流行感知360度视频流

宾夕法尼亚州立大学计算机科学与工程系

电子邮件:xuc23，txt51，gxc27@psu.edu

***摘要—基于切片的流技术已被广泛用于节省360视频流的带宽。然而，确定直接影响带宽使用的正确区块大小是一项挑战。为了解决这个问题，我们建议通过考虑观看流行度来编码视频，其中流行观看的区域被编码为宏块以节省带宽。我们提出了识别和构建宏块的技术，并考虑到实际问题(如头部移动的随机性)来调整宏块的大小。在某些情况下，用户的观看区域可能没有被构建的宏块覆盖，然后使用传统的平铺方案。为了支持流行感知360度视频流，客户端选择具有正确质量级别的正确切片(一个宏切片或一组传统切片)，以在带宽限制下最大化QoE。我们将这个问题公式化为一个NP难的优化问题，然后提出一个启发式算法来求解。通过基于真实轨迹的广泛评估，我们证明了所提出的算法能够显著提高QoE并节省带宽使用。***

1. 介绍

360度视频在YouTube、脸书等视频平台越来越受欢迎[1，2]。由于在相同的感知质量[3，4，5]下，360度视频比传统视频大得多，因此流传输360=度视频更具挑战性，尤其是在带宽有限的无线(例如蜂窝)网络上。

许多研究人员[6，7，8，9]通过仅下载部分视频数据来应对这一挑战。由于移动设备的视野(FoV)有限，在给定时间只能观看一部分下载的视频。因此，为了节省带宽，应该只下载该FoV内的视频数据，而不是整个视频。为了实现这个想法，一个广泛使用的方法是基于切片的流[6]。在这种方法中，视频被分成一系列视频片段，每个片段包含固定的视频持续时间。每个片段被进一步分成不重叠的可独立解码的块，每个块被编码成具有各种质量的多个副本。基于视场预测和带宽估计，用户可以获取以正确质量水平编码的切片子集，从而可以在不损害体验质量的情况下减少带宽使用。

切片大小会显著影响要下载的数据量。将视频分成小块会降低视频编码的效率。视频编解码器，如H.264 [10]和H.265 [11]，使用运动补偿预测技术进行视频压缩，其中视频帧通过参考过去或未来的视频帧进行编码。分割视频

分成小尺寸的可独立解码的块减少了每个块内这种参考帧的池，然后降低了压缩效率。因此，每个编码区块的数据大小将更大，并且将消耗更多的带宽。相比之下，虽然大尺寸切片可以提高压缩效率，但必须下载更多的FoV之外的数据，从而浪费更多的带宽。因此，使用正确的图块大小对视频进行编码非常重要。

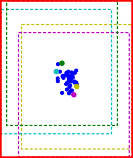
在本文中，我们建议通过考虑观看流行度来编码视频；即用户在观看相同视频时可能具有相似的观看兴趣(即观看区域)。通过将这些用户的观看区域编码为块(称为宏块)，可以实现高压缩效率，因为普遍观看的视频区域被编码成一个宏块，而不是被分成多个小块。此外，宏块在用户观看区域之外包含的数据更少，从而节省了带宽。构建宏块，我们面临以下挑战:(1)如何识别宏块？(2)如何确定正确的宏块大小？为了应对这些挑战，我们利用了观看同一视频的用户的历史观看数据。由于他们的共同兴趣，这些用户可能有相似的观看区域，并且他们的观看中心彼此靠近。我们首先识别这些观看中心并将它们聚集在一起，基于此我们可以识别宏块。由于头部移动的随机性，如果宏块太小，用户可能会在下载的宏块外观看视频。为了解决这个问题，宏块被构造成覆盖用户的观看区域加上一些边缘区域，这是基于用户观看中心的变化来确定的。

在一些情况下，用户的观看区域可能没有被构建的宏块覆盖，然后使用传统的平铺方案(即，4x6平铺方案)。也就是说，宏块被添加到传统的分块解决方案中，以减少大多数用户的带宽用量，而很少用户必须使用传统的分块方案。为了支持知名度感知360度视频流，我们面临以下挑战:(1)如何消除头部运动随机性的影响？(2)如何确定正确的瓦片(一个宏瓦片或一组传统瓦片)和正确的质量水平，从而在网络带宽受限的情况下使QoE最大化？为了应对这些挑战，我们提出了一种流行感知360度视频流算法，该算法首先预测每个视频片段的用户观看区域，然后在必要时预取相应的宏块或传统块。

(a)

一

0.8



压缩效率

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

0.6

0.4

0.2

0

1X1 4x4 4x6 4x8 6x12 8x12

平铺方案(行×列)

(b)

一

0.8

压缩效率覆盖效率

0.6

0.4

0.2

1.4

1.2

覆盖效率

诺姆。带宽使用

一

0.8

0.6

0.4

0.2

0

61%

固定平铺宏块

1X1 4x4 4x6 4x8 6x12 8x12宏块

平铺方案

(c)

图1: (a)使用4x6平铺方案分割视频。压缩效率和覆盖效率的比较。带宽使用的比较。

我们把这个问题表述为一个优化问题，并提出了一个算法来解决它。

本文的主要贡献如下。

* 我们建议通过考虑观看流行度来编码视频，其中流行观看的区域被编码为宏块以节省带宽。
* 我们将流行感知360度视频流问题公式化为一个NP难的优化问题，并提出了一种基于启发式的算法来解决它。
* 我们使用真实的头部运动数据轨迹来评估所提出的解决方案的性能。评估结果表明，该算法能显著提高用户体验质量，节省带宽使用。

论文的其余部分组织如下。我们在第二节介绍了背景和动机。第三节介绍了系统模型和问题表述。第四节介绍了我们的流行感知360度视频流算法。在第五节中，我们介绍了评估结果。第六节讨论了相关工作，第七节对论文进行了总结。

1. 背景和动机

图块大小会显著影响360视频流中要下载的数据量。为了说明这一点，我们进行了一些基于48个用户观看一个360度视频的头部运动数据轨迹的实验[12]。我们使用文献中常用的不同平铺方案分割视频，用(行x列)表示；即1x1(无平铺)、4x4 [13]、4x6 [3、14、15、16]、4x8 [17]、6x12 [18]和

8x12 [17]。为了公平比较，我们将FFmpeg [19]与编码器x264一起使用，使用相同的编码参数对切片进行裁剪和编码。用户的观看区域由观看中心和设备的FoV决定，其被认为是水平和垂直的100度[15，20，21，22，23]。图1(a)示出了被分成4行6列的视频，即24个块。图中的每个点代表一个用户的观看中心。虚线的黄色、青色、绿色和紫色块分别代表所有用户(即，图1(a)中的用户组)的最右边、最左边、最上面和最下面的观看区域。

