



# 如何提供更好的VR体验

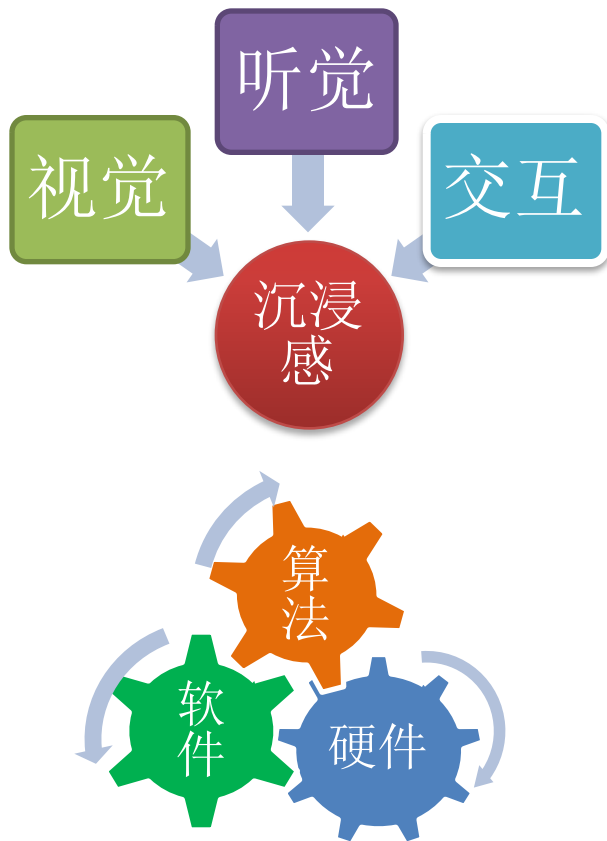
王西颖  
2017.4



悦 享 品 质

# 0.影响VR体验的关键因素

- 延迟
- 光学与显示效果
- 图像视频播放
- 交互
- 音效
- 位置跟踪
- AI



# 1. 降低MTP延迟

## 1. Motion-To-Photon 延时

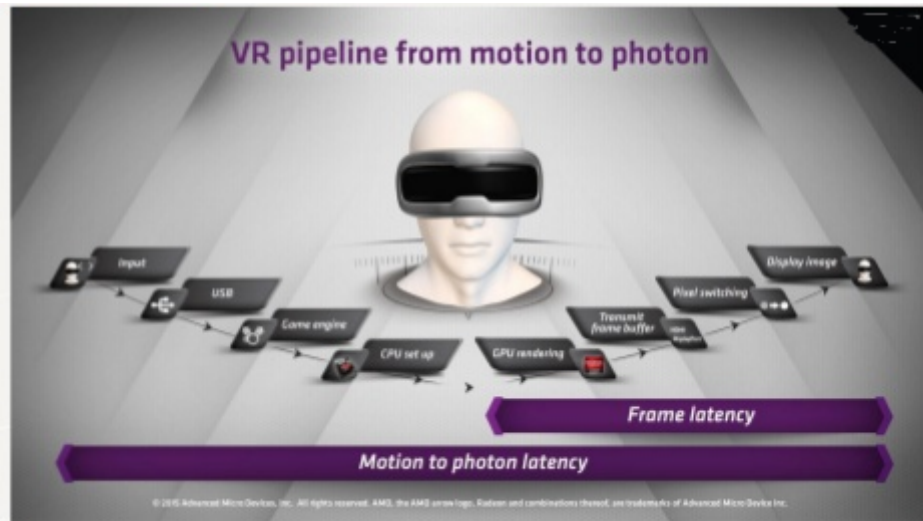
- 传感器延时
- 图像渲染延时
- 安卓系统的显示延时
- 屏幕显示延时

## 2. 传感器延时的优化

## 3. 图像渲染延时的优化

## 4. 安卓系统显示优化

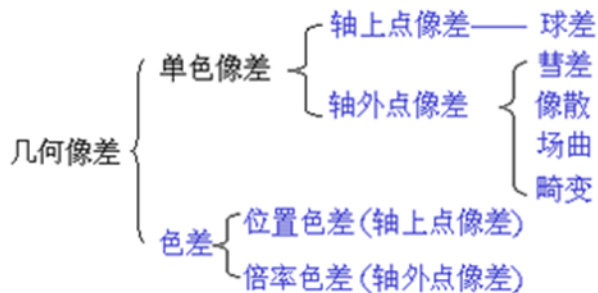
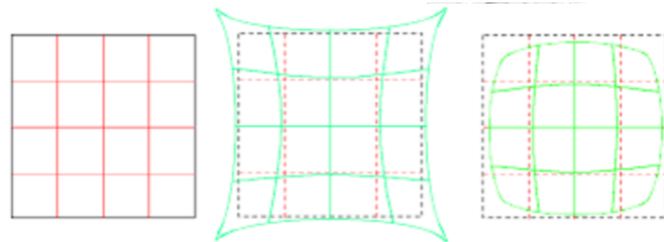
## 5. 屏幕显示优化



