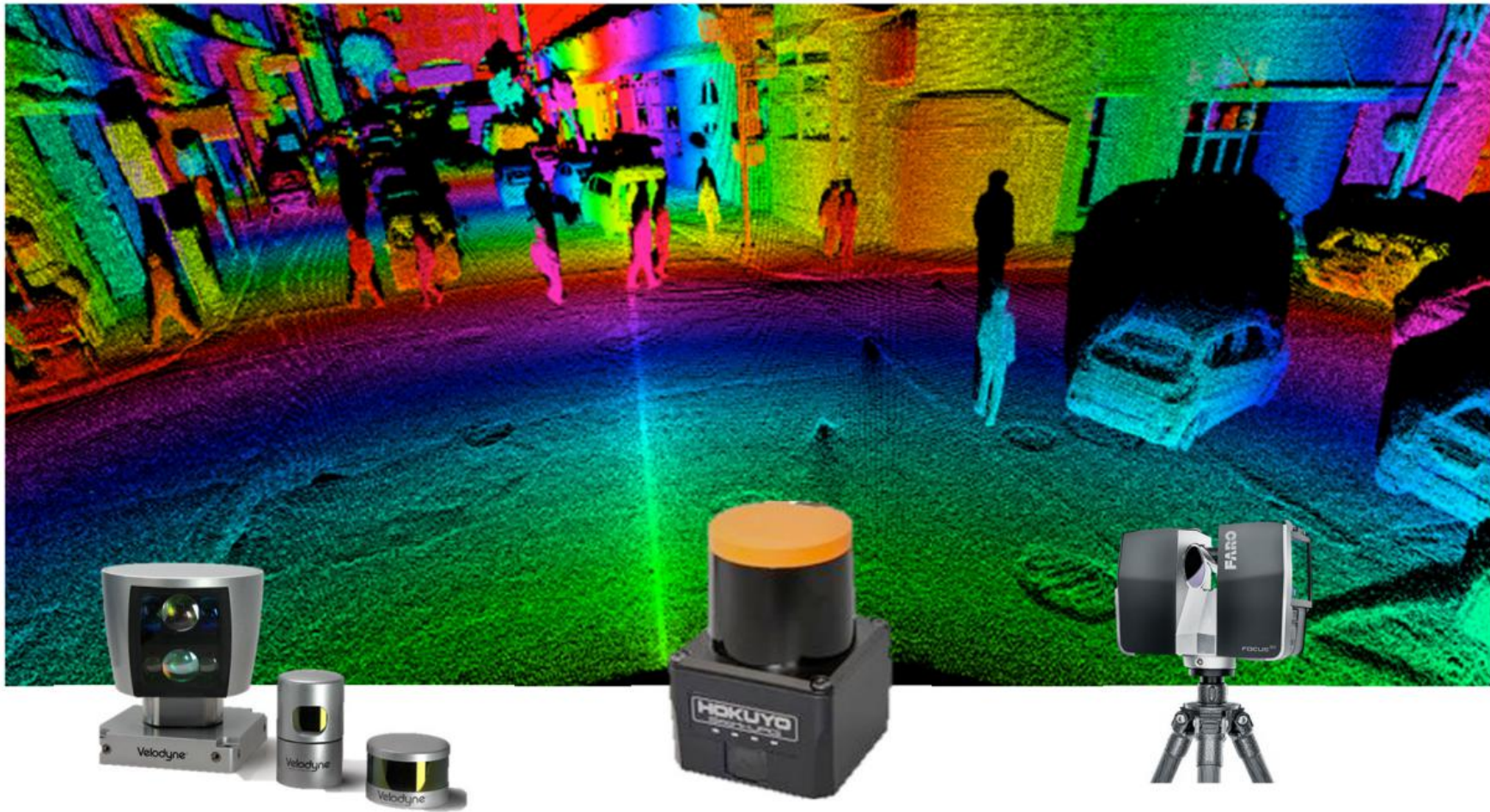


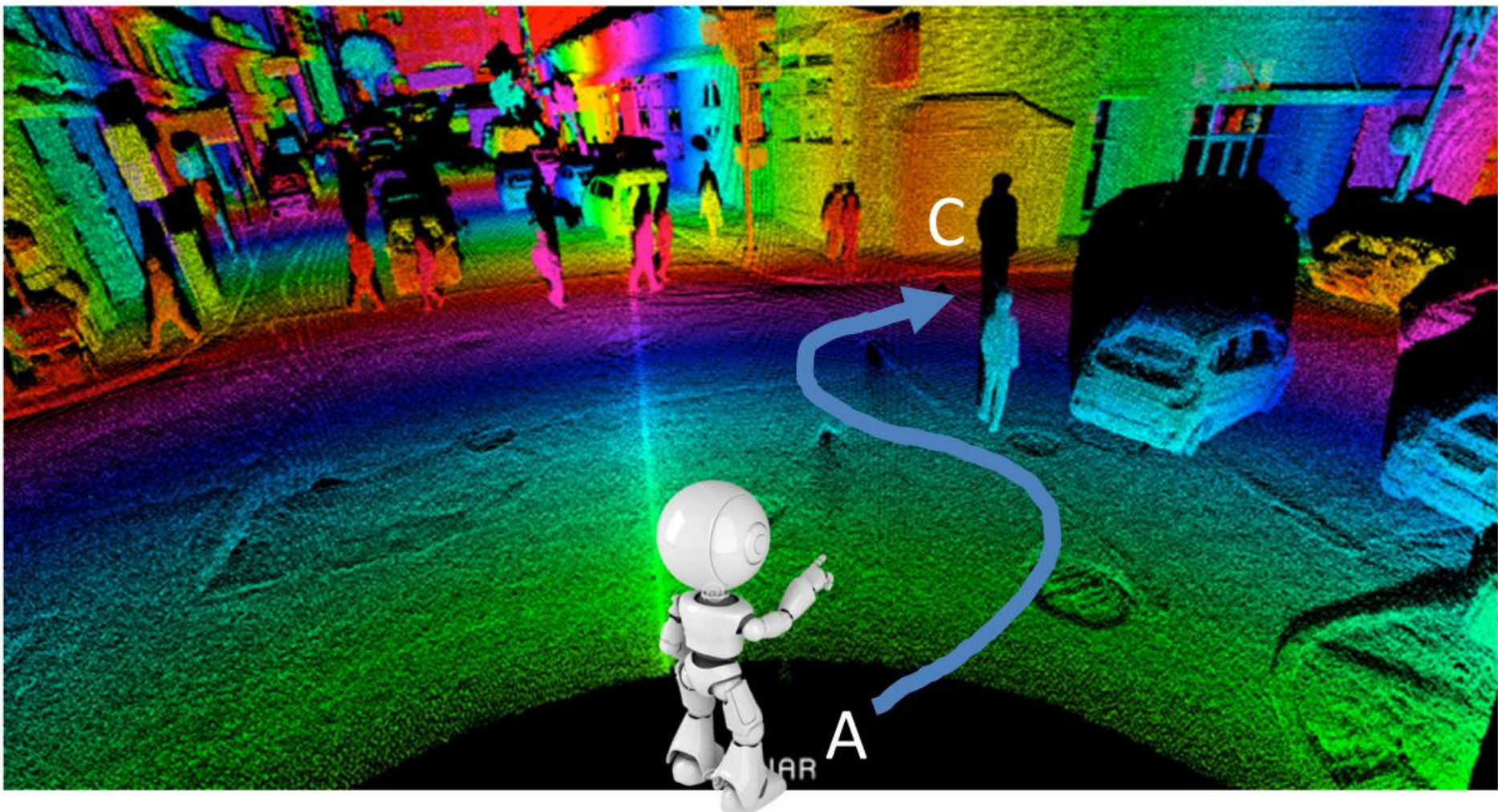
三维重建基础

✓ 应用领域概述



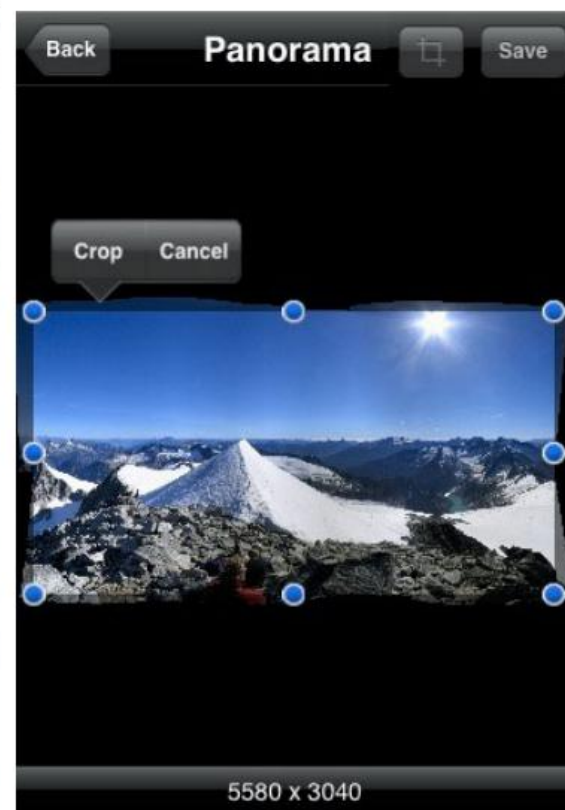
三维重建基础

✓ 应用领域概述