图1(b)比较了不同平铺方案的有效性。覆盖率是FoV内的视频数据除以所有下载的视频数据。某些图块的边界区域可能不在FoV内，然后是边界

对于不同的切片方案，面积(覆盖率)会有所不同。为了比较这些分块方案，我们使用覆盖效率度量，即基于这些分块方案的最大覆盖比归一化的覆盖比。类似地，压缩效率是基于这些切片方案的最大压缩比归一化的压缩比，其中压缩比[23]定义为表示覆盖FoV的切片所需的总字节数除以以字节为单位的总编码数据大小。如图1(b)所示，较小的瓦片(例如8x12)可以实现更高的覆盖效率，而较大的标题(例如1x1)可以实现更高的压缩效率。请注意，较小的切片(例如8x12)具有较低的压缩效率，因为大量空间/时间冗余无法移除。例如，对于大切片，如果一个区域与其他区域具有相似的内容，则可以移除空间/时间冗余，但是如果使用小切片，这将是不可能的。

下载的视频数据(或带宽使用)与覆盖效率和压缩效率有关。图1(c)示出了不同分块方案的带宽使用，基于1×1分块方案归一化。如图所示，与传送整个视频的1x1切片方案相比，当视频被分成更小的切片，然后只下载覆盖用户观看区域的切片时，数据大小可以减小。然而，当切片太小(小于4x8)时，压缩效率太低，因此下载的数据变大。例如，8x12平铺的带宽使用变得比1x1平铺方案的带宽使用更大。因此，找到合适的切片方案来减少带宽使用是一个挑战。

如图1(a)所示，我们构建了一个大尺寸的图块(即红色块)，称为宏图块，它覆盖了所有的观看区域(虚线块)。这种宏块由于其大的块尺寸而具有高压缩效率。它具有高覆盖效率，因为只使用了一个大的瓦片，而不是由青色块表示的多达12个瓦片。相比之下，没有一种传统的分块方案能够同时实现高压缩效率和高覆盖效率。如图1(c)所示，与4x6平铺方案(最佳方案)相比，宏块可以进一步减少61%的带宽使用。

为了构建宏块，我们利用了观看同一视频的用户的历史观看数据。由于他们的共同兴趣，这些用户可能有相似的观看区域和他们的

观看中心彼此靠近。我们首先识别这些观看中心并将它们聚集在一起，基于此我们可以识别宏块。我们将在第四节中介绍宏瓦结构的细节。

1. 系统模型和问题表达

在这一节中，我们介绍了视频模型、QoE模型和问题公式。

其中Q0(k)Q0(k-1)表示段间临时质量变化(即第kth个和第(k-1)个视频段之间的质量变化)，k代表段内空间质量变化，其被计算为k [3]的标准偏差。

* *重新缓冲。再缓冲将显著影响生活质量*

− **

** * −* ** **

因为视频在重新缓冲事件期间会冻结。再缓冲时间如下。

*稀有*

1. *视频模型*

*I()=(s(k)l，0)*

(4)

视频被分成一系列视频片段，每个片段包含固定的视频持续时间。使用传统的平铺方案(例如，4x6)将每个片段进一步划分为C个平铺块。此外，还构造了M个宏块，细节将在第四节中描述。在服务器端，每个图块(和宏图块)被编码成对应于V个不同质量的V个副本。360♀视频流可以看作是一系列的下载任务。在每个任务中，客户端选择具有正确质量的正确切片(一个宏切片或一组传统切片)。让L表示客户端请求切片时，缓冲区中已下载但尚未观看的视频的视频长度，以秒为单位。为了避免停滞事件(或再缓冲)，图块应该

*r*

*k*

+

在客户端的视频播放器清空缓冲区(L = 0)之前完全下载。

1. *QoE模型*

对于每个视频片段k，类似于[3，24，25，26]，QoE模型通过考虑以下度量来量化用户感知质量:平均视频质量、质量变化和再缓冲。QoE模型如下:

*q(k)= q0(k)-ωviv(k)-ωrir(k)(1)*

其中S(k)是段数据大小，r是下载吞吐量，以及(x)+= 0. maxx

1. *问题定式化*

在本小节中，我们将流行感知360♀视频流问题形式化。让βv (βv)代表相应的宏块(或常规块)是否将被下载。具体而言，如果宏块m以高质量编码，则βv = 1

*m*

*c*

*m*

v级下载，带宽使用为Bv；

*m*

否则βv = 0。如果下载了以质量等级v编码的图块c，并且带宽使用率为Bv，则βv = 1；否则βv = 0。用户应该下载宏块来覆盖他的观看区域。如果这样的宏块不存在，或者不够

*m*

*c*

*c*

*c*

为了覆盖他的观看区域，将下载一组传统的瓦片。

在我们的知名度感知360♀视频流中，目标是最大化用户的感知QoE，这可以通过为每个视频片段选择具有正确质量水平的正确切片(一个宏切片或一组传统切片)来实现，以在网络带宽限制下最大化感知质量。优化问题的公式如下。

**马克斯·q(v∀m,vβv =1)+q(v∀c,vβv =1)(5)**

其中k代表瓷砖的视频质量

*m c*

*M V C V*

Σ Σ β

Σ Σ+ 1( *β ) = 1* (5a)

为k段下载，Q0是平均质量，Iv

质量缺陷是由质量变化引起的吗，Ir是

再缓冲事件引起的质量损害，以及ωv

**标准时间。**

*v m*

*m=1 v=1*

*V*

Σ

*五. c*

*c=1 v=1*

和ωr分别是质量变化和再缓冲的权重。Q0、Iv和Ir的定义如下。

* *平均质量。因为只有观看区域中的视频内容有助于用户感知质量，所以计算观看区域中所有图块的平均质量，如等式所示。2*

*Q0(k) = q(k) (2)*

其中k表示观看区域中的块的平均视频质量(即视频比特率), q(。)是一个映射函数，它将片段的视频质量映射到

**

用户感知质量[7]。

* *质量变化。两个连续片段之间的质量差异可能会导致用户感到头晕等不适，应在QoE模型中考虑。当用户下载一组瓦片时，这些瓦片的质量变化会影响用户的感知质量，在QoE模型中也应该考虑到这一点[3]。质量变化定义如下。*

*iv(k)= q0(k)-q0(k−1)+^k* (3)

*βv ≤ 1，对于c = 1，...，C (5b)*

*v=1*

*c*

σσβB+σσβB≤R L(5c)