## 2. 提高显示效果

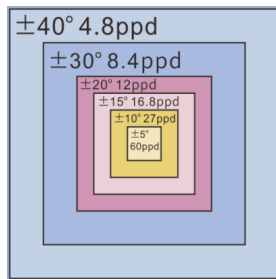
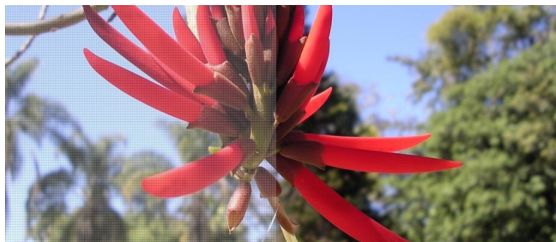
1. 屏幕
2. 光学畸变
3. 色散
4. 光学设计相关

- FOV
- 清晰度
- 畸变导致几何形状的变化，而像差影响了成像的清晰度



## 2.1 屏幕

- 高分辨率 2K/4K/8K
- 高刷新率  
60/75/90/120Hz
- 高PPI/PPD 30
- 低显示延时,降低拖影
- 单屏or双屏

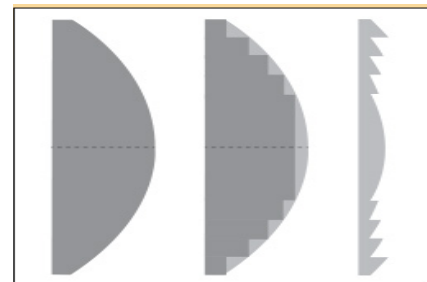


人眼视觉

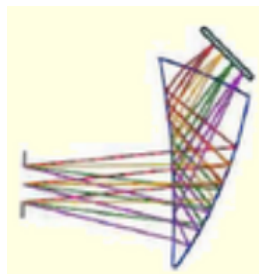
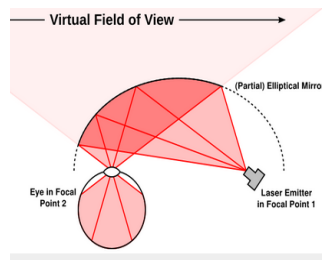
Product	Horizontal pixels per eye	Horizontal Field of View (degrees per eye)	Pixel Density (pixels/degree)
Oculus DK1	640	90	7.1
OSVR HDK	960	90	10.7
HTC VIVE	1080	90	12.0
Sensics dSight	1920	95	20.2
Sensics zSight	1280	48	26.6
Sensics zSight 1920	1920	60	32.0
Human fovea			60.0

## 2.2 光学成像

- 非球面与菲涅尔
  - 非球面与球面比较，边缘像差更小
  - 非球面镜片比球面更薄
  - 菲涅尔的尺寸厚度大幅降低，减重明显；但问题是带来了杂散光，在明暗对比度大的场景下更加明显

