookmark58)可变帧速率低级编码已经被探索用于帧不常改变的视频[[21].](#_bookmark55)

1. 讨论

**存储。**CoRE块的扩展部分中的帧速率降低意味着连续帧之间的不相似性增加，因此压缩效率降低(例如，对于潜水视频，平均扩展帧大小为0.0265MB，而主要部分为0.0156MB)。即便如此，覆盖主要部分4s之外的6s扩展仅花费总块大小的30%左右。

CoRE块是基于预测的用户视图方向构建的。一种方法是服务器构建

一旦CoRE块接收到预测的视图方向，它就会按需运行。这与客户数量不成比例。我们采用的另一种方法是预先计算和存储覆盖所有可能视图方向的CoRE块。我们使用水平和垂直方向的20°离散化:赤道有18个区块，随着纬度向两极增加，这个数字逐渐减少，总共有105个CoRE区块。服务器使用DASH传输视图方向与客户端请求最匹配的CoRE块。这种方法可以很好地适应客户机的数量，但代价是在服务器上增加存储。通过只存储视频中感兴趣区域的CoRE块，可以减少存储空间。例如，对于Roller视频，在中的所有跟踪中仅请求了15.6%的CoRE区块[Table II.](#_bookmark23)

**CPU对GPU。**CoRE旨在支持没有GPU的客户端:只需对单个视频文件进行低级解码(IV[-B4),](#_bookmark26)高级CoRE解码经过优化[(§II-D).](#_bookmark16)平铺需要对几十个视频文件进行低级解码，这就排除了只使用CPU的实现[(Figure 11a),](#_bookmark32)因此，GPU是当前工作的重点[[6],](#_bookmark40)[[8],](#_bookmark42)[[17].](#_bookmark51) 如果有图形处理器，CoRE可以从中受益，实现更快的低级解码。在我们的安卓设备上，在GPU上对CoRE块进行低级解码比在CPU上快9.7倍；对于切片，GPU相对于CPU的优势仅在于

×

3.5，因为图块的数量超过了GPU的硬件解码器数量。在能耗方面，对于CoRE，GPU比CPU有3倍的优势，对于tiling，GPU比CPU有1.1倍的优势。我们得出结论，CoRE在图形处理器上的性能优于切片，甚至比在中央处理器上的性能有更大的优势。

×

× ×

**与其他方法的集成。**最先进的360视频流系统[[6],](#_bookmark40)[[8],](#_bookmark42)[[17],](#_bookmark51)[[29]](#_bookmark63)提出了提高性能的强大优化。这些优化中有几个可以整合到CoRE中。例如，尽管CoRE中的非线性采样提供了对视图预测误差的鲁棒性，但是视图轨迹预测技术，例如Flare采用的技术[[17],](#_bookmark51)可以轻松与CoRE集成。协议栈中的一些Flare系统优化也可以集成到CoRE中，例如请求取消。Rubiks中提出的优化低级编解码器[[8]](#_bookmark42) 也可以由CoRE使用。MPC的带宽预测和比特率自适应思想[[27],](#_bookmark61) 闪光[[17],](#_bookmark51) 红宝石[[8],](#_bookmark42) 和帕诺[[6]](#_bookmark40) 可以合并到CoRE中，以选择要向服务器请求的后续CoRE块的比特率。

1. 结论

我们提出了CoRE，这是一种在空间和时间域进行非线性采样的有效方法，可以提高360°视频流对视角方向预测误差和瞬时带宽波动的鲁棒性。我们对CoRE原型的实验表明，CoRE节省了带宽，没有丢失像素，几乎没有停顿。CoRE不需要GPU就能在客户端达到30fps。作为未来的工作，我们计划探索分布式应用程序的类似图像概括，它不仅允许客户端改变视图方向，还允许客户端改变视点。

