



北京航空航天大学
BEIHANG UNIVERSITY

Research Progress

Weekly Progress and Reports

李奕君 @ 2019-04-12

虚拟现实技术与系统国家重点实验室
STATE KEY LABORATORY OF VIRTUAL REALITY TECHNOLOGY AND SYSTEMS

1. Main Tasks

2. Weekly Progress

3. Main Problems and Solutions

4. Next Weeks

Main Tasks (1/1)

- 负责，三维场景人体与对象交互的接触点标注工具 [2019/6]
- 参与，SA2019, [20190501投稿]
Human & Bicycle Co-modeling from Single Image in the Wild

Weekly Progress: 2019.04.08 ~ 2019.04.12

● 本周工作进展

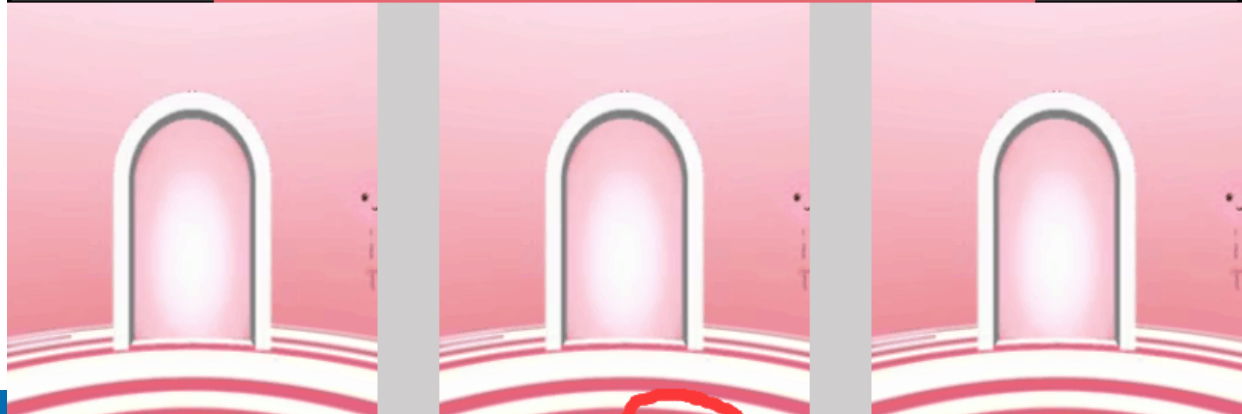
- 制作了粗糙的全景/VR视频兴趣点标注工具
- 设计了根据手工标注兴趣点驱使3个相机运动的dp算法

Weekly Progress: 2019.04.08 ~ 2019.04.12

● 标注:

- 用户在等距柱状投影格式的全景视频上标注兴趣点
- 鼠标点击一下代表这一帧的该像素点是感兴趣的（兴趣点一直存在直到被用户删除）
- ClearP按钮清空当前帧所有的兴趣点
- ExportP按钮导出兴趣点
- ImportP按钮导入兴趣点