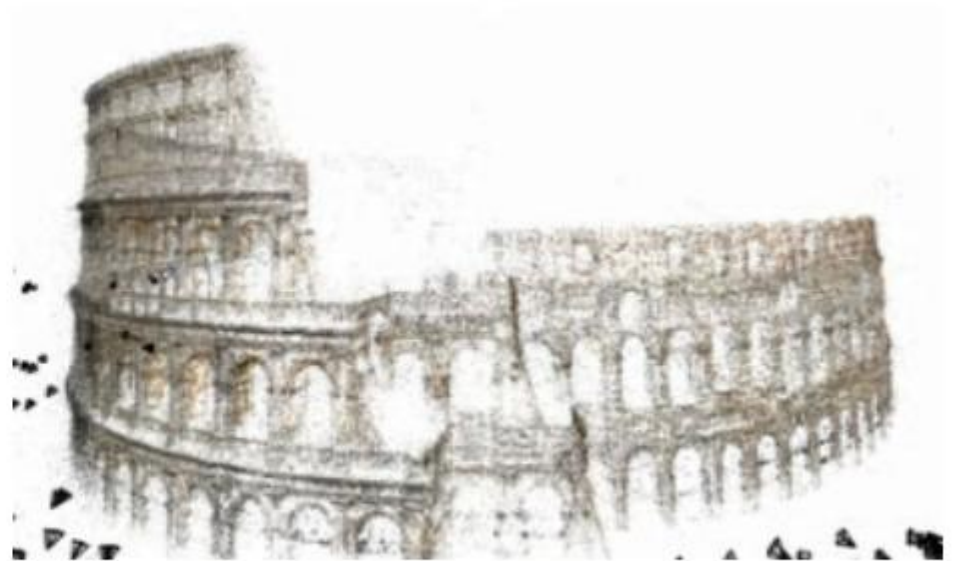
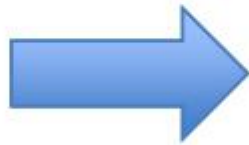
三维重建基础

✓ 应用领域概述



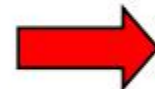
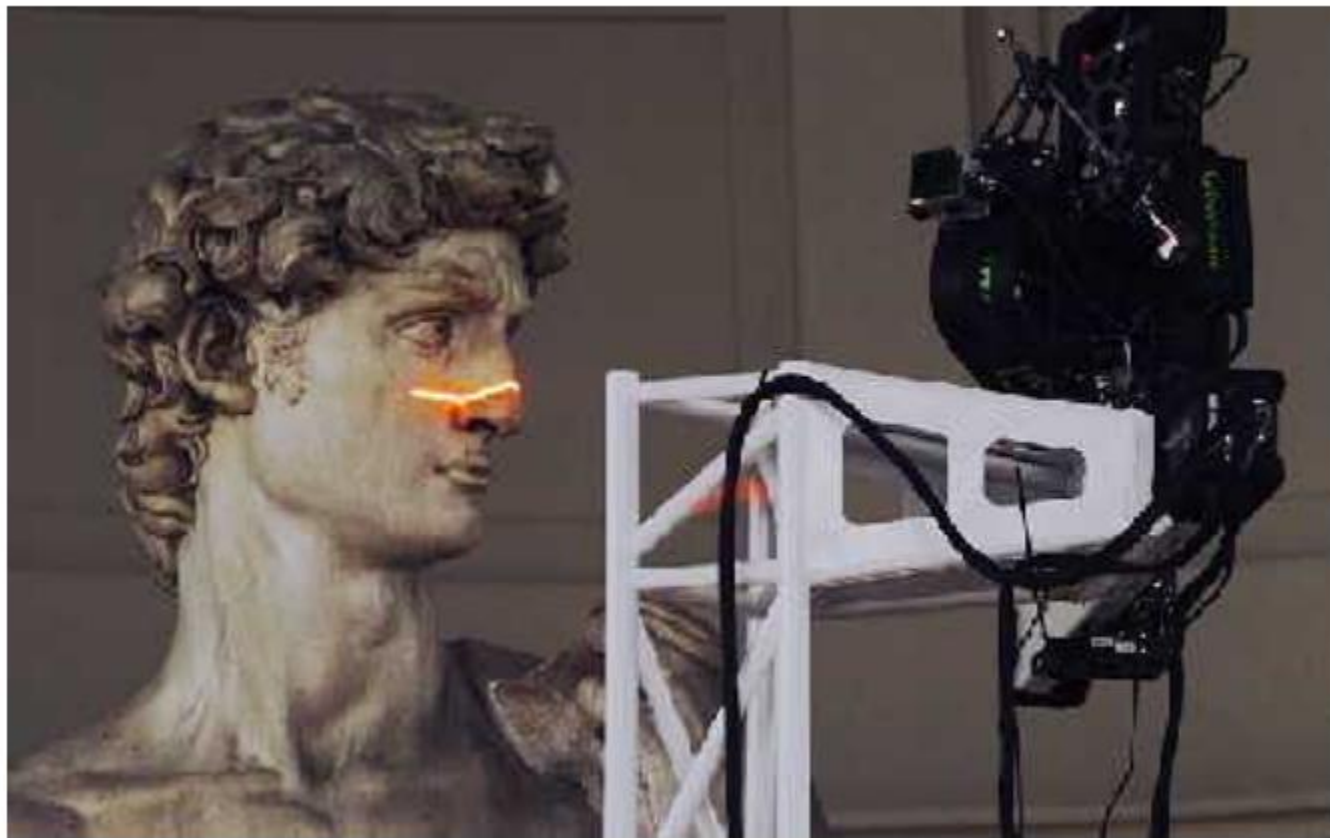
三维重建基础