参考

1. 《稳健的4K现场视频流》。在MOBICOM会议录(2019)中。
2. 传统和全向媒体的自适应流式传输。<http://conferences.sigcomm.org/sigcomm/2017/files/tutorial-adaptive->streaming.pdf，2017年。ACM SIGCOMM 2017教程。
3. 360度视频头部运动数据集。《多媒体系统会议录》(2017年)，第199-204页。
4. 360度视频头部运动数据集。[http://dash.ipv6.enstb.fr/headMovements/,](http://dash.ipv6.enstb.fr/headMovements/)2019.
5. 广义鱼眼观点。在计算系统中人的因素的SIGCHI会议记录(纽约，纽约，美国，1986)，CHI '86，ACM，第16-23页。
6. 关，杨，郑，c，张，x，郭，z，和江，J. Pano:通过更好地理解质量感知来优化360视频流。在SIGCOMM会议录(2019)中。
7. H.J. SELTMAN。实验设计和分析。<http://www.stat.cmu.edu/>hseltman/309/Book/，2019年。
8. 何，j，库雷希，M. A，邱，l，LI，j，LI，f，和HAN，L. Rubiks:实用的智能手机360度流。《移动机器人学报》(2018年)。
9. ISO。HTTP上的动态自适应流。[https://www](http://www.iso.org/standard/75485.html). iso或[g/standard/75485.html](http://www.iso.org/standard/75485.html)2019.
10. 视频流质量影响观众行为:使用准实验设计推断因果关系。在2012年互联网测量会议记录(美国纽约州纽约市，2012年)中，国际管理委员会第12届会议，美国计算机学会，第211-224页。
11. 双曲线浏览器:一种可视化大层次的焦点+上下文技术。信息可视化阅读，S. K .卡德，J. D .麦克嵌体，和b .施耐德曼，编辑。摩根·考夫曼出版公司，旧金山，加利福尼亚州，美国，1999年，第382-408页。
12. MAHIMAHI项目。Mahimahi无线网络跟踪。https://github.com/ravinet/mahimahi/tree/master/traces, 2019。
13. 神经自适应视频流。在ACM数据通信特别兴趣小组会议记录(美国纽约州纽约市，2017年)中，SIGCOMM '17，ACM，第197-210页。
14. 计算TCP重传定时器。https://tools.ietf.org/html/rfc6298,，2011年6月。
15. 适马透镜:结合空间、时间和半透明性的焦点-背景转换。在CHI ' 08:SIGCHI计算系统中的人的因素会议(意大利佛罗伦萨，2008年4月)的会议记录，美国计算机学会，编辑。，ACM，ACM。
16. 通用针孔照相机:可视化的有效和高效的非均匀采样。IEEE可视化和计算机图形学学报16，5 (2010)，777–790。
17. 移动设备的实用视口自适应360度视频流。在MOBICOM会议录(2018)中。
18. 在线视频的近最佳比特率适配。在电气和电子工程师协会信息通信会议记录(2016年4月)中。
19. 孙立军、端木、方、刘、于、王、于、叶、于、、戴，
20. 5G网络中的多路多层360度视频流。在Proc中。美国计算机学会多媒体系统会议(2018年)。
21. 低延迟状态下视频直播的最优策略。在IEEE ICNP会议录(2019)中。
22. 伟大的视频和音频压缩:主要提示和常识。泰勒和弗朗西斯美国，2009年。
23. 王，j，ZHENG，y，NI，y，XU，c，QIAN，f，LI，w，JIANG，w，CHENG，y，CHENG，z，LI，y，XIE，x，SUN，y，AND WANG，z .高速铁路上基于LTE的TCP性能的主被动测量研究。在Proc中。ACM MOBICOM的主席(2019年)。
24. 随机预测通过蜂窝网络实现高吞吐量和低延迟。作为第十届USENIX网络研讨会的一部分

*系统设计和实施(NSDI 13)(伊利诺伊州隆巴德，2013年)，第459-471页。*

1. 管理图中边缘拥挤的交互方法。2003年信息可视化研讨会。第03期8714(2003年10月)，第51-58页。
2. 谢，l，徐，z，BAN，y，ZHANG，x，AND GUO，Z. 360probdash:使用基于瓦片的http自适应流来提高360视频流的qoe。在第25届美国计算机学会多媒体国际会议(纽约，纽约，美国，2017)会议录中，MM '17，计算机械协会，第315-323页。
3. 谢和张:LTE蜂窝网络上的全景移动视频电话。在第13届新兴网络实验和技术国际会议(2017年，美国纽约)会议录中，CoNEXT '17，计算机械协会，第336-349页。
4. 动态自适应视频流的控制理论方法。在2015年ACM数据通信特别兴趣小组会议记录(美国纽约州纽约市，2015年)中，SIGCOMM '15，计算机械协会，第325-338页。
5. 眼360度视频流的测量研究。在第八届多媒体系统会议(美国纽约州纽约市，2017)的会议记录中，MMSys'17，计算机械协会，第27-37页。
6. 周，c，肖，m，和刘，y。克莱斯特尔:朝着最小化360度视频流的带宽。在IEEE INFOCOM 2018中

*- IEEE计算机通信会议(2018年)，第962-970页。*