Video360

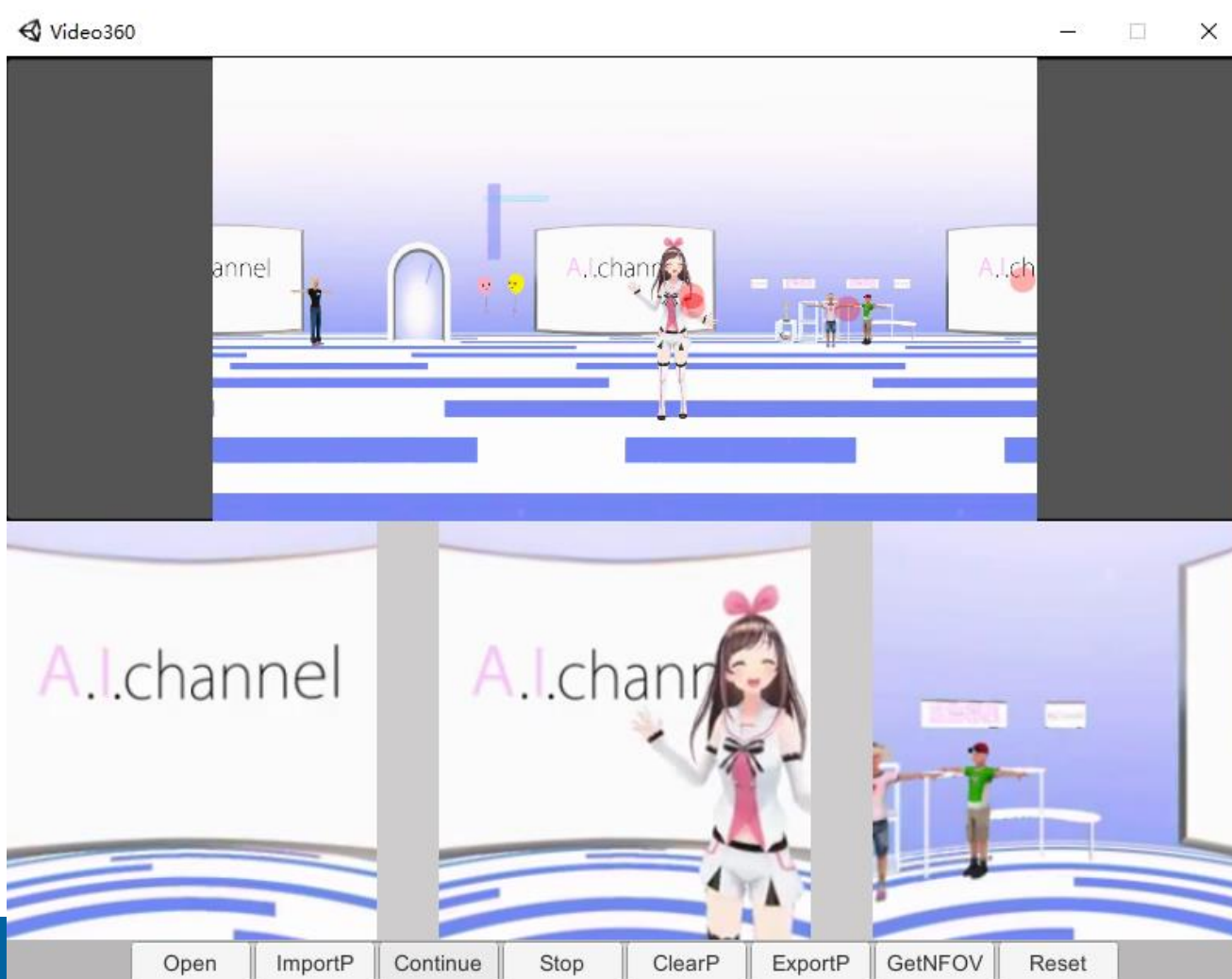


Open ImportP Continue Stop ClearP ExportP GetNFOV Reset

Weekly Progress: 2019.04.08 ~ 2019.04.12

● 生成NFOV视频:

- 标注完后点击GetNFOV按钮对三个视野为60° 的相机的运动进行规划



Weekly Progress: 2019.04.08 ~ 2019.04.12

● 算法设计(Deprecated):

- 相机视野大小 60°
- 将球面分成18份:
俯仰角3份($-60^\circ, 0^\circ, 60^\circ$) X 方位角6份($0^\circ, 60^\circ, 120^\circ, 180^\circ, 240^\circ, 300^\circ$)
- 状态设计:
- $F[t,i,j,k]$ 表示第 t 个时间, 相机1、2、3分别对准视角 i 、 j 、 k 所获得的最大收益
- $i \neq j, j \neq k, k \neq i$ (不能同时占用一个视窗)
- 其中只考虑兴趣点发生改变的时间点, 并且相邻时间点的间距较大(即对于相邻两个时间点 t_1 、 t_2 , 相机可以选择不移动也可以选择从 t_1 时刻开始向其它视角移动, 并在 t_2 时刻到达目标视角, $t_1 < t < t_2$)

Weekly Progress: 2019.04.08 ~ 2019.04.12

● 算法设计(Deprecated) :

- 每个相机单独考虑代价
- 设前一个时间为 $preT$,当前时间为 $nowT$,该相机要从视角 ou 转移到 u
- 令 $T=nowT-preT$, t 为转移花费时间,
- $degreeCostPerSec$ 为每秒移动 $degreeCostPerSec$ 度对应的代价为1
- $ATAC(ou,u)$ 为从视角 ou 移动到 u 的距离代价
- 这里取 $ATAC(ou,u)=\text{欧几里得距离}(ou,u)=\sqrt{(ou.x-u.x)^2+(ou.y-u.y)^2}$
- $score[u]$ 表示视角 u 对应视窗的价值, 代码里等于兴趣点的个数
- $transferValue=(T-t)*score[u]-ATAC(ou,u)/t/degreeCostPerSec$
- 最大化 $transferValue$, 则 $t=\sqrt{ATAC(ou,u)/score[u]/degreeCostPerSec}$
- 令 $tvu=transferValue(preT,nowT,u,ou)$, $tvv=transferValue(preT,nowT,v,ov)\dots$
- $f[nowT,u,v,w]=\max(f[preT,ou,ov,ow]+tvu+tvv+tvw)$