*M*

*V*

*C*

*V*

*v v v v*

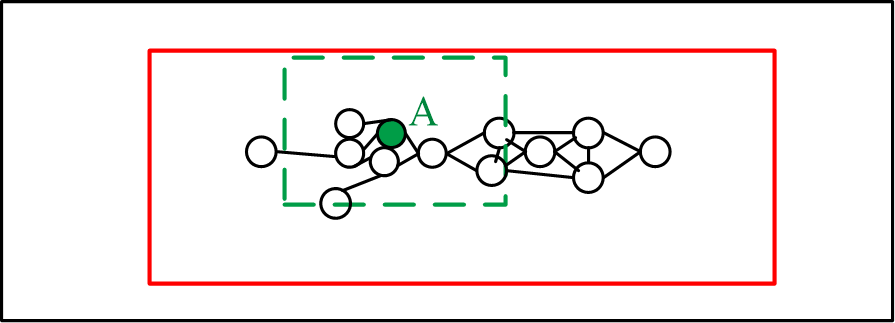
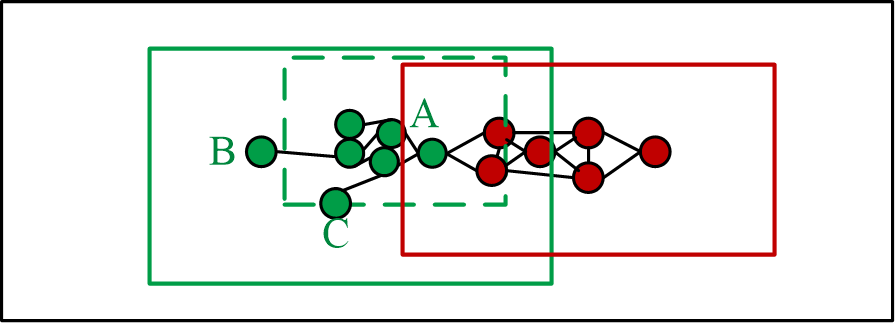
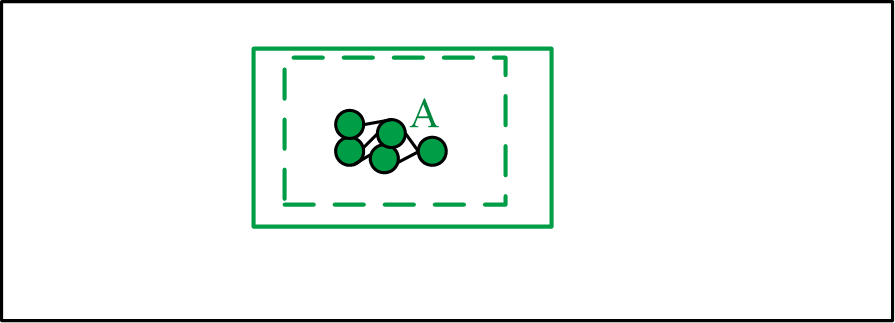
*m m c c*

*m=1 v=1 c=1 v=1*

其中Q(。)是等式中定义的QoE。1，R为网络带宽，当且仅当x > 0时，1(x) = 1，否则1(x) = 0。约束(5a)强制下载宏块或一组常规块以供查看

区域。约束条件(5b)规定，只下载一个传统图块的质量版本。类似地，仅下载宏块的一个质量版本，这可以从约束(5a)中推断出来。约束(5c)保证视频数据在回放前可以成功下载。

给定用户的观看区域，即覆盖观看区域的候选宏块(和传统块的候选集合)是已知的，我们可以在等式中分解该问题。5分成两个子问题:一个是确定宏块的正确质量级别；另一个是确定瓷砖的正确质量等级。然后，实现更好的生活质量

* 1. (二)(三)

图2:宏块结构。(a)宏块太大。(b)将一个大的宏块分割成两个宏块。(c)宏块优化(仅显示绿色)。

将是Eq的解决方案。5.如果QoE模型不考虑瓦片的质量变化，即整体QoE等于下载瓦片的平均质量水平，则后一个子问题可以简化为等式。6，其中是覆盖观看区域的一组传统瓦片。

**

*V*

σσmax*Q(β v)* (6)

*五. c*

*c∈C v=1 V*

Σ

s.t. βv = 1，对于c ∈  (6a)

*c*

*v=1*

σσβVBv≤R L(6b)

*c∈C v=1*

*c*

*c*

*V*

1. 流行感知360视频流

在本节中，我们首先描述如何构建宏块，包括两个部分:基于查看区域识别宏块和宏块优化。然后，我们提出了基于宏块思想的流行感知360视频流算法。

1. *基于查看区域识别宏块*

用户在观看相同的360°视频时，可能会有相似的观看兴趣。然后，它们可能具有相似的观看区域，并且它们的观看中心彼此靠近。为了构建宏块，我们必须首先识别这些观看中心，并将它们聚集在一起。因为集群的数量

(宏块)不是先验已知的，许多人都知道

引理1。情商的问题。6是NP难。

*证据。通过对多选择背包问题的约简，可以证明该问题是NP难的。在MCK问题中，有许多类别的项目，其中每个项目都有一个值和权重。给定一个有重量限制的背包，问题是从每个类中选择一个项目，使得总价值最大化，并且总重量不超过重量限制。*

对于MCK问题的任何实例，我们都可以在等式中构造问题的实例。6如下。我们构建一个瓷砖c作为一个类，其中这个瓷砖的质量版本(V)对应于该类的项目。对于vth版本，其质量等级v设置为vth项目的值，其带宽使用Bv设置为vth项目的权重。网络带宽限制RL被设置为背包的重量限制。

*c*

方程中这个问题的一个解决方案。6最大化瓷砖的整体质量。当质量版本被视为项目时，解决方案从每个类别中选择一个项目，以在权重约束下最大化项目的总价值。因此，该解决方案也是MCK问题的解决方案。这就完成了归约，从而完成了证明。

定理1。流行感知360视频流问题是NP难的。

*证据。Eq中的流行感知360视频流问题。5比情商里的题难多了。6、因为情商上的问题。6是Eq的一个子问题。5.基于引理1，方程中的问题。5是NP难，因此知名度感知360○视频流问题是NP难。*

由于流行度感知的360视频流问题是NP难的，我们提出了一种基于启发式的算法。

聚类算法，如k-means聚类，不能直接应用。其他非参数聚类算法，如基于密度的聚类算法

1. 在手之前不需要知道集群的数量，但是，它们可能会导致另一个问题；也就是说，集群可能会不断增长并变得过大，从而失去节省带宽的好处。例如，如图2(a)所示，由于簇的观看中心跨越大的区域，所构建的宏块(红色块)将太大。请注意，这是集群和宏块之间的关键区别。如果宏块太大，使用宏块节省带宽的好处将会丧失。为了解决这些问题，我们提出了以下聚类算法。

该算法使用两个参数λ和γ，以确保彼此靠近的观看中心聚集在一起，并且基于每个聚类构建的宏块不会太大。参数λ用于确定两个观察中心是否足够靠近以至于在一个群集中。如果两个观看中心的距离小于或等于λ，则这两个观看中心被认为是靠近的(即，两个用户观看相似的视频内容)。参数γ定义了每个簇的最大大小；即一个簇的任意两个观察中心的最大距离不超过γ。