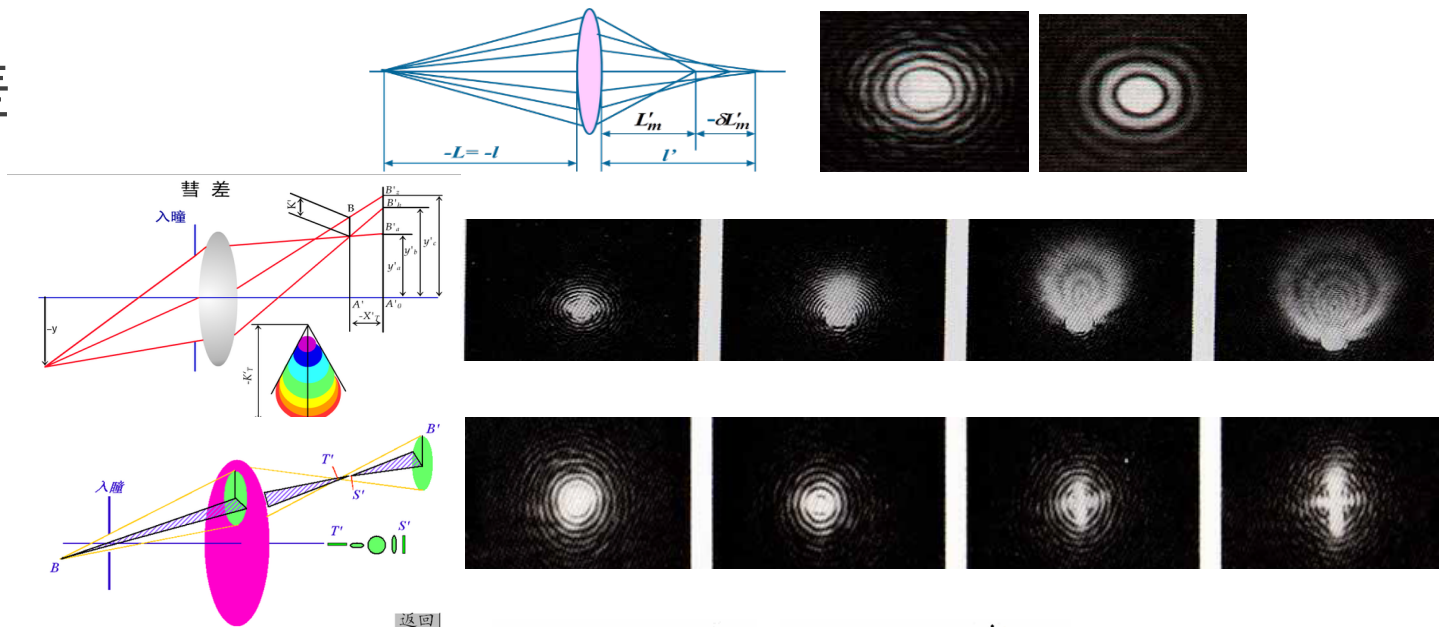


- 其他形式的Near-Eye-Display成像方式

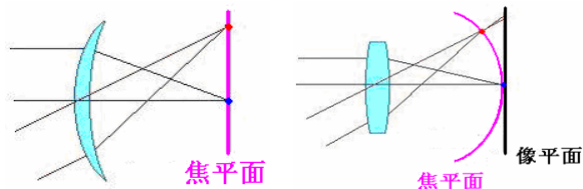


# 2.3 成像像差

- 轴上球差
- 慧差
- 像散
- 场曲



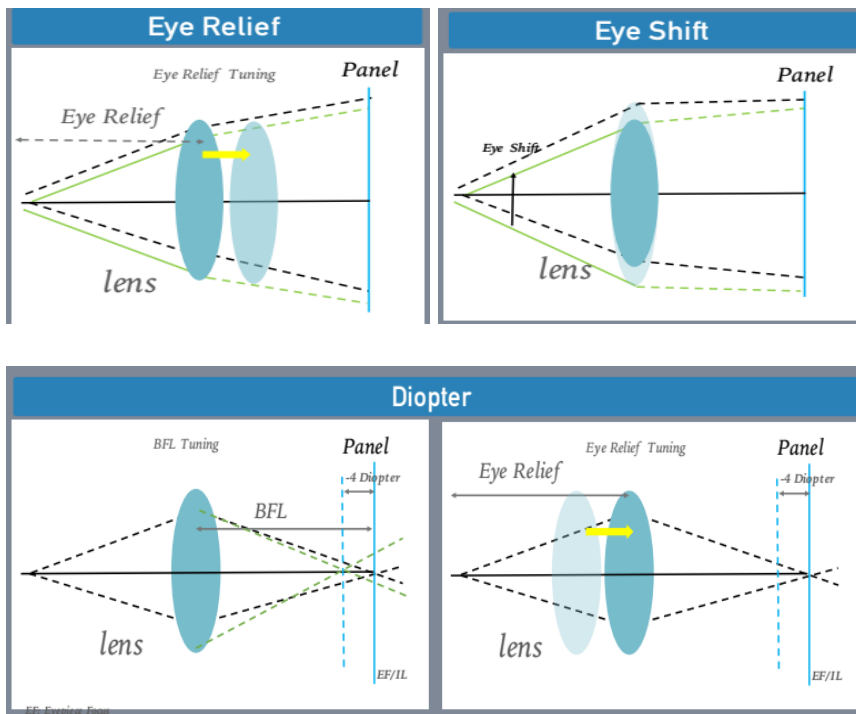
返回



场曲

## 2.4 光学设计

- 近视与物距调节
- 瞳距调节与自适应
- FOV多大合适



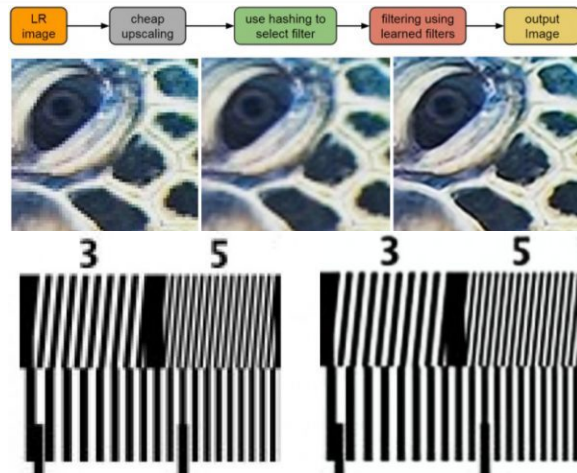


### 3. 视频类应用的技术问题

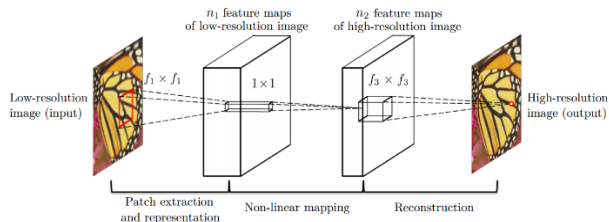
- VR中视频种类的自动分类
  - 分辨率VGA/720P/1080P/2K/4K
  - 普通or全景
  - 全景：180度/360度
  - 2D or 3D
  - 3D：左右/上下
- 纱窗效应，观影模糊，摩尔条纹（moiré pattern）
  - 图像/视频的超分辨率LR- > HR
  - 图像deblur等后处理技术