● 算法缺陷(Deprecated) :

- 若 $\text{nowT} - \text{preT}$ 过小将不会影响相机的运动
- nowT 的状态无法从时间小于 preT 的状态转过来, 即如果 $\text{oldT} < \text{preT}$, 存在 $f[\text{oldT}, \text{ou}, \text{ov}, \text{ow}] \rightarrow f[\text{nowT}, \text{u}, \text{v}, \text{w}]$ 优于 $f[\text{oldT}, \text{ou}, \text{ov}, \text{ow}] \rightarrow f[\text{preT}, \text{i}, \text{j}, \text{k}] \rightarrow f[\text{nowT}, \text{u}, \text{v}, \text{w}]$ 的情况
- 复杂度为 $18^6 \times \text{兴趣点变化的次数}$

Weekly Progress: 2019.04.08 ~ 2019.04.12

● 改进算法 NEW

- 相机视野大小 60°
- 将球面分成60份:
 - 俯仰角5份(-60° , -30° , 0° , 30° , 60°)
 - 方位角12份(0° , 30° , 60° , ..., 330°)
- 状态设计:
- $f[t,i,j,k]$ 表示第 t 个时间, 相机1、2、3分别对准视角 i 、 j 、 k 所获得的最大收益
- 每 timeGap 秒为一单位时间
- 对每一单位时间的每一个格子预先处理好兴趣分数分布存为分数图 score
- $\text{score}[i][j]$ 表示第 i 个时间状态 j 对应的分数

Weekly Progress: 2019.04.08 ~ 2019.04.12

● 改进算法:

- 令 **ATAC**(ou,u) 为从视角ou移动到u的距离代价(AngleToAngleCost)
- 这里取 **ATAC**(ou,u)=欧几里得距离(ou,u)= $\sqrt{(\text{ou.x}-\text{u.x})^2+(\text{ou.y}-\text{u.y})^2}$
- 令 **ODPS**(optimalDegreePerSec) 为每秒最佳运动度数、
CMPC(cameraMotionPenaltyCoefficient) 为相机运动一次的固定代价
- 令 **transferValue**(ou,u,t) 为相机从视角ou到视角u的价值
- 若 $\text{ou}=\text{u}$, 则 **transferValue**=score[t][u]
- 若 $\text{ou}\neq\text{u}$
 - 则 $\text{transferValue}=-\text{abs}(\text{ATAC}(\text{ou},\text{u})/\text{timeGap}-\text{ODPS})/\text{ODPS}*\text{timeGap}-\text{CMPC}$
- 令 **FOC**(u,v) 为指向视角u和v的两个相机的重叠代价(FieldOfViewOverlapCost)
- 令 **FOPC** 为重叠面积的影响系数
- $\text{FOC}(\text{u},\text{v})=\text{u和v重叠面积}/\text{相机视野面积}*\text{timeGap}*\text{FOPC}$