参数λ和γ会影响聚类算法的性能。如果λ很小，可能会错过一些应该聚集的观察中心。利用大λ，彼此远离的观看中心(即，用户具有不同的观看兴趣)可以聚集在一起。γ越大，团簇越大；而对于较小的γ，可以构建更多的团簇。为了确定λ和γ，我们还需要考虑传统切片大小的影响，因为宏切片是在传统切片方案的基础上构建的，以减少带宽使用。以4x6平铺方案为例，如图1(a)所示。一些

一

x y

0.9

0.6

累积分布函数（Cumulative Distribution Function的缩写）

0.3

0

0 5 10 15

角度变化( )

* 1. 视频1

一

0.9

x y

0.6

累积分布函数（Cumulative Distribution Function的缩写）

0.3

0

0 10 20 30

角度变化( )

* 1. 视频8

表一:视频痕迹。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 身份证明 | 长度 | 内容 | 身份证明 | 长度 | 内容 |
| 一 | 4:38 | 偶像舞蹈 | 5 | 2:44 | 柯南秀 |
| 2 | 6:13 | 节日庆典 | 6 | 3:21 | 自由式滑雪 |
| 3 | 2:52 | 表演时间拳击 | 七 | 2:44 | 足球比赛 |
| 四 | 6:01 | 篮球比赛 | 8 | 4:52 | 移动的犀牛 |

x (y)坐标的偏差。图3示出了一个例子

图3:用户在x和y方向上角度变化的CDFs。由于页面限制，只显示了两个视频。

分享相似观看兴趣的用户下载两列瓷砖，其他人下载三列瓷砖；即不同之处在于一列传统瓦片。因此，我们根据经验将γ设置为传统瓷砖的宽度

集合λ = γ/4。

给定一组点(用P表示)，其中每个点

表示一个用户的观看中心的位置，我们使用Np = q q P q p dist(p，q) λ来表示靠近点P的点集，其中dist(p，q)表示两个点P和q之间的欧几里得距离。聚类

*  ∈* *∧* *∧* *≤ *

算法的工作原理如下。

1. 从具有最大封闭点数量的点开始聚类，即p = argmaxp P Np。

* ∈*

1. 通过添加接近任意的点来扩展集群

集群内部的点。扩展过程一直持续到找不到更多的接近点。

1. 检查集群中任意两点之间的最大距离是否大于γ。如果是这样，则使用k-means聚类算法将聚类分成两个聚类。
2. 从P中移除聚集点。
3. 重复步骤(a)至(d ),直到P = ∅.
4. *宏块优化*

当用户观看表中列出的八个360视频[12]时

1. 用户的头部运动数据是在两种不同的设置下收集的。当观看视频1至4时，用户被指示关注视频内容。看视频5到8时，用户可以自由探索视频；即，变化会受到视频内容和用户独特观看行为的影响。从图3中可以看出，x的变化

(y)视频片段内的坐标很小。x和y方向的变化以度数表示，可以通过乘以视频分辨率将其转换为像素。让Ax和Ay分别表示沿x和y方向的变化。然后，构建的宏块应该覆盖用户的观看区域加上其x (y)方向两侧的Ax ( Ay)边缘区域。

2 2

为了形式化宏块构造的问题，我们为每个用户I引入一个二进制变量αi，其中αi = 1，如果用户的观看区域用于构造宏块，即，

用户能够下载构建的宏块；其他方面，αi = 0，即用户下载一组传统的瓦片。宏观瓦结构的问题可以公式化

有情商。7，其中目标是当下载构建的宏块或以相同质量水平编码的一组传统块时，最小化群集中所有用户的总带宽使用。

*新泽西*

Σ

通过简单地覆盖来为每个簇构建宏块

**部**

{αi}

*αIBm+(1αI)Bc* (7)

*i=1*

聚集群集中用户的所有观看区域可能会构建不必要的大宏块。例如，如图2(b)所示，宏块(绿色块)比用户的观看区域(绿色虚线块)大得多。另一方面，由于头部运动的随机性，如果macortile太小，用户可能会在下载的macrotile之外观看视频。因此，根据群集确定正确的宏块大小非常重要。

为了确定正确的宏块大小，我们需要确定构建宏块时应该考虑哪些用户的观看区域，使得用户下载宏块时的带宽使用(命名为Bm)小于下载一组传统宏块时的带宽使用(命名为Bc)。Bm和Bc分别表示构建的宏块的数据大小和覆盖相同观看区域的传统块的数据大小。

为了解决头部运动随机性的影响，宏块应该覆盖用户的观看区域加上一些边缘区域。可以基于用户观看中心(即，x和y坐标)的变化来确定边缘区域，这些观看中心在视频流期间以固定采样率(例如，50 Hz)记录。视频片段内x (y)坐标的变化被定义为标准

其中Nj是jth集群中的用户数量。求解Eq后。7、我们可以用所有αi = 1个用户的观看区域来构建宏块。

虽然强力搜索可以保证方程的最优解。7，其计算复杂度为O(2Nj)。为了减少实践中的施工时间，我们提出了一种

迭代方法，类似于随机样本共识范式[28]。对于每次迭代，它的工作原理如下。

1. 随机选择用户观看区域的子集。
2. 对宏块进行编码，让Bm表示所构建的宏块的带宽使用情况。
3. 检查所构建的宏块是否覆盖用户I

*1, ...Nj.如果是，αI = 1；否则，αi = 0。*

σNj

1. 检查总带宽使用量(即αiBm + (1 αi)Bc)是否小于上一次迭代。如果是，用构造的宏块更新宏块

*i=1*

*−*

在当前迭代中。

如果仅下载覆盖预测观看区域的宏块，则当用户突然导航到下载的宏块之外的区域时，用户可以观看空白区域。为了解决这个问题，除了以高质量下载覆盖观看区域的瓦片(或宏瓦片)之外，用户

还会以最低质量下载剩余的切片。更具体地说，对于每个构建的宏块，如图2(c)所示，我们通过沿着构建的宏块的两个水平边缘切割视频，将剩余区域裁剪成四个部分。这四个部分以最低质量级别编码，也将与宏块一起下载。下载这四个部分所需的额外带宽非常少，因为压缩效率很高，而且这些视频的编码质量最低。

算法1:流行度感知360视频流

**输入:右、左、右**

**输出:βv或βv**

*m*

*c*

{ }

1预测用户的观看区域

2确定宏图块m和图块集

C

如果m存在，则为3

4返回选择宏块(m，r，L)

**其他5个**

6个返回选择符(，r，L)

C

**7结束**

**8**

9函数选择宏块(m，r，L):

**10伏为←1伏做**

**11如果Bv ≤ r L，则**

1. *知名度感知360视频流* 12

*β*

**13**

*m*

在最后一小节中，宏块构建在14

**结束**

**结束**

*m*

返回v

基于历史观看数据的服务器。为了支持观看区域未被构建的宏块覆盖的用户，服务器还使用常规的

国家切片方案(例如4x6)。对于视频流，客户端

15返回βv1 // v1是最低质量等级

**16结束**

*m*

17个功能选择项(C，r，L):