✓ 应用领域概述



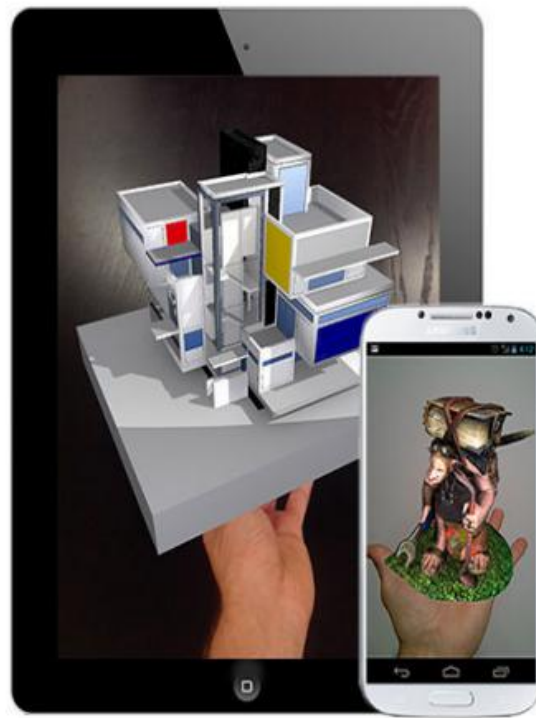
三维重建基础

✓ 原型设计



三维重建基础

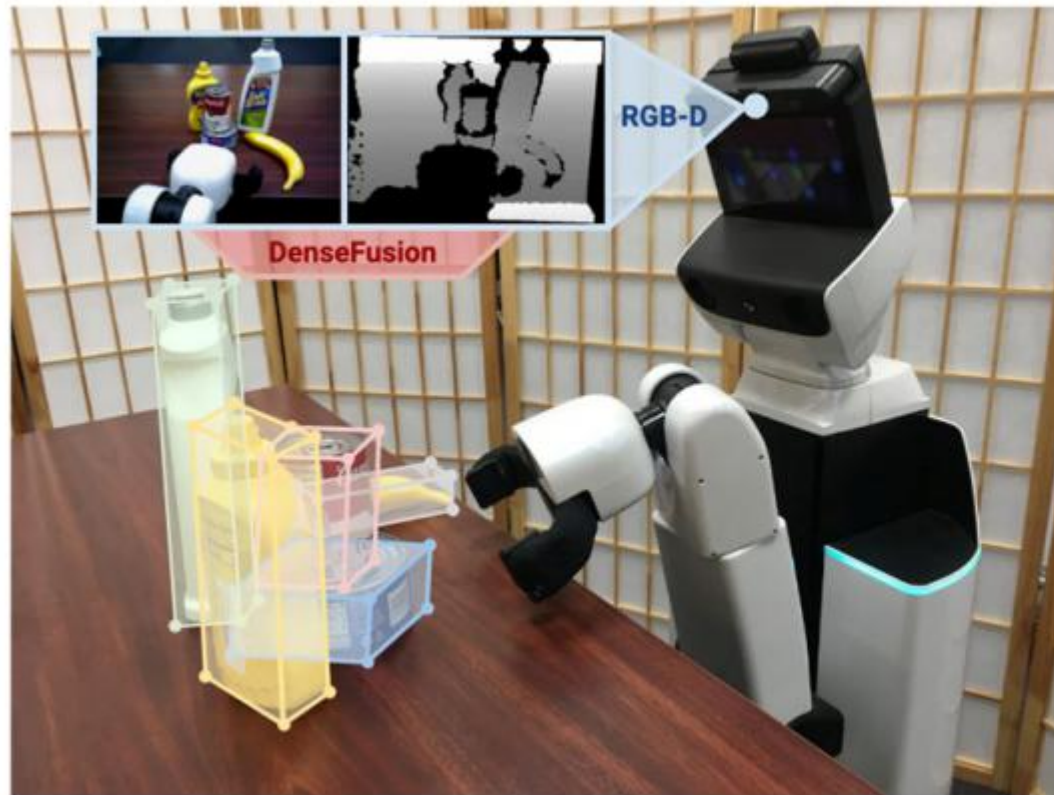
✓ 虚拟现实



- Magic leap
- Daqri
- Meta
- Etc...