# 3.1 图像超分辨率

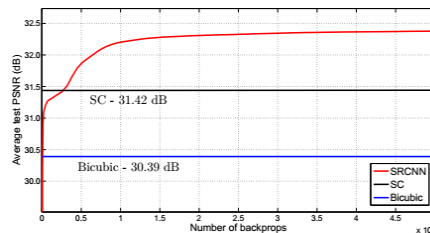
- 估计和恢复已经丢失的高频信息: 边缘, 纹理等
- 传统线性插值: 最近邻NN, Bilinear, 双三次bicubic
- 基于稀疏编码的非线性学习方法
  - 构建LR和HR dictionary
- 基于深度学习的方法
  - 以SRCNN, Google-RAISR等技术为代表
  - End-to-end 不需要中间过程, 效果好, 速度快, 可运行在移动设备上
  - 解决高频部分在低分辨率显示时产生moiré pattern, 避免aliasing artifacts



RAISR的结果与传统方法比较

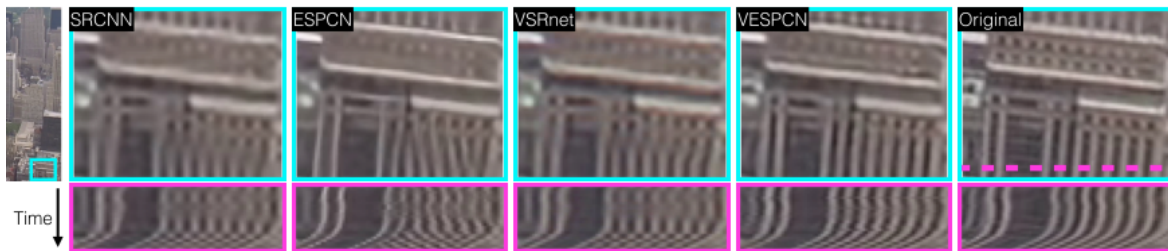


SRCNN与sparse-coding, bicubic比较PSNR的结果

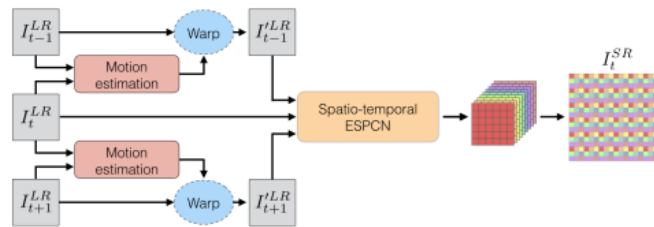


# 3.1 视频超分辨率

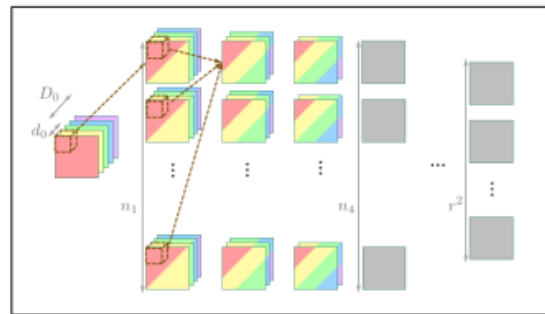
- 同时利用空域和时域信息进行细节恢复
- 基于深度学习的方法
  - Twitter的时空子像素CNN网络方法 (VESPCN)
  - 前提假设: 相同场景的后帧可以通过前帧和某种运动模型得到, 即时空连续性
  - 基础是ESPCN(efficient sub-pixel convolution NN)
  - 帧间运动补偿: multi-scale spatial transformer NN



VESPCN (9 layers,3 frames) 与其他方法的比较



Twitter视频增强系统: spatial-temporal network 以及motion compensation



3D Convolution

# 4. 自然的交互

1. 手柄

2. 手势

- 裸手手势
- 手柄手势
- 手套

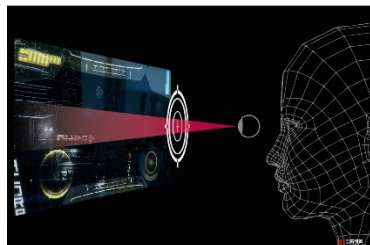


3. 语音

4. 视线 (gaze tracking)