***要做的事***

18对于V←σV

**如果**

*c∈v*

***Bv ≤ r L则***

**19**

*c*

**20**

选择正确的图块(宏图块或一组传统图块)21

**22**

**结束**

*c*

**结束**

*c ∈ C的βc = 1*

使得QoE最大化23

*r= r lσ*

*c∈*

*Bv //剩余带宽*

在网络带宽限制下。在本小节中，我们24

* **25**

排序(C) //按照距离的升序对切片进行排序

**foreach c ∈ C do**

提出了流行度感知360视频流算法26

**如果r> =(bv+1bv ),那么**

其首先预测每个视频27的用户观看区域

段，然后预取相应的宏块or 28

*c c*

c=c∞{ c }

*r=r(bv+1 bv)*

如果需要，可以使用传统的瓷砖。

**29**

**30结束**

2

*c c*

**结束**

为了预测用户的观看区域(即，观看中心)，31

**32**

我们使用岭回归模型[29]，因为它对33具有稳健性

**如果|C | > 1 |C|那么**

*v=v+1*

*c*

*对于c ∈ C，β = 1*

**结束**

*v*

应对过度拟合。观看360°视频时，用户的历史观看中心由嵌入式传感器记录

**34**

**35结束**

返回{βc }

在该设备中，每个观看中心由它在视频帧上的(x，y)坐标表示。由于观察中心的坐标是以固定的采样率(例如，50 Hz)记录的，记录的x和y坐标可以被认为是时间序列数据。为了预测用户观看中心的x坐标，使用用户观看最近视频片段时x坐标的历史数据来训练模型。然后，给定要预取的视频片段的时间戳，训练好的模型预测用户观看该视频片段时观看中心的x坐标。类似地，预测观看中心的y坐标。基于预测的观看中心(即，x和y坐标)和FoV，预测用户的观看区域。基于预测的观看区域，该算法确定是否存在覆盖预测的观看区域加上一些边缘区域的宏块，该宏块的计算类似于上一小节中的计算。如果存在这样的宏块，则算法从最高质量级别开始搜索，直到找到宏块的最高可能质量级别，使得以该质量级别编码的宏块可以

成功下载(算法1中的第9-16行)。

如果这样的宏块不存在，将下载一组传统的块。下面显示了如何确定传统瓷砖的质量等级(算法1中的第17-35行)。该算法首先为这些瓦片找到可能的最高质量等级，使得它们可以在网络条件下被成功下载。如果剩余带宽足够大，该算法会将某些图块的质量等级提高一个等级。因为瓦片更靠近

观看中心可能对用户的感知质量有较大的影响，算法根据观看中心和图块中心之间的距离来提高图块的质量。也就是说，该算法按照瓷砖到观看中心的距离的升序对瓷砖进行排序。该算法从最靠近观看中心的图块开始搜索，然后根据剩余带宽的数量确定具有高一级质量的图块的最大数量。由于QoE受到质量变化的影响，因此只有当超过一半的下载块能够提高它们的质量水平时，才执行质量水平提高。

* 1. 绩效评估

1. *实验设置*

性能评测基于48位用户观看8个360°视频的头部运动数据轨迹[12]，视频轨迹详情见表一，这些视频涵盖了不同场景，如运动、表演、电视剧等。对于每个视频，我们随机选择四十个用户的头部运动数据轨迹来构建视频切片(和宏切片)，其余八个数据轨迹用于评估360°视频流的性能。

360视频流系统由两个主要组件组成，即服务器和客户端模块。在服务器端，类似于[3，30，31]，我们将每个视频分成一个一秒的片段序列。每个视频片段进一步在空间上被划分成小块(和宏块)。然后，使用带有编码器x264的FFmpeg [19]将所有切片(和宏切片)编码成五个质量等级(5到1，5为

一

宏块比率(%)

0.9

0.6

0.3

0

1 2 3

4 5 6 7 8

视频标识

(a)

一

0.9

1. 宏块
2. 宏块
3. 宏块

用户覆盖率(%)

0.8

0.5

0

1 2 3 4 5 6 7 8

视频标识

(b)

仅当集群至少有5个用户(即数据集中十分之一的用户)时才进行集群。

* 1. *宏块覆盖率:图4(a)显示了为视频片段构建的宏块数量。由于用户在观看相同的360°视频时有相似的观看兴趣，因此只需要有限数量的宏块。如同所示*

在图中，超过95%的视频片段只需要

图4: (a)具有不同数量宏块的视频片段。

(海湾

用户的百分比

宏块红色。

一

0.8

0.6

0.4

0.2

0

0

级别=5

级别=2

0.2 0.4 0.6 0.8

标准化带宽使用

一

级别=1

级别=4

级别=3

累积分布函数（Cumulative Distribution Function的缩写）

一

0.8

0.6

累积分布函数（Cumulative Distribution Function的缩写）

0.4

0.2

0

0 0.2 0.4

0.6

级别=5

级别=4级别=3级别=2级别=1

0.8 1

一个宏块用于视频1、2和3。视频4讲述的是一场篮球比赛，当篮球运动员移动时，用户的凝视方向也随之移动。因此，82.5%的片段有一个宏块，14.1%的片段有两个宏块，少于4%的片段有两个以上的宏块。对于视频

一

级别=5级别=4级别=3级别=2

级别=1

0.8

0.6

累积分布函数（Cumulative Distribution Function的缩写）

0.4

0.2

0

1. 视频1

一

0.8

0.6

累积分布函数（Cumulative Distribution Function的缩写）

0.4

0.2

0

标准化带宽使用

1. 视频2

在用户可以自由浏览视频的5到8个细分市场中，超过92%的细分市场平均少于三个宏块。

图4(b)示出了宏块覆盖的用户的百分比，并且大多数用户的观看区域被宏块覆盖。具体来说，94.1%、90.3%、94.6%和88.4%

级别=5级别=4级别=3级别=2

级别=1

观看视频时，宏块可以为用户提供服务

0 0.2 0.4 0.6 0.8 1

标准化带宽使用

1. 视频7

0 0.2 0.4 0.6 0.8 1

标准化带宽使用

1. 视频8

分别是1、2、3和4。即使对于用户可以自由探索的视频5到8，该值平均也超过80%

图5:宏块数据大小的CDFs。只有四个视频

由于页面限制而显示。

最高质量)使用不同的恒定速率因子(crf)值，从18到38，间隔为5 [3]。在客户端，我们将移动设备的FoV设置为水平和垂直100度。类似于[4]，我们使用过去几段的下载吞吐量的调和平均值来估计网络带宽。对于QoE模型，我们将权重设置为(ωv，ωr) = (0.25，0.25)，这是[26]中使用的设置。