三维重建基础

✓ 机械手臂



三维重建基础

✓ 辅助驾驶

[▶▶ manufacturer products](#)[consumer products ◀◀](#)

Our Vision. Your Safety.



rear looking camera

forward looking camera

side looking camera

EyeQ Vision on a Chip

[> read more](#)

Vision Applications



Road, Vehicle, Pedestrian Protection and more

[> read more](#)

AWS Advance Warning System

[> read more](#)

News

- > [Mobileye Advanced Technologies Power Volvo Cars World First Collision Warning With Auto Brake System](#)
- > [Volvo: New Collision Warning with Auto Brake Helps Prevent Rear-end](#)

[> all news](#)

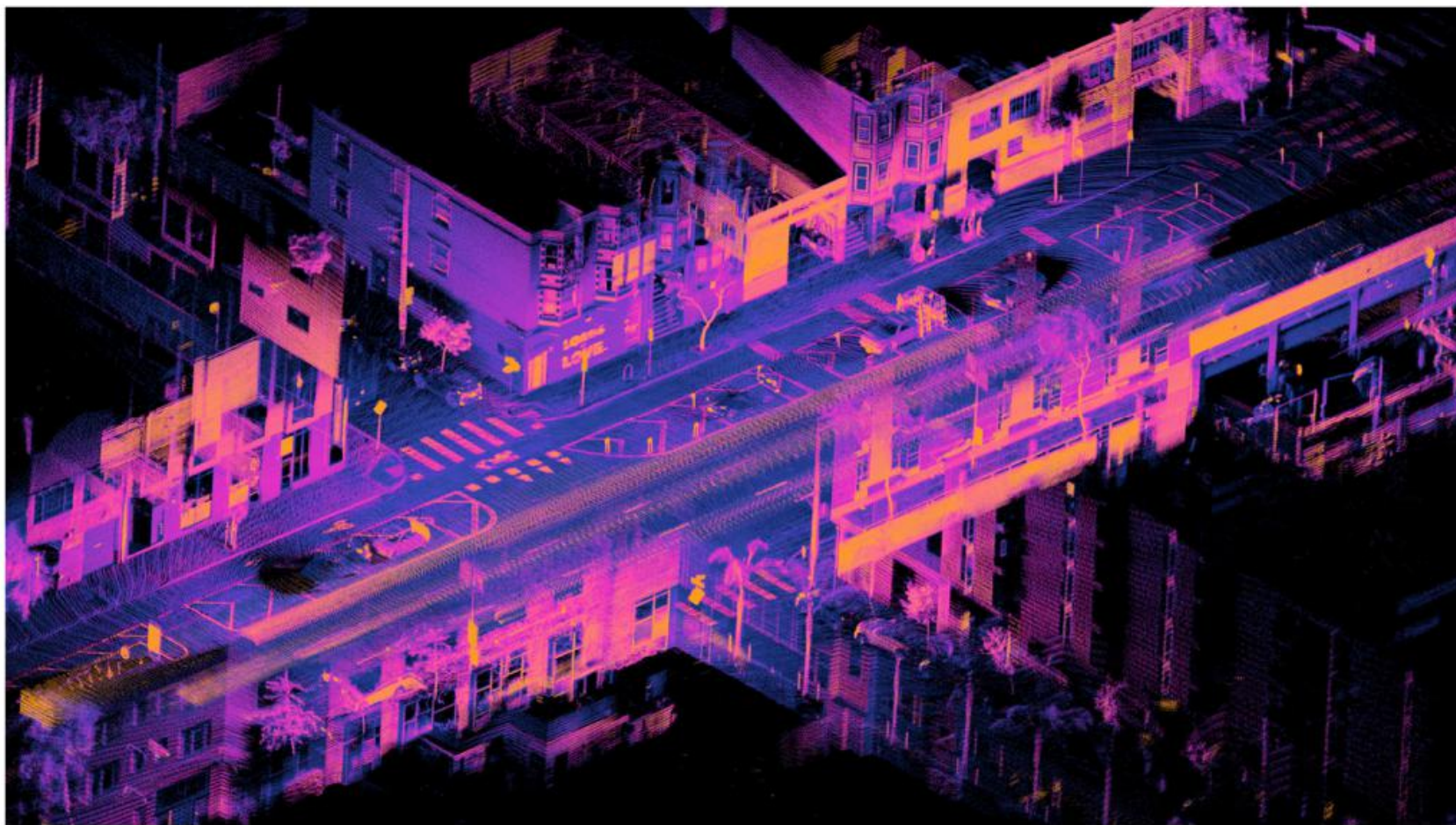
Events

- > [Mobileye at Equip Auto, Paris, France](#)
- > [Mobileye at SEMA, Las Vegas, NV](#)

[> read more](#)