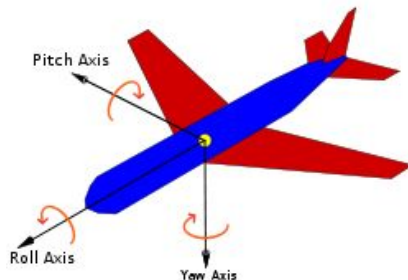
5. 表情

6. 身体动作 (body motion capture)



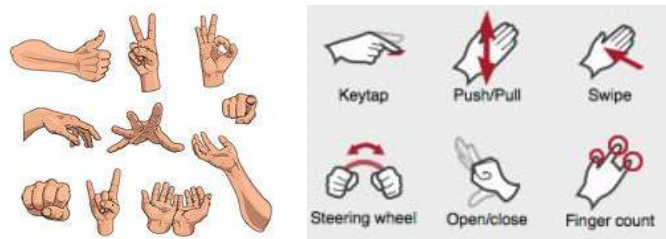
# 4.1 手柄

- 传统的手柄→带IMU的运动感知手柄
- 六轴和九轴传感器
  - 陀螺仪，加速度计，地磁传感器(电子罗盘)；
  - 陀螺仪→ 各轴向的旋转角速率，通过角速率的时间积分来估计旋转角度；加速度计→ 各轴向的加速度
  - **六轴融合**：陀螺仪在较短时间内比较准确而长期使用会有数据漂移；加速度计在较长时间数据稳定，但对振动等短时噪声敏感。因此二者具有**互补性**
  - 六轴融合常用算法：互补滤波，卡尔曼滤波，EKF或者粒子滤波
  - **九轴融合**：即使有了六轴融合，也只能用于俯仰(pitch)和横滚(roll)角, 对于偏航角(yaw), 由于和重力方向正交，加速度计无法测量，因此还需要采用其他数据，如**地磁来校准陀螺仪的偏航漂移**，常用的有AHRUpdate算法
- 3DoF：朝向数据 ( roll-yaw-pitch ) ，而6DoF的姿态数据:  
( roll-yaw-pitch ) + ( X-Y-Z )



# 4.2 手势

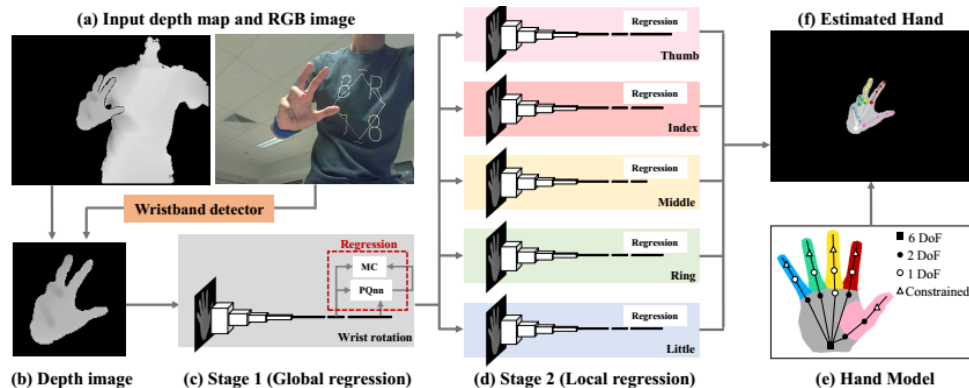
- 手柄手势
- 裸手手势
  - 传统机器学习的方法，深度学习的方法
  - 手势形状，动作的识别
  - 关节点定位和运动捕捉，例如leap motion
- 数据手套or颜色手套



静态手型和动态手势



基于手套的手势捕捉方法

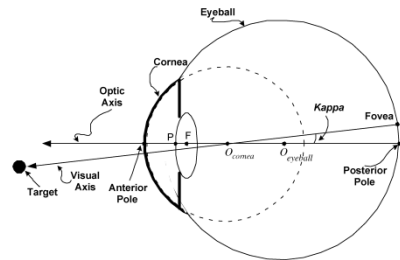
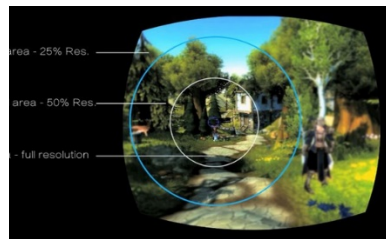


