视域定向扩展，不规则视域

我们优化传输视频质量和时延。

云渲染

远程渲染所产生的长时延，极易产生传输相对体验的滞后，渲染一个比视口更大的区域对用户而言是有必要且有益的。

视口的大小容易考虑为，首先与延迟有关，在固有带宽条件下，更大的延迟需要匹配更精确的视口，以降低所要传输的数据量。

其次与用户交互性有关，更快的视野移动需要更宽泛的视口来抵消视野移动性、修正视野偏离，从而避免黑边问题。

通常的解决方案以视野落点为中心，向用户提供内容服务，如文献工作X，仅传输视野范围内的区域内容；再比如文献工作X，将所传输区域向外扩展一圈，并将扩展区域称之为XX；此外，也有部分工作采用整体传输方案，将视野区域内高质量传输，视野区域外低质量传输，来避免出现黑边问题。

其中存在几个问题和挑战。

1. 视野判知问题。如何提前判断用户未来可能的视野区域。完全准确的判知非常困难，甚至不可能做到。当采用部分传输方案时，能够降低所需传输的数据量，但当预测判知失败的情况下，黑边非常容易出现，这会严重影响用户体验
2. 视域扩展问题。如何围绕视野变化更有效率、灵活地扩展视域。在视野快速移动的情况下，向四周扩展需要成倍地增加分片数量，这其中的数据量随扩展区域的增加呈现线性增长的趋势。

此外还有许多被忽略的点，比如头部移动性对视域展示的影响，乃至对用户体验的影响。

头部不同的行为模式，包括头部旋转频率、角速度等，以识别用户特性。

因此，定向扩展和基于视野的内容判知是一个非常重要的关键问题和亟待解决的重要挑战。

在本文中，我们设计了视野预测驱动/赋能的视域离散化定向/动态扩展策略，以分片为单位增加视域区域，探索视域变化的边际效应，实现传输数据量、时延和用户体验之间的均衡。具体来说，相关的贡献被总结如下：

1. 采用场景时空特性提取的预测方案提前判知观看者头部运动，来降低XXX，补偿实时渲染延迟。
2. 渲染一个更大的视域，以便在用户头部移动过程中有足以补偿运动延迟的、可旋转的余量。其所需要的视口大小，应当刚好够补偿，为了确定这个大小，基于用户数据导出了**视口大小和时延的关系**。

**数据部分：**

**文献1所说参与者不同视频下不同的时间段下，参与者头部移动的频率不同，**

**我们进一步探究了**

**约有99%的概率旋转角度在100°以内。**

**Segment为2s时，头部移动的约束条件。**

**区分不同的类型，游荡型的，和移动型的。**

**移动型的，可以划分为，高速移动和低速移动。**

**画面是以segment为单位传输的，即使存在黑边，还会重传吗？**

**游荡型的，视域是向四周扩展，不规则圆形。**

**移动型的，视域向移动方向扩展，呈现杠铃型。根据移动速度大小，杠铃的杠有粗有细。**

## Section III

**首先，我们综合了目前领域内收集的多个视野开源数据集，包括XXX、XXX等。各数据集的特点如表X所示：**

1. **参与人数量 2. 视频数量 3. 视频特征（帧率/采样率） 4. 坐标类型 5. 视频持续时间 6. 视频比特率大小 7. 视频分辨率**

**在这一章节，我们围绕所收集的数据集，分析观看者行为的视野移动特点。**

**根据参考文献X对分片大小的测试，分片不应当过大以支持频繁的画面切换。同时，分片也不应当过小造成超额的开销。因此在本文中以segment为2s为标准，将360°环绕环面统一划分为24\*16的紧密分片，并观察视野移动特性。**

**分析内容：**

1. **视野移动角速度**

**移动角速度的CDF。这里的角速度采用segment持续时间内的spherical Haversine distance。**

**（1）具体可以细分为，俯仰角的角速度和转角的角速度等。**

1. **分片被关注的概率同样满足28分布。？根据画面大小来看，约有12-20%的场景内容可以呈现到视野范围内。由于用户在segment持续时间内也存在视野移动，数据显示仍有X%的分片在观看过程中被访问。**
2. **专注时间。用户的视野落点通常是动态的。我们将视野落点维持在同一个分片内的状态称为专注状态（即视野移动速度小于15°）。观察用户专注的持续时间分布**。**专注时间与15°内的移动速度相对应，有X%的概率会持续多久。**
3. **视野转移/切换频率？**
4. **视域覆盖范围？**