Weekly Progress: 2019.04.08 ~ 2019.04.12

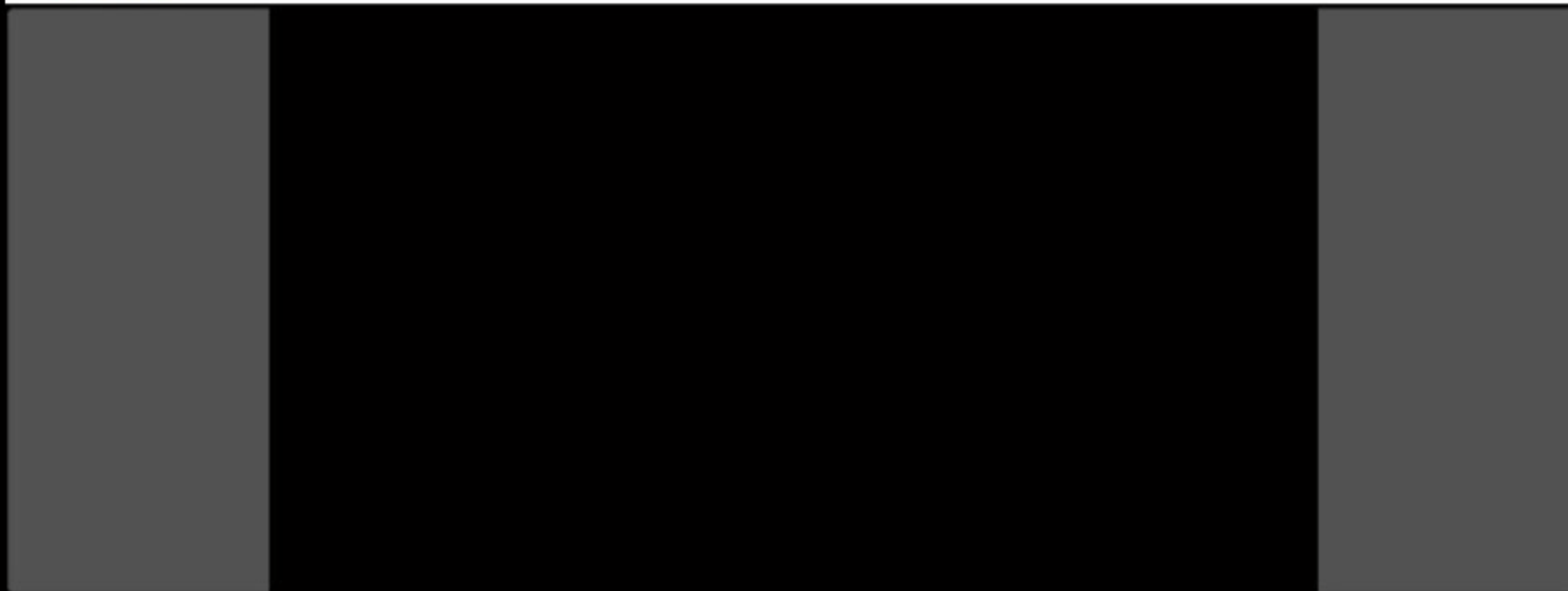
● 改进算法:

- 令 $\text{FOC}(u,v,w) = \text{FOC}(u,v) + \text{FOC}(v,w) + \text{FOC}(w,u)$
- 设 $\text{FOC}(u,v,w,ou,ov,ow)$ 为从 $f[t-1,ou,ov,ow]$ 到 $f[t,u,v,w]$ 的重叠区域代价
- 若 $ou == u$ 、 $ov == v$ 、 $ow == w$
- 则返回 $\text{FOC}(u,v,w)$
- 若 $ou == u$ 、 $ov == v$ 、 $ow != w$
- 则返回 $\text{FOC}(u,v)$
- ...
- 其余情况返回 0

Weekly Progress: 2019.04.08 ~ 2019.04.12

● 改进算法:

- 令
- $tvu = \text{transferValue}(ou, u, t)$
- $tvv = \text{transferValue}(ov, v, t)$
- $tvw = \text{transferValue}(ow, w, t)$
- $f[t, u, v, w] = \max\{f[t-1, ou, ov, ow] + tvu + tvv + tvw - \text{FOC}(u, v, w, ou, ov, ow)\}$
- 对于最终时刻**endT**找到得分最大的状态倒推找出相机路径
- C#上使用**并行计算**对枚举每个时间求状态的循环体优化
- 在4分20秒的VR视频上求解花费时间约20分钟
- Video...



Open

ImportP

Pause

Stop

ClearP

ExportP

GetNFOV

Reset

timeCost: 1166

Weekly Progress: 2019.04.08 ~ 2019.04.12

● 存在问题:

- 仰角设定范围为 -60° 到 60° , 无法很好地将球顶画面居中显示
- 计算复杂度跟相机个数成指数级 , 无法很好地扩展到更多相机(贪心地去除价值小的状态?)
- 三个相机没有特定的任务, 没有区分度(如何给相机分配角色)
- 相机从一个角度转移到另一个角度的代价应用球面距离公式计算?
- 单纯地使总的覆盖兴趣点最大可能造成一个相机无法持续捕捉一个物体的运动
- 多个相机对比单个相机优势在哪?
- 在控制相机旋转角度的基础上是否能放大缩小视野使感兴趣内容更加突出?

Next Weeks

● 下周的工作计划

- 1.继续完善相机运动算法
- 2.利用现有算法提取视频感兴趣的内容特征
- 3.将相机取景窗口显示在全景视频界面上



北京航空航天大学
BEIHANG UNIVERSITY

THANKS!

虚拟现实技术与系统国家重点实验室
STATE KEY LABORATORY OF VIRTUAL REALITY TECHNOLOGY AND SYSTEMS