三维重建基础

✓ 定位与追踪



三维重建基础

✓ 如何三维重建？

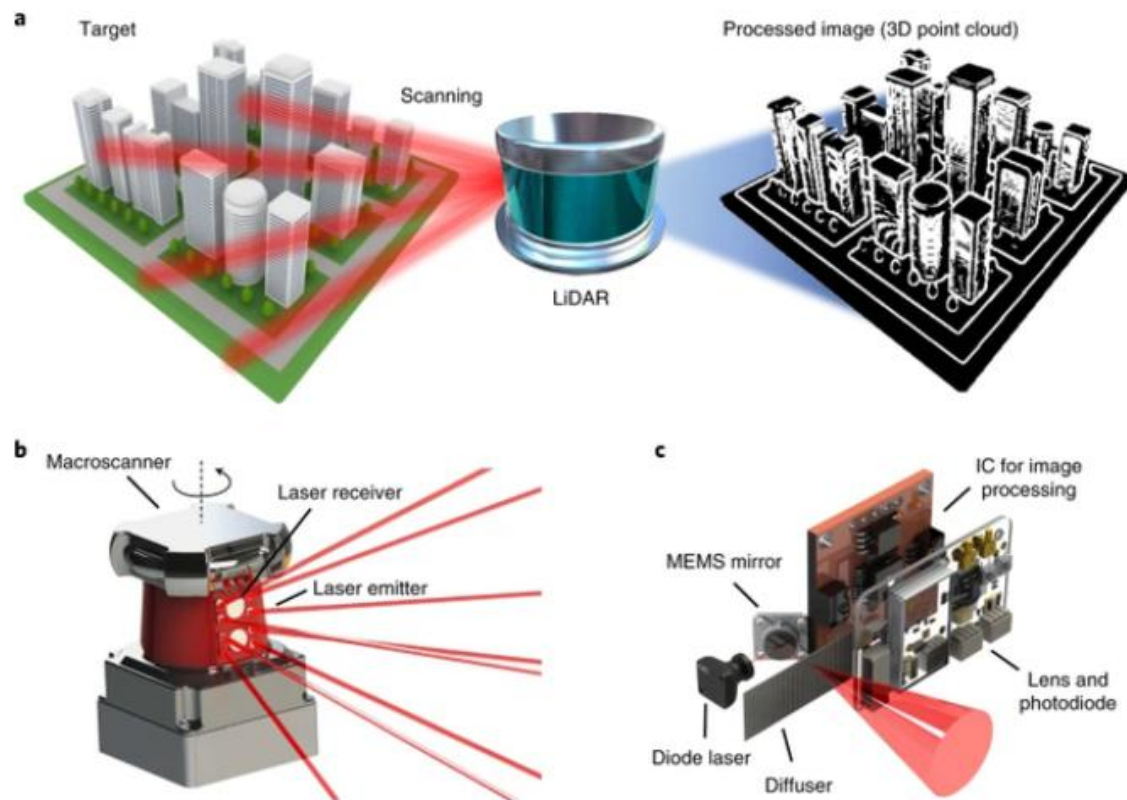
✎ 人工，软件等方法慢慢磨（那是相当的慢啊）



三维重建基础

✓ 如何三维重建？

✎ 激光雷达等获取深度与3维信息（这个可贼拉贵）



三维重建基础

✓ 如何三维重建？

✎ 使用各个视角图像进行三维重建（这个便宜还容易）



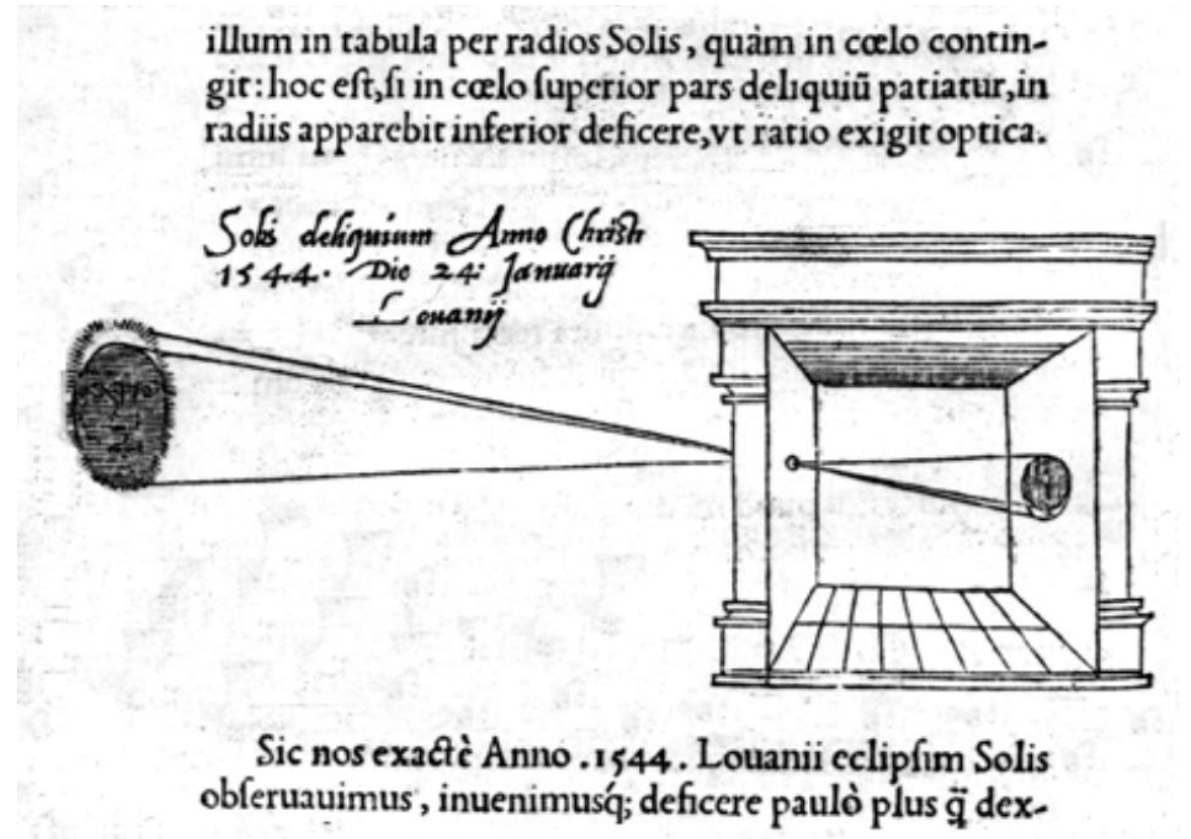
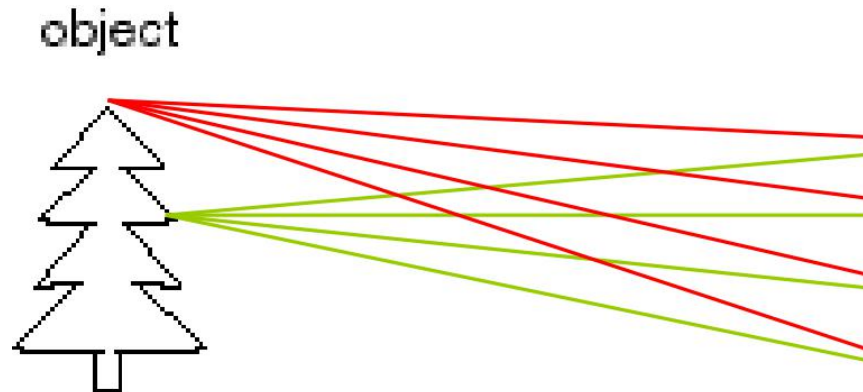
三维重建基础