**综上，我们可以合理推断得到如下结论：**

1. **不同视频、不同观看者所展现出的观看兴趣与交互行为是不同的。相比垂直移动，观看者更习惯于在水平上转换角度。视野的预测应当综合考虑视频特征及观看者行为。**
2. **抛开视频内容与观看者行为习惯，遵循28定律的视野移动行为本身仍有迹可循。与之相对应的，场景画面有潜力被针对性渲染。**

### Section IV 视野预测

**视野预测模式：AI驱动为主，规则性策略为辅的预测框架。**

### Section IV 视域扩展

**通过所测试的实验，我们可以得到一个统计学上的数据，即2s内，人们在观看过程中的视野变化以100%的概率小于100°，这在本文中被视为最大可被下载的分片范围，也约束了视域扩张的极限。**

**因此，在本文中，一个场景不再以视野落点自身来定义，而是以视野在前后移动过程中的两点和视野差来定义。**

**为此，依据视野移动速度，我们将视野转变的行为划分为不同的模式，包括：专注型（segment持续时间内移动15°以内），环视型（15°-40°）和扫视型（40°-100°）。**

**算法在运动方向上扩张视域和视野边距。**

**首先，考虑视域扩展的效用问题。视域的大小在本文中与用户所观看范围的分片紧密相关，所决策对象即为所需传输的分片个数及哪些分片应当被及时地传输（未被及时传输的将会通过请求更新画面，这将消耗更多的带宽资源并招致更大的时延水平）。为了简化处理，本文中将用户视野覆盖分片的区域统一归为视域内，即，即便画面未全部覆盖，也被统一地计入视域。为此，所设计效用函数如下：**

**F(S) = f(S)+αg(S)+βR**

**采用线性组合模型来捕捉视域的效用。第一类XXX；第二类XXX；第三类XXX。**

**max**

****

**约束条件：**

1. **视域限制：小于100°，视域内的分片数量小于X个。大于XX，至少需要满足一个FoV大小，不然肯定需要重新请求和重传。?还是说预测效率低，留到后面再传更合适？**
2. **时延限制：小于Xms，应用所期望能够完成内容的最终时限。**

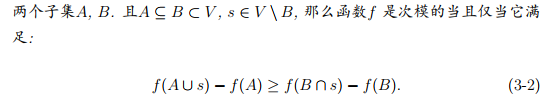
**其中，f()表示带宽消耗，g()表示时延水平。前者表征传输对网络的压力，时延则表示响应的快慢。最后，R则与用户体验息息相关。**