基于深度学习的方法

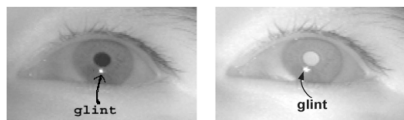


# 4.3 视线

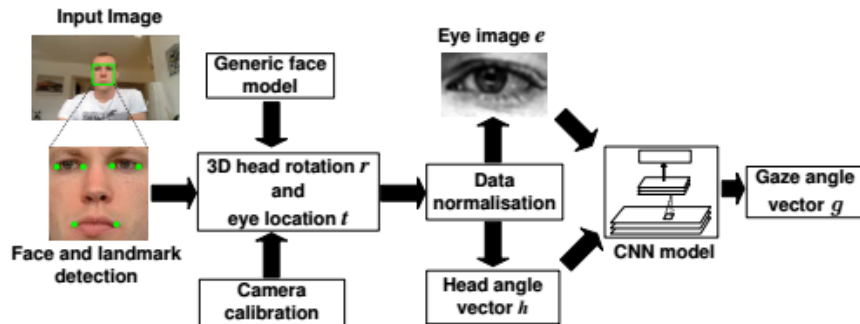
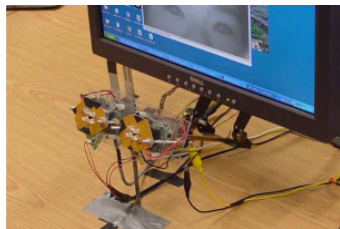
- 视线跟踪在VR中有什么用？
  - 交互
  - 关注区域渲染 ( foveated rendering )
- 跟踪方法
  - 瞳孔检测
  - 反光点检测
  - 三维视线line-of-sight重建
  - 关注区域计算



眼球成像模型



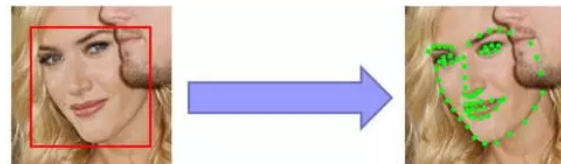
暗瞳/亮瞳和反光点



Appearance-based 视线估计方法

# 4.4 表情

- 人脸检测
  - 算法从Viola Jones到deep learning
  - 鲁棒性：侧脸，非均匀光照/高光，遮挡，眼镜/头发等等问题
- 人脸特征定位(Alignment)和跟踪
  - AAM,ASM,SDM,
- 表情识别
- 人脸三维重建
- 人脸风格化



人脸图像

特征点位置



姿态



表情



光照



遮挡



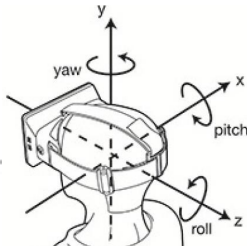
VR社交: Facebook space



## 5. 位置跟踪

- 基于6轴或9轴传感器的3DoF跟踪
- 头部/手柄的6DoF跟踪
  - 单目摄像头的方案 ( Oculus Rift )
  - 双目摄像头的方案 ( PSVR )
  - 激光光塔的方案 ( HTC Vive )
- 其他的方案
  - 蓝牙, 超声波, 红外线, 射频等等, 存在精度不够高, 抗干扰能力差等问题

# 5.1 6Dof位置跟踪



## • Outside-in 与 Inside-Out

### Outside-in method

Oculus Rift



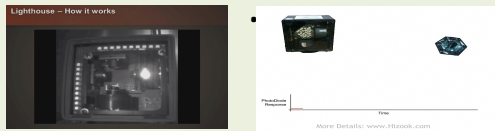
blinking.mp4

Sony PSVR



PS move tracking.mp4

HTC Vive



**优点:** 鲁棒性好, 基本不受外界环境影响; 不会增加HMD的额外负重和功耗; 计算可以在本地完成;

**缺点:** 需要单独摄像头附件, 无法做到all-in-one, 携带不便; 工作范围受摄像头FOV限制; 传输延时略高;

### Inside-out method



Oculus all-in-one



MS HoloLens



QC all-in-one

**优缺点:** 与Outside-in方式恰好相反; 算法复杂度更高;

# 5.2 Inside-out tracking / SLAM

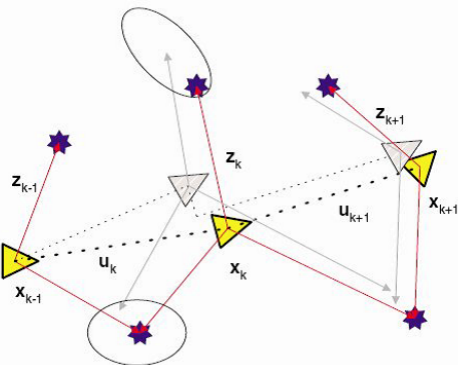
## • 传感器

- LIDAR 激光雷达：单线，多线
- 摄像头：单目，多目，IR近红外，IR远红外，结构光深度（单目/双目）TOF深度，双目深度
- 难度：单目 > 双目 > RGBD > RGBD+IMU