基于具有不同吞吐量模式的长期演进吞吐量跟踪来模拟网络流量[32]。为了评估视频流在不同网络条件下的性能，我们将网络吞吐量跟踪线性扩展为两种类型，称为跟踪1和跟踪2，其中跟踪1的平均网络吞吐量是跟踪2的两倍。跟踪2的网络吞吐量范围为1.3 Mbps至10.9 Mbps，平均为4.8 Mbps。

我们比较了以下方法的性能。

* *拼接:使用传统的拼接方案(例如4x6)将每个视频片段分成固定大小的拼接块。Ctiling方案已被用于许多现有的流解决方案[3，14，15，16，33]。*
* *Ftiling:每个视频片段被分成固定数量的块，这些块可能有不同的大小。与[22]类似，每个片段首先被分成450个小块(即15x30切片)，然后根据用户的视图将这些小块分成10个切片。*
* *优化:除了使用4x6切片方案构建的切片之外，宏切片也使用建议的解决方案构建。*

1. *宏瓦结构的性能*

我们根据经验将γ设置为传统瓷砖的宽度，λ = γ/4。为了减少覆盖过少用户的不必要宏块的数量，我们为

视频。因此，大多数用户只需要下载这些宏块，而不是传统的块，以节省视频流期间的带宽。对于未被宏块覆盖的少数用户，他们必须在视频流期间下载传统的块。请注意，这些用户的观看中心更有可能远离宏块覆盖的区域，为了节省带宽，最好不要将它们包含在宏块中。

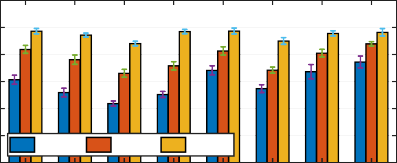
*2)宏块与常规块:用户的查看区域可以被一个宏块或一组常规块覆盖。为了了解宏块的优势，我们比较了宏块和传统块的数据大小，两者的编码质量相同。当视频片段存在多个宏块时，使用平均值。由于不同的用户可能下载不同组的传统瓦片，因此使用平均值。*

图5示出了对每个视频片段使用宏块的数据大小CDFs，基于传统块的数据大小标准化。从图中可以看出，宏块的平均数据大小比传统块小得多，因为与将视频分成小块相比，将普遍观看的视频编码为宏块可以实现高压缩效率和高覆盖效率。以视频1为例，如图5(a)所示。当视频以质量等级5、4、3、2和1编码时，使用宏块的中值数据大小分别是传统块的54%、45%、35%、29%和22%。换句话说，当用户下载以最高质量水平编码的所有视频片段的宏块时，带宽使用平均可以节省46%(即，1- 0.54=0.46)。下载质量级别为4、3、2和1的宏块时，带宽节省分别为55%、65%、71%和78%。

1. *知名度感知360的性能○视频流*

我们使用用户头部运动数据跟踪来评估流行感知360视频流的性能。Sim-

5



激活升级优化

四

QoE

3

2

一

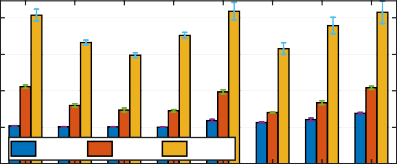
0

1 2 3 4 5 6 7 8

录像

四

3



激活升级优化

QoE

2

一

0

1 2 3 4 5 6 7 8

录像

250

200

生活质量改善(%)

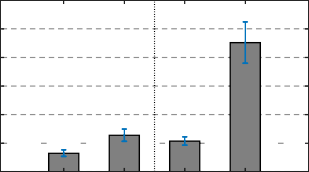
150

100

50

0

提升优化



网络跟踪1

网络跟踪2

提升优化

一

带宽使用

0.8

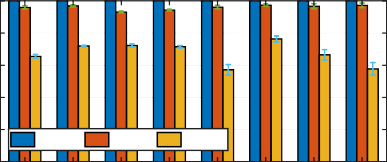
0.6

0.4

0.2

0

网络追踪1



激活升级优化

1 2 3 4 5 6 7 8

录像

1. 网络跟踪1

一

0.8

带宽使用

0.6

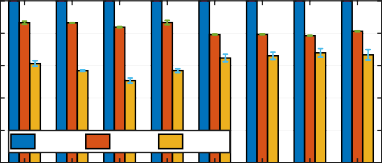
0.4

0.2

0

1. 网络跟踪2

图6: QoE对比。



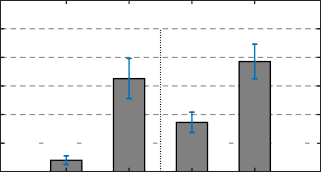
激活升级优化

1 2 3 4 5 6 7 8

录像

1. 网络跟踪2
2. 整体QoE改善

50



网络跟踪1

网络跟踪2

带宽节省(%)

40

30

20

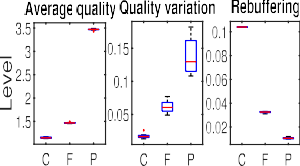
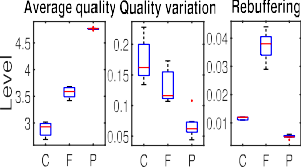
10

0

提升优化提升优化

1. 整体带宽节省

图8:带宽使用比较(使用基于Ctiling标准化)

与Ctiling相比，QoE提升高达226.1%，而Ftiling实现的QoE提升非常小。QoE的大幅提高是因为尽管Ctiling和Ftiling必须使用低质量视频，但Ptiling仍然可以下载以高得多的质量级别编码的宏块。

* 1. 网络跟踪1 (b)网络跟踪2

图7:比较QoE的三个成分(C:Ctiling，F: Ftiling，P: Ptiling)。

ilar设为[3，34，31]，播放缓冲区设为3秒。

* 1. *QoE提升:图6比较了不同网络条件下不同方法的QoE。可以看出，在所有视频跟踪方面，Ptiling在网络吞吐量跟踪方面都优于Ctiling和Ftiling。在优化中，通过使用宏块，可以实现高压缩效率和高覆盖效率。然后，在相同的网络条件下，用户可以下载以较高质量级别编码的宏块，而不是以较低质量级别编码的传统块(或由Ftiling构建的块)，因此可以提高QoE。Ftiling将用户的视图聚集到固定数量的集群中是低效的。这是因为许多用户观看的视频区域(即宏块覆盖的区域)被分成不必要数量的小块，降低了视频压缩效率。此外，分割宏块外部的视频区域(作为宏块的补偿一起编码)会降低编码效率，从而导致Ftiling的高带宽需求。当网络条件变差时(从图6(a)到图6(b))，这些方法的QoE下降；然而，优化后的生活质量比优化前和优化后下降得更慢。例如，对于视频1，与使用网络跟踪1相比，使用跟踪2时，Ctiling的QoE降低了66.1%，Ftiling的QoE降低了49.5%，但Ptiling的QoE仅降低了16.1%。*