✓ 相机成像

📌 这件事有几百年了！

📌 为啥需要小孔成像？

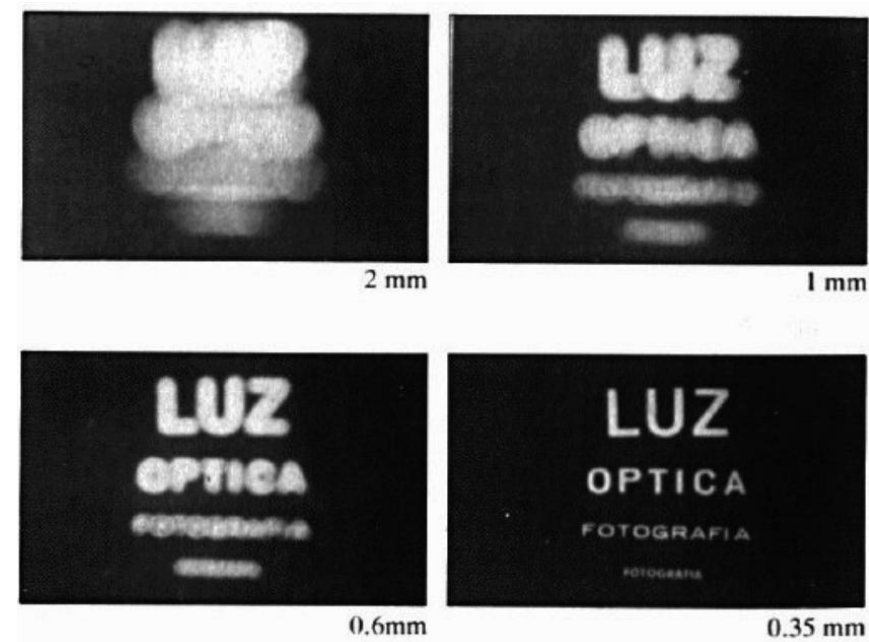
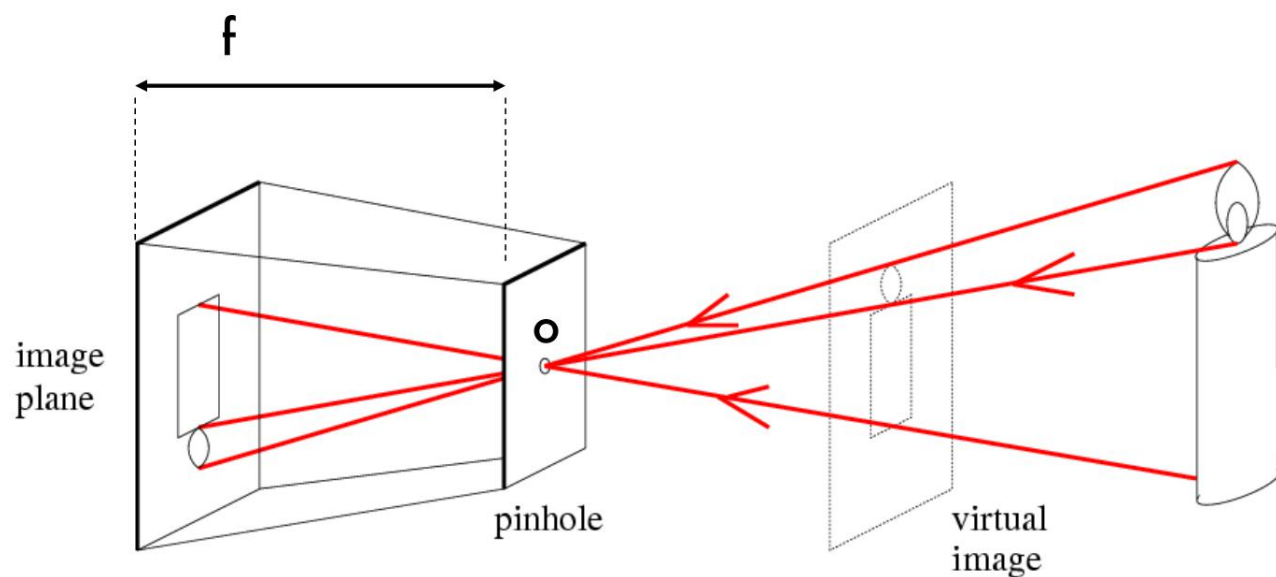
📌 没有小孔，同一点会落很多



三维重建基础

✓ 整体概述: (Truncated Signed Distance Function)

✎ f 就是焦距, o 就是光圈; 右图为光圈大小对清晰度的影响



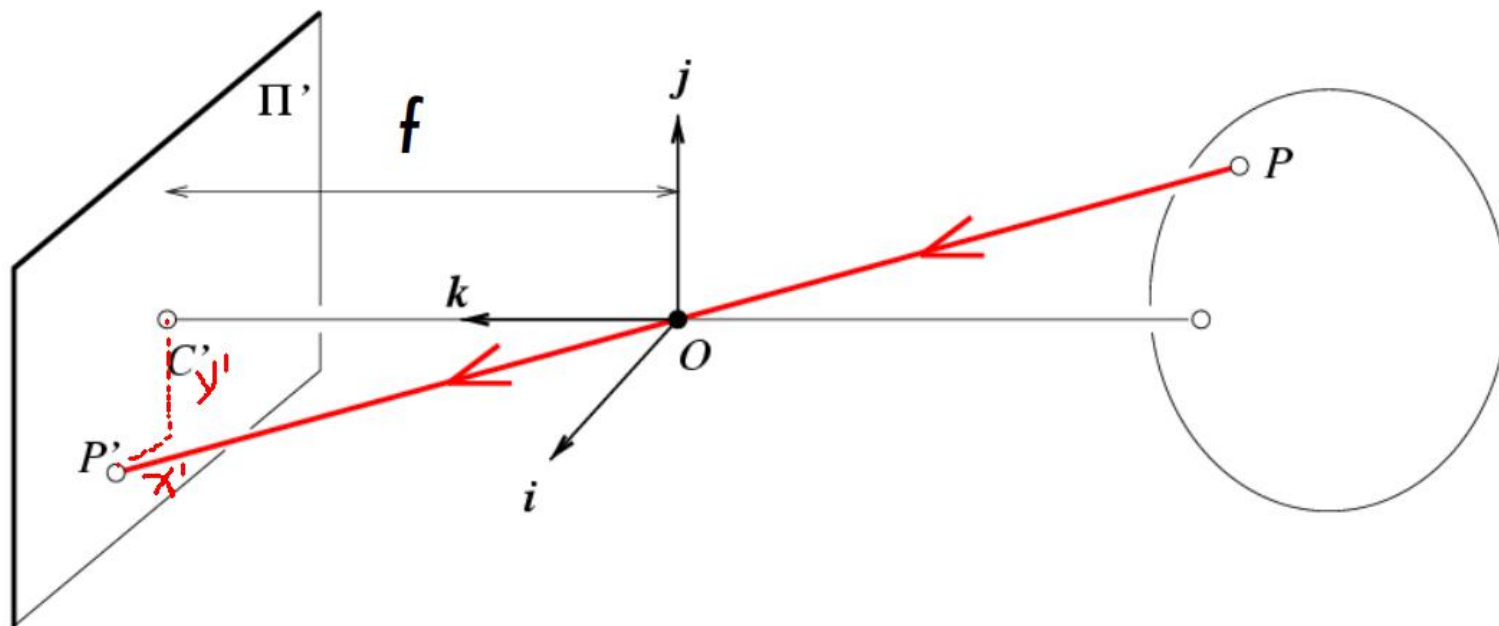
三维重建基础

✓ 相机坐标系与像平面坐标系

✎ 空间点p在图像中哪呢?