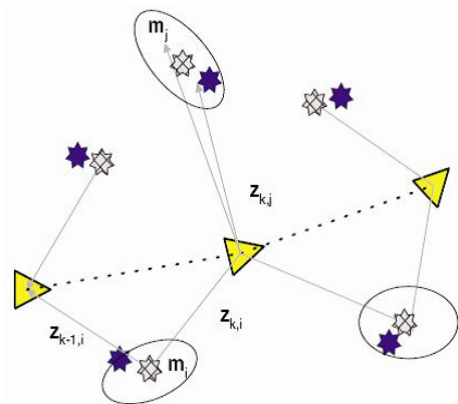


- V-SLAM: visual SLAM (Simultaneous Localization and mapping)
- Localization: 通过地图中的landmark计算当前位置和姿态；
- Mapping: 当前位置已知对未知landmark进行测量，建立地图

- $x_k$ : location of vehicle at time  $k$
- $u_k$ : a control vector applied at  $k-1$  to drive the vehicle from  $x_{k-1}$  to  $x_k$
- $z_k$ : observation of a landmark taken at time  $k$
- $X^k$ : history of states  $\{x_1, x_2, x_3, \dots, x_k\}$
- $U^k$ : history of control inputs  $\{u_1, u_2, u_3, \dots, u_k\}$
- $m$ : set of all landmarks



- $X^k$ : history of states  $\{x_1, x_2, x_3, \dots, x_k\}$
- $z_k$ : observation of a landmark taken at time  $k$
- $m_i$ : true location of the  $i^{\text{th}}$  landmark
- $m$ : set of all landmarks



# 5.3 vSLAM

- monoSLAM: 单目SLAM
  - 无确切尺度scale, 在相似变换空间Sim(3)而不是欧式空间SE(3)
  - 需要motion parallax, 如果是静止或者同心旋转则失效
- binoSLAM: 双目SLAM
  - 通过stereo matching可直接得到深度, 静止或纯旋转也可行
  - 受基线大小和分辨率的限制
- RGBD-SLAM: 基于深度的SLAM
  - 无需费力计算深度, 直接使用点云进行计算
  - 受depth sensor性能影响: 帧速, 分辨率, 精度



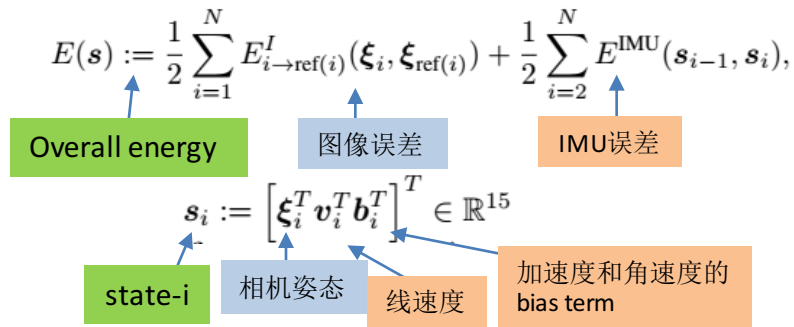
- VSLAM的四大组成部分: VO, 后端优化, 建图, loop检测
  - **VO**: visual odometry, 计算相邻帧直接的运动, 基于特征的方法就是PnP问题, 得到相对的R和T; 非特征的方法, 例如RGBD-SLAM中的ICP ( Iterative Closet Point ), 通过迭代直接求解变换矩阵, 还有在单目条件下的SVO(fast semi-direct VO, ICRA, 2014), LSD-SLAM(large scale direct SLAM, ECCV' 2014);
  - **后端优化**: 解决VO的累计误差drift问题。早期用filtering处理, KF/EKF/... ; 现在主要是用BA ( bundle adjustment, 光束平差 ), 基于图优化 ( graph optimization ) 的方法
  - **Loop closure detection**, 闭环检测。是一种检测观测数据相似性的方法, 目的是降低累计误差 ( accumulated error )。常用的是bag-of-word ( BOW ) 模型, 或其他在线学习分类的方法。

# 5.4 IMU+VO=VIO

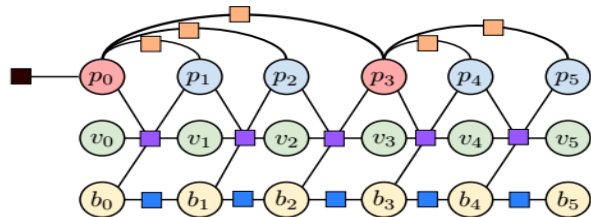
- Visual-Inertial-Odometry：视觉传感器与惯性传感器的互补和融合，视觉准且慢，陀螺仪快但飘
- Filtering-based VIO and optimization-based VIO, 早期当做pure sensor fusion问题，用filtering方法求解，二者是loosely-coupled的关系。当前主流是作为**最优化问题**求解，二者是**紧耦合**关系。
- Direct VIO with stereo camera, ICRA2016. (stereo LSD-SLAM基础上的延续)
- 重新定义了全局能量目标函数，L-M方法求最优解