图6(c)显示了与Ctiling相比，Ptiling和Ftiling的总体QoE改善。对于网络跟踪1，优化可以将QoE提高64.1%，大约是Ftiling (32.5%)的两倍。对于网络跟踪2，优化可以

图7比较了QoE的三个分量(如等式。1)对于视频6。优化可以获得比Ctiling和Ftiling高得多的平均质量，尤其是对于网络跟踪

2.轨迹1的优化经历较少的质量变化，其中大多数片段以最高质量下载。对于trace 2，Ptiling的质量变化很大，因为Ptiling仍然下载高质量的片段，但有时必须下载低质量的片段。至于再缓冲，Ptiling在这三种方法中表现最好。

* 1. *带宽节省:图8比较了这三种方法的带宽使用，其中带宽使用基于Ctiling的带宽使用被标准化。如图8(a)所示，对于网络轨迹1，Ptiling具有约70%的Ctiling带宽使用，而对于网络轨迹2，它具有约60%的Ctiling带宽使用，如图8(b)所示。由于其高压缩效率，优化实现了高带宽节省。回想一下，优化方法将许多用户观看的视频区域编码为宏块，并将宏块之外的其余视频区域分成四个大部分，从而实现了高压缩效率。相反，对于Ftiling来说，将视频分成大量的小块会降低视频编码效率，因此与Ptiling相比，带宽成本要高得多。*

优化和Ftiling的整体带宽节省如图8(c)所示。与Ctiling相比，Ptiling方法将网络跟踪1的带宽使用率降低了32.6%，而Ftiling的带宽使用率仅为4.1%。对于网络跟踪2，与Ctiling相比，Ptiling方法减少了38.5%的带宽使用，这比Ftiling高得多。回想一下，知名度感知360视频流的目标是最大化QoE。如图6(c)所示，与Ctil- ing相比，优化方法将QoE提高了64.1%和226.1%

一

激活升级优化

0.8

0.6

累积分布函数（Cumulative Distribution Function的缩写）

0.4

0.2

0

30 35 40 45 50 55

PSNR(分贝)

一

0.8

激活升级优化

0.6

累积分布函数（Cumulative Distribution Function的缩写）

0.4

0.2

0

30 35 40 45 50 55

PSNR(分贝)

为每个片段的每个质量水平编码对应于不同浓度方向的22个版本[21]。此外，这些方法给移动设备带来了严重的处理开销。例如，在偏置三次投影中，在视频渲染期间，从两个不同的面对边缘或角落处的纹理颜色进行采样，因此

一

0.8

0.6

累积分布函数（Cumulative Distribution Function的缩写）

0.4

0.2

0

1. PSNR(踪迹1)

一

0.8

0.6

累积分布函数（Cumulative Distribution Function的缩写）

0.4

0.2

0

1. PSNR(踪迹2)

接缝处会产生赝像。为了获得更好的视觉质量，需要硬件支持或软件技术来平滑地过滤人脸[38]。

在基于图块的流传输中，视频首先被投影到

激活Ftiling

优化

0.95 0.96 0.97 0.98 0.99 1

激活升级优化

SSIM

1. SSIM(踪迹1)

0.95 0.96 0.97 0.98 0.99 1

SSIM

1. SSIM(踪迹2)

像素密度均匀的等矩形平面，然后切割成瓦片。只有与用户预测重叠的图块

图9:视觉质量对比。

分别用于网络跟踪1和跟踪2。下载高质量编码的宏块可以显著提高QoE并减少带宽使用，这证明了优化方法的有效性。

* 1. *视觉质量:作为比较客观QoE的补充(即，五个质量等级的定义如第五-一节所述)，我们还使用以下客观质量指标来比较这三种方法的性能:峰值信噪比(PSNR) [35]和结构相似性(SSIM)指数[36]。PSNR和SSIM常用于量化压缩后图像的感知质量，其中原始未压缩图像用作参考。PSNR (SSIM)越高，感知质量越好。对于视觉质量比较，使用下载的宏块(或多个宏块)渲染实际用户的视图，并且使用无损H.264编码(即，在x264中设置crf=0)从原始视频渲染“参考”视图。在每个视频片段中逐帧评估实际用户视图和参考视图之间的PSNR和SSIM。我们使用每个片段中所有帧的平均PSNR和SSIM分别表示该片段的PSNR和SSIM。以视频1为例，如图9所示，与Ctiling和Ftiling相比，Ptiling方法实现的PSNR和SSIM要高得多。如图9(a)所示，对于网络轨迹1，中值PSNR对于优化为47.3分贝，对于傅里叶变换为43.5分贝，对于压缩为41.1分贝。图9(c)和9(d)显示了优化方法在性能上优于Ctiling和Ftiling*

SSIM的两个网络痕迹。

* 1. 相关著作

关于为360°视频流节省带宽，已经进行了大量的研究，这些研究可以分为两类:偏移投影方法和基于切片的流方法。在偏置投影中，例如偏置立方投影[2]和金字塔投影[37]，全球面被编码，更多的像素被用于用户更可能观看的特定方向。偏移投影视频的多个版本被编码，每个版本都在球体上以不同的方向集中像素。在视频流期间，下载像素集中方向与用户预测的观看方向最匹配的版本。但是，这些方法会在服务器端产生大量存储开销。例如，脸书

观看区域以高质量传送，而其他瓦片以低质量传送或者根本不传送。由于实现简单，基于切片的流方法广泛应用于360视频流。一些研究人员使用固定大小的平铺方案对视频进行编码[3，4，26，39，40]，而另一些研究人员建议将视频分成固定数量的具有不同平铺大小的平铺块[21，22]。但是，很难为固定切片方案设置恒定值(即切片数量)。在这些方案中，许多用户观看的视频区域(即宏块覆盖的区域)被分成不必要数量的小块，降低了视频压缩效率。此外，将视频区域划分在宏块之外(宏块被一起编码作为对宏块的补偿)降低了编码效率，因此导致高带宽需求。作为对传统固定平铺方案的补充，我们将普遍观看的视频内容编码为宏块，这可以同时实现高压缩效率和高覆盖效率。

* 1. 结论

本文提出了一种流行度感知的360视频流算法，以最大化网络约束下的用户体验。在所提出的解决方案中，通过考虑观看流行度来对视频进行编码，其中流行观看的区域被编码为宏块以节省带宽。为了构建宏块，我们利用了观看同一视频的用户的历史观看数据。我们首先识别这些用户的观看中心，并将它们聚集在一起，基于此我们可以识别宏块。由于很少用户的观看区域可能没有被所构建的宏块覆盖，因此也使用传统的平铺方案(即4x6)。对于视频流，客户端为每个视频片段选择具有正确质量的正确切片(宏切片或一组传统切片)，从而在带宽限制下最大化QoE。我们将流行度感知的360视频流问题公式化为一个NP难的优化问题，并提出了一种启发式算法来求解。通过真实的头部运动数据轨迹和轨迹驱动仿真，证明了该算法能够显著提高QoE，节省带宽使用。