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow P' = \begin{bmatrix} x' \\ y' \end{bmatrix} \quad \begin{cases} x' = f \frac{x}{z} \\ y' = f \frac{y}{z} \end{cases}$$

✎ 要得到其映射关系（通过相似三角形，将摄像机坐标系P->像平面坐标系p'）



三维重建基础

✓ 像素坐标系

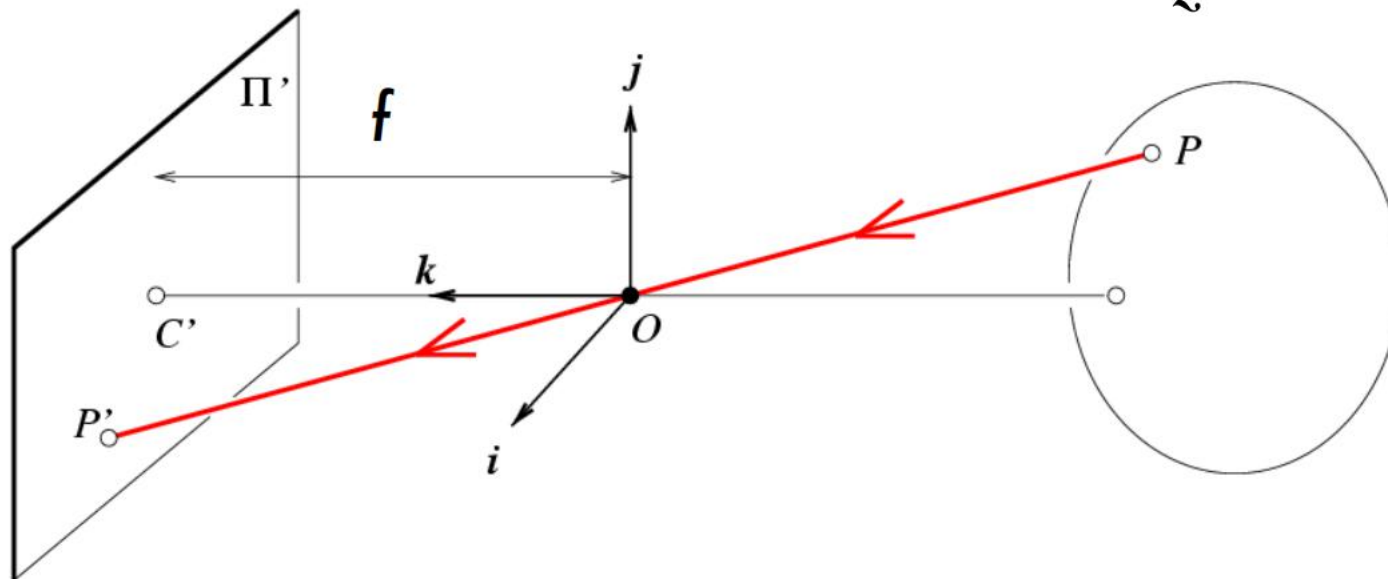
$$(x, y, z) \rightarrow \underbrace{\left(f \ k\right)}_{\alpha} \frac{x}{z} + c_x, \underbrace{\left(f \ l\right)}_{\beta} \frac{y}{z} + c_y$$

✎ 注意偏置(像素不是以中心开始) Units: k, l : pixel/m (看你设备咋样0.01m/像素)

f : m

✎ 注意还需要转换单位:

$$P = (x, y, z) \rightarrow P' = \left(\alpha \frac{x}{z} + c_x, \beta \frac{y}{z} + c_y\right)$$



三维重建基础

✓ 线性?

✎ 对应关系是线性的? (例如两个坐标系X的对应, 其中 α 固定, 但是Z呢?)

$$P = (x, y, z) \rightarrow P' = \left(\alpha \frac{x}{z} + c_x, \beta \frac{y}{z} + c_y \right) \quad (X \text{ 变了, } Z \text{ 大概率也变了})$$

✎ 如何解决这个问题呢? (如果非线性的, 那就很难转换了)

✎ 使用齐次坐标来完成这个任务

三维重建基础

✓ 齐次坐标

✎ 齐次坐标变换: $(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ $(x, y, z) \Rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$

✎ 其实就是增加一个维度, 欧氏空间 \rightarrow 齐次空间

✎ 在变换回来: $\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$ $\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$

三维重建基础

✓ 坐标转换

📎 齐次坐标中的转换（就是为了得到变换矩阵，且这个矩阵是不变的）

$$P_h' = \begin{bmatrix} \alpha x + c_x z \\ \beta y + c_y z \\ z \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \text{--- } P_h$$

Homogenous

Euclidian

$$\underbrace{P_h'}_{\text{Homogenous}} \rightarrow \underbrace{P' = \left(\alpha \frac{x}{z} + c_x, \beta \frac{y}{z} + c_y \right)}_{\text{Euclidian}}$$

三维重建基础

✓ 摄像机内参K

✎ 表示空间点到图像中的对应关系

✎ 相机本身参数，固定的

✎ 后续项目中数据集会给出内参

✎ 也是三维重建中必须知道的一个指标

Camera matrix K

$$P' = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

三维重建基础

✓ 任何定义物体的位置呢?

✎ 你往左，我往右会撞上吗?