**可以注意到，S为离散的分片集合，该问题是个集合的离散优化问题。为此我们考虑集函数的特性与次模优化策略。下面给出次模函数的定义：**

**Definition 1 次模函数。对于有限集合T和定义在其幂2T上的一个实函数F：2T➡R，f是次模函数，当且仅当对于T的任意两个子集X subset T和Y subset T，满足如下条件：**

**F(X)+F(Y)≥F(X∪Y)+F(Y∩X)**

**其中X，Y为任意的分片集合。F()为该视域的效用函数。集函数F()同样满足收益递减的属性，**



**这表明，在分片集合中增加元素s，次模函数增益永远不会随着集合的变大而增加。下面，给出证明。**

**......**

**因此该函数是个次模函数。**

**证毕。**

**证明其单调性。**

**......**

**因此该次模函数是单调非递减的。**

**次模函数的非负线性加权组合也是次模的。**

**证毕。**

**其具有次模性、凸性。具有收益递减的特质，从画面中选择的分片越多，那么额外增加的分片所带来的重要信息就会越来越少。**

**因此该问题是个带约束的次模优化问题。**

**子模性是一个广泛有用的概念。**

**解决方案：贪婪策略。**

**对于最优化问题，解决方法通常有两种, 精确求解算法和近似求解算法. 精确求解算法通常用来处理一些简单问题。而近似求解算法通常用来解决一些难问题。**

**对于近似求解算法, 衡量其优劣性的指标,大体上有两个, 分别为近似比和求解效率. 近似比, 又称性能比, 定义为算法所得解对应的目标函数值与最优解对应的目标函数值的比值. 求解效率是探讨所设计算法是否可以在多项式时间内终止.**

**Definition 近似比/性能比，对于任意的最优化问题**

**贪婪策略可以很好地利用次模函数边际效益递减的性质。**

**加入streaming？进一步强化，数据流形式的次模优化**

**次模性与贪婪算法结合就很厉害，执行速度快，还有很好的近似比。**

围绕

Viewport/FoV Margins

移动范围（分片），移动速度（角速度）。

当目标函数有子模性，可以利用贪心算法能在多项式时间内以常数因子逼近问题最优解。目前已被应用在传感器放置、文本摘要、视频分割、数据子集选择等领域

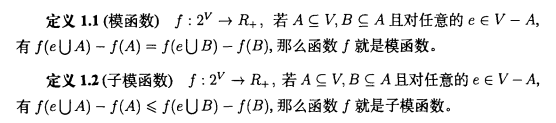
1. 提出一个目标函数，并证明其子模性(submodularity)

子模性定义

子模函数的非负线性组合仍然是子模的。

1. 设计XXX贪婪算法，实现子模函数(submodular function)最大化。
2. 使用XXX真实数据，并在原型系统中进行测试，验证算法的有效性。

定义一个有限集合V，2V是V中元素组成的子集集合，集合函数为f：2V→R。





子模函数具有效益递减的属性。在 S 中增加一个元素所增加的收益要小于等于在 S 的子集中增加一个元素所增加的收益。边际效应递减，这一点与凸的性质很类似。

从寻优角度看， 如果目标函数具有子模性， 则存在一个简单的贪心算法能在多项式时间内以常数因子( 1－1 /e) 逼近问题的最优解。

引入实际问题，并将其转化为具有XXX约束的子模函数最大化或最小化问题。

该问题往往是NP难问题，可扩展性往往收到限制，一般只有理论近似的解决方案，贪婪策略。

广义贪婪算法、单位成本贪婪算法。

子模函数脱胎于经济学、博弈论、组合优化合运筹学，在处理覆盖优化问题上过程简单、结果良好。

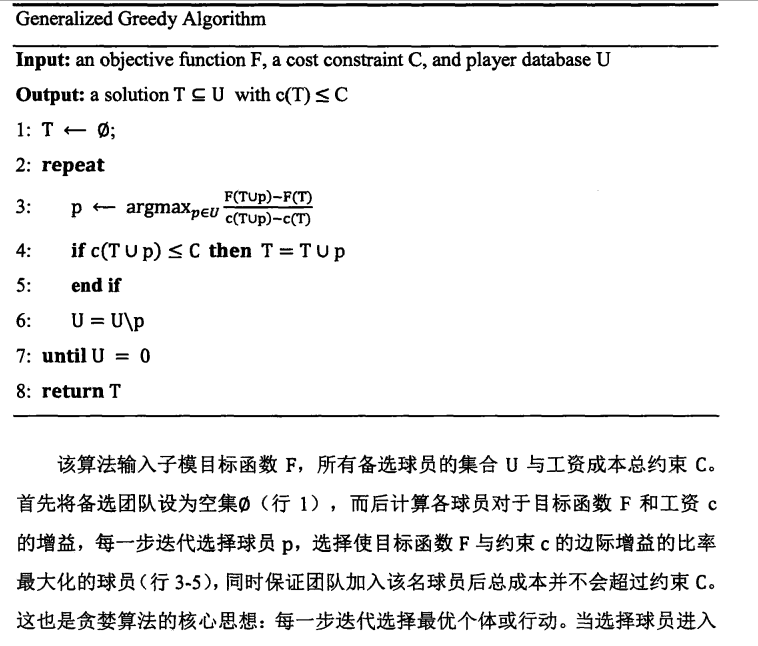
自适应子模，在无噪声情况下的主动学习。

Matlab工具箱，子模函数优化工具 SFO Submodular Function Optimization.

开放性问题：处理更一般情况的约束类问题，在线相关问题和自适应扩展问题。

大规模子模函数最小化方法，约束最小化的近似算法。

**广义贪婪算法**



**约束贪婪算法**

适用于团队组成人员数量固定，约束函数线性、离散的情况。