* 1. 感谢

这项工作得到了美国国家科学基金会的部分支持。

参考

1. 谷歌。360度编码器设置下的YouTube直播。https://support.google.com/youtube/answer/6396222.
2. 脸书。兜帽下:360号楼视频。https://code.facebook.com/posts/1638767863078802.
3. F.钱、韩、肖和戈帕拉克里希南。Flare:实用视口-适用于移动设备的自适应360度视频流。在ACM MobiCom，2018年。
4. J.何，库雷希，邱立华，李建华，李富华，韩立华。Rubiks:实用的智能手机360度视频流。在ACM MobiSys，2018年。
5. 米（meter的缩写））达萨里、巴特查亚、巴尔加斯、萨胡、巴拉苏布拉-马年和达斯。使用超分辨率流式传输360度视频。在IEEE INFOCOM，2020年。
6. F.钱，纪立群，韩斌，和戈帕拉克里希南。通过蜂窝网络优化360°视频传输。2016年第五届万物细胞研讨会论文集:操作、应用和挑战。
7. 米（meter的缩写））阿尔姆奎斯特，v .阿尔姆奎斯特，v .克里希纳摩尔蒂，n .卡尔松，和
8. 渴望。360♀视频流中的预取攻击性权衡。在ACM多媒体系统会议(MMSys)，2018年。
9. B.陈，严振中，金海辉，和k .纳赫斯特。基于平铺的实时360视频流的事件驱动拼接。在ACM多媒体系统会议(MMSys)，2019年。
10. H.庞，张，王，刘，孙。走向低延迟多视点360♀交互式视频:一种多模态深度强化学习方法。在IEEE INFOCOM，2019年。
11. https://trac.ffmpeg.org/wiki/Encode/H.264.[11][http://x265.org/hevc-h265.](http://x265.org/hevc-h265)
12. C.吴，谭，王，杨。一个用于探索虚拟现实球形视频流中用户行为的数据集。在ACM多媒体系统会议(MMSys)，2017年。
13. 米（meter的缩写））肖、周正华、魏思南、刘永和陈少华。BAS-360♀:通过HTTP/2探索360度视频的空间和时间适应性。在IEEE INFOCOM，2018年。
14. 米（meter的缩写））格拉夫、蒂莫勒和穆勒。通过HTTP实现带宽高效的全向视频自适应流:设计、实现和评估。在ACM多媒体系统会议(MMSys)，2017年。
15. A.Mahzari，A. T. Nasrabadi，A. Samiei和R. Prakash。自适应360视频流的视野感知边缘缓存。在ACM Int'l Conf中。多媒体上，2018。
16. X.德·西蒙尼、西蒙和弗罗萨德。多视点全向视频的动态自适应流传输。在ACM多媒体系统会议(MMSys)，2018年。
17. R.I. da Costa Filho、M. C. Luizelli、M. T. Vega、J. van der Hooft、S. Petrangeli、T. Wauters、F. De Turck和L. P. Gaspary。预测移动网络中虚拟现实视频流的性能。在ACM多媒体系统会议(MMSys)，2018年。
18. 长度谢，张x，郭z。CLS:一个基于跨用户学习的系统，用于提高360度视频自适应流的质量。在ACM Int'l Conf中。多媒体上，2018。
19. FFmpeg。[http://www.ffmpeg.org。](http://www.ffmpeg.org/)
20. Y.包，吴洪浩，张铁生，拉姆利，刘。拍摄运动目标:基于运动预测的360度传输

学位视频。在IEEE国际会议中。关于大数据，2016年。

1. 米（meter的缩写））肖，周春华，刘玉英，陈素珊。OpTile:在360度视频流中实现最佳平铺。在ACM Int'l Conf中。多媒体上，2017。
2. C.周，肖敏，刘y。ClusTile:致力于360度视频流的带宽最小化。在IEEE INFO- COM，2018年。
3. 米（meter的缩写））肖、王树生、周立军、刘立军、李志军、刘永和陈树生。

面向高带宽效率360度视频的迷你视图布局。在ACM Int'l Conf中。多媒体上，2018。

1. X.陈，谭，曹。智能手机上的能量感知和上下文感知视频流。在IEEE国际会议中。关于分布式计算系统(ICDCS)，2019年。
2. Y.杨，胡伟，陈晓，曹。面向移动视频流的能量感知中央处理器频率缩放。IEEE Trans。关于移动计算，2019年11月。
3. Y.张、赵萍、卞国军、刘永年、宋立军、李晓军。DRL360:深度强化学习的360度视频流。在IEEE INFOCOM，2019年。
4. E.舒伯特、桑德、埃斯特、克里格尔和徐。DBSCAN重访，重访:为什么以及如何(仍然)使用DBSCAN。ACM Trans。关于数据库系统，2017年。
5. K.g .德帕尼斯。RANSAC算法概述。技术报告，约克大学，2010年。
6. 岭回归。https://towardsdatascience.com/ridge-回归更好的使用-2f19b3a202db。
7. X.陈、谭天成、曹国伟、拉波塔。智能手机上的上下文感知和能量感知视频流。IEEE Trans。出现在移动计算上。
8. Y.关，郑振中，张，郭，姜。Pano:通过更好地理解质量感知来优化360视频流。在ACM SIGCOMM，2019年。
9. J.范德胡夫特，彼得安吉利，瓦特尔斯，于塞根斯，

页（page的缩写）朗道·阿尔法斯、t·博斯腾和f·德·图尔克。基于HTTP/2的HEVC视频在4G/LTE网络上的自适应流式传输。IEEE通信快报，2016年。

1. J.陈，胡，罗，王，吴。SR360:以超分辨率提升360色视频流。在2020年关于网络和操作系统支持数字音频和视频(NOSSDAV)的ACM研讨会上。
2. 长度谢，许，班，张，郭。360ProbDASH:使用基于切片的HTTP自适应流来提高360视频流的QoE。在ACM MM，2017年。
3. PSNR，峰值信噪比。https://en.wikipedia.org/wiki/Peak信噪比。
4. SSIM，结构相似性指数。https://en.wikipedia.org/wiki/Structural相似性。
5. 脸书。面向360°视频和虚拟现实的下一代视频编码技术。https://code.fb.com/virtual-reality.
6. ARM。白皮书:360度视频渲染。https://community.arm.com/developer/tools-软件/图形/b/博客/帖子/白皮书-360度视频渲染。
7. 长度孙，毛永年，宗，刘永年，王永年。基于群集的360度视频直播。在2020年ACM多媒体系统会议(MMSys)上。
8. Y.张，关永年，卞国军，刘华佗，宋，和

X.李。EPASS360:移动设备上的QoE感知360度视频流。IEEE Trans。关于移动计算，2020年。