## 图像和IMU信息的互补性:

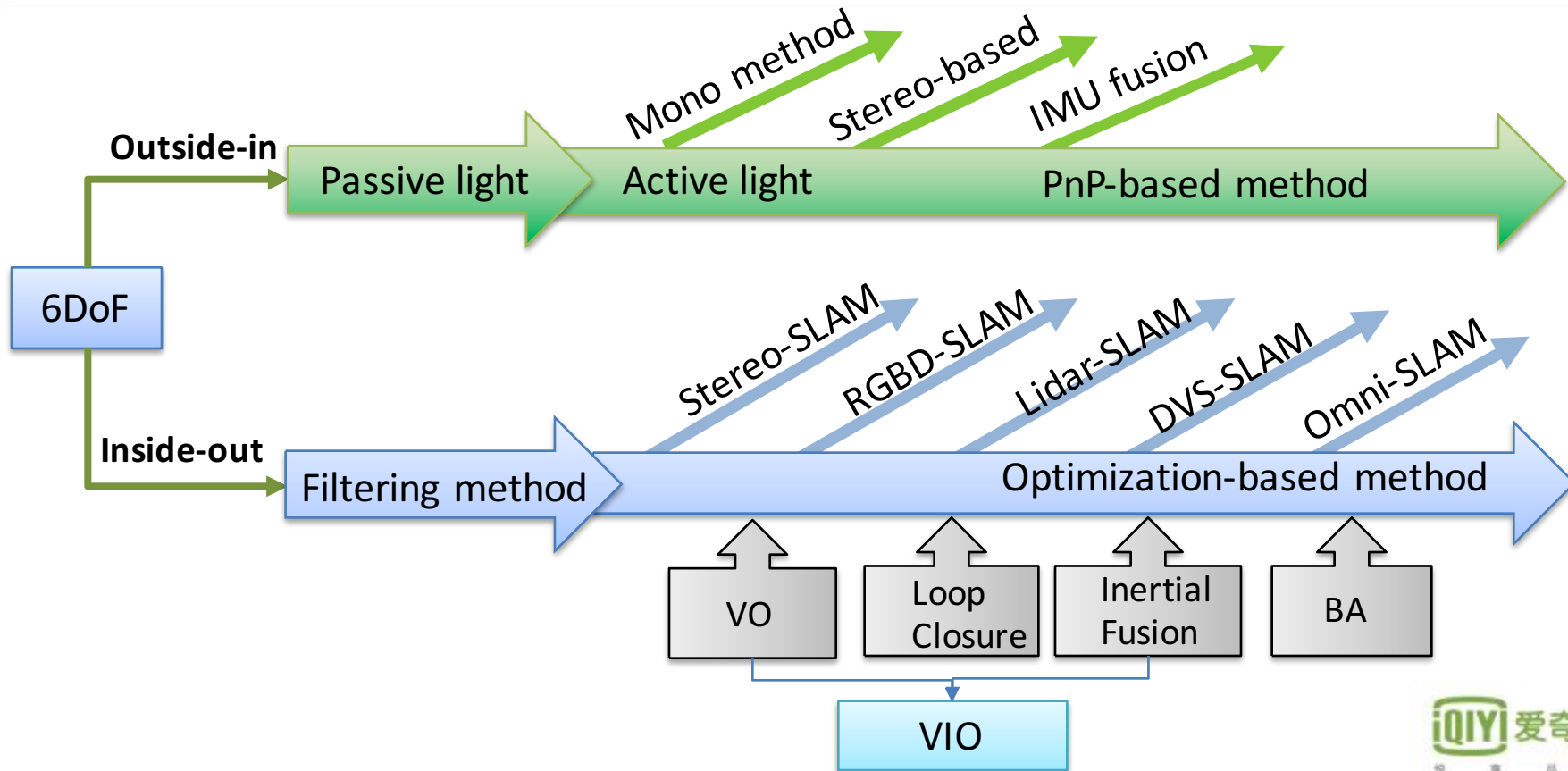
1. 图像在纹理少的区域失效的情况下，IMU仍可以正常工作；
2. 摄像头一般帧速比较慢，而IMU的采样率可以很高。针对头部快速运动的情况，IMU可以运动预测有效避免跟踪失败；
3. 缩小帧间匹配特征点的搜索范围；



$$E(s) = \frac{1}{2} r^T W r$$
$$= \frac{1}{2} \begin{bmatrix} r_I^T & r_{\text{IMU}}^T \end{bmatrix} \begin{bmatrix} W_I & 0 \\ 0 & W_{\text{IMU}} \end{bmatrix} \begin{bmatrix} r_I \\ r_{\text{IMU}} \end{bmatrix}.$$



# 5.5 6DoF跟踪summary



## 6. 当VR遇到AI

1. 根据用户喜好习惯的VR内容自动推荐
2. 身份认证
3. UGC内容的自动标注与管理
4. 虚拟助手“双儿”的自然交互与行为养成



# 爱奇艺的VR产品

## “奇遇”一体机







IT大咖说  
知识分享平台



联系方式: [wangxiying@qiyi.com](mailto:wangxiying@qiyi.com)