✎ 这个就需要世界坐标系了

✎ 来捋一捋咱们现在有3个坐标系了

✎ 现在还得想办法怎么转换世界坐标

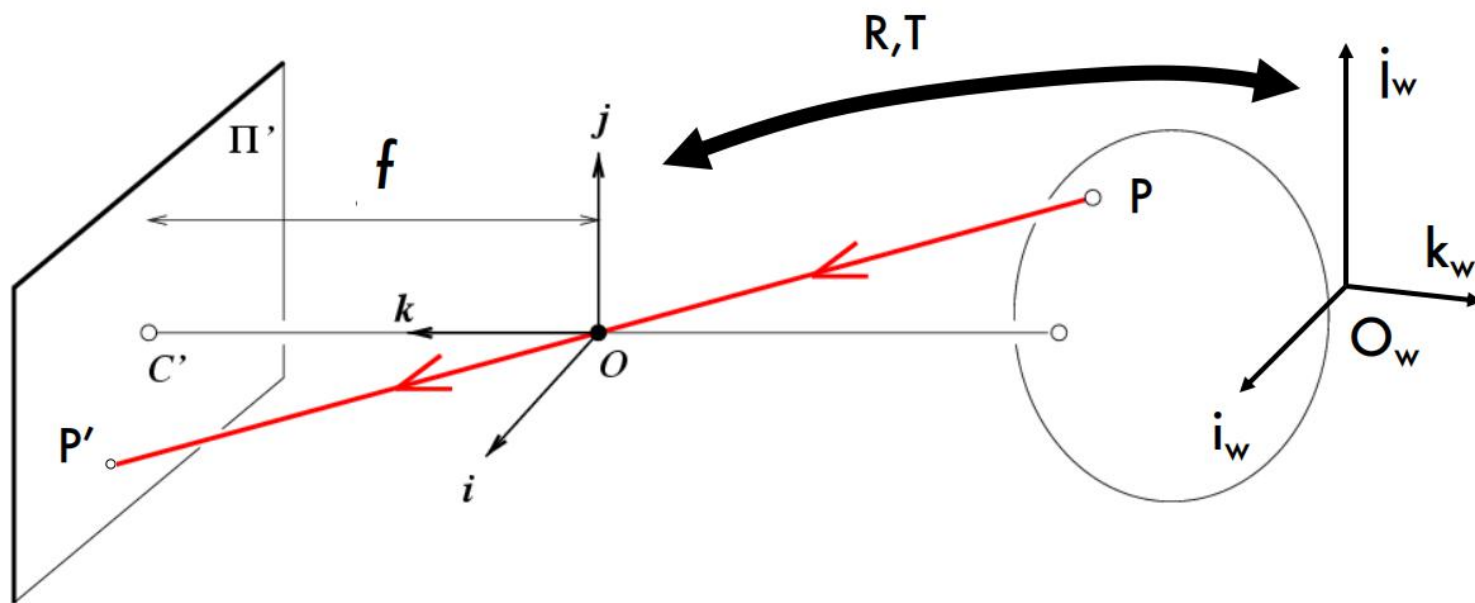


三维重建基础

✓ 世界坐标系与相机坐标系

✎ 世界->相机, 需要一个旋转平移矩阵: $P = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4 \times 4}$ $P_w \leftarrow \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$

✎ 其中R是3*3矩阵, T是3*1矩阵 (X,Y,Z三个方向), RT就是摄像机位姿



三维重建基础

✓ 坐标系变换

✎ 像素与世界坐标系的关系（其实需要我们知道内外参就可以映射）

✎ 像素坐标系->相机坐标系->世界坐标系的对应：

$$P' = K \begin{bmatrix} I & 0 \end{bmatrix} P = K \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}_{4 \times 4} P_w = K \begin{bmatrix} R & T \end{bmatrix} P_w$$

Internal parameters

External parameters

三维重建基础

✓ 相机标定

✎ 要做一件什么事呢？

✎ 就是求解相机的内外参数

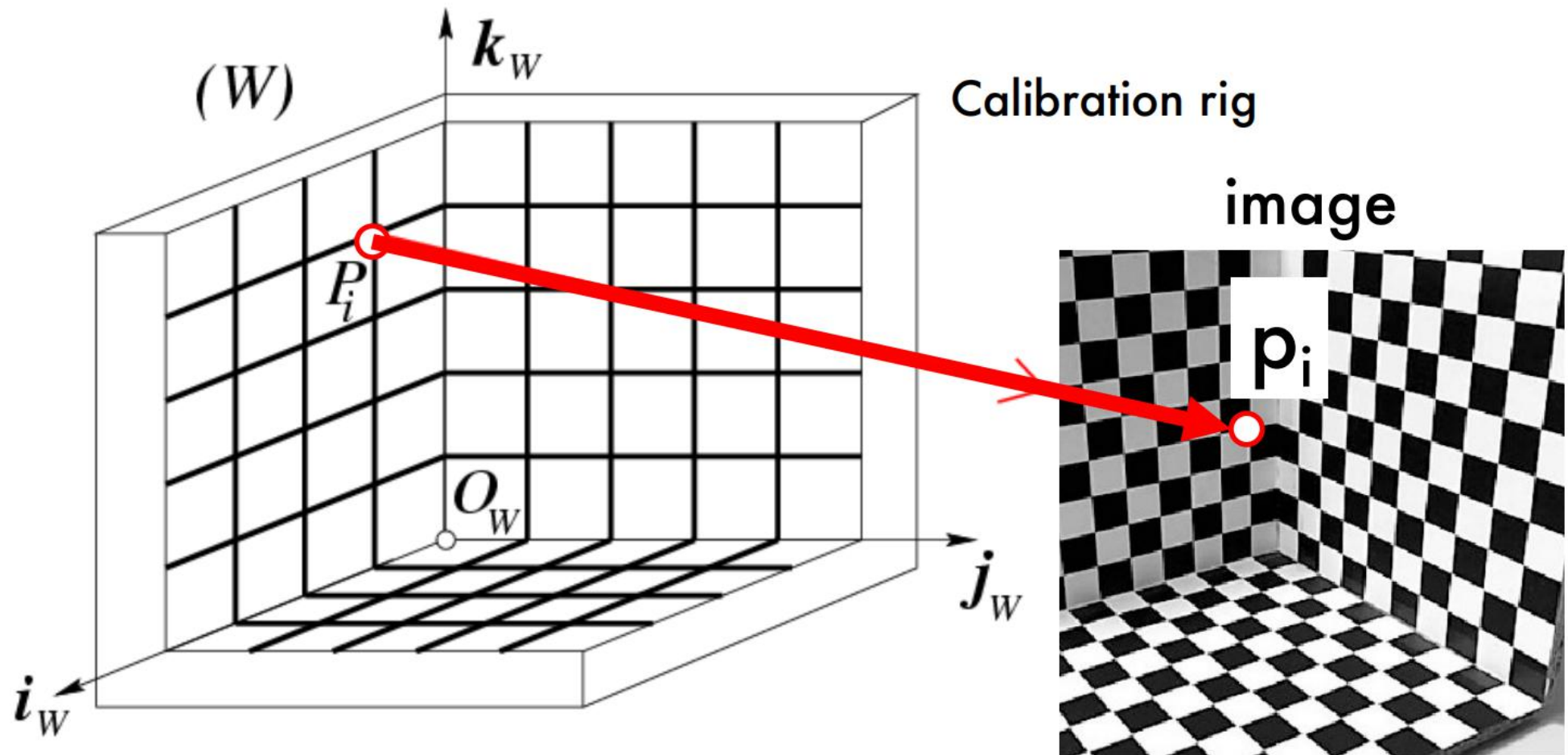
✎ 那你就得告诉我像素坐标和世界坐标，才能求解中间的参数

✎ 相机标定就是利用多组对应位置，如上图，来求解相机内外参数的过程

$$P' = M P_w = \underbrace{K}_{\text{Internal parameters}} \underbrace{[R \quad T]}_{\text{External parameters}} P_w$$

三维重建基础

✓ 相机标定



三维重建算法

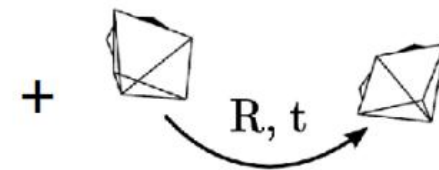
✓ 商汤: NeuralRecon

✎ 矛盾点: 落地所需的设备

✎ 高端的食材咋地都好吃

✎ 但很多应用要集成到普通摄像头中

✎ NeuralRecon就是单目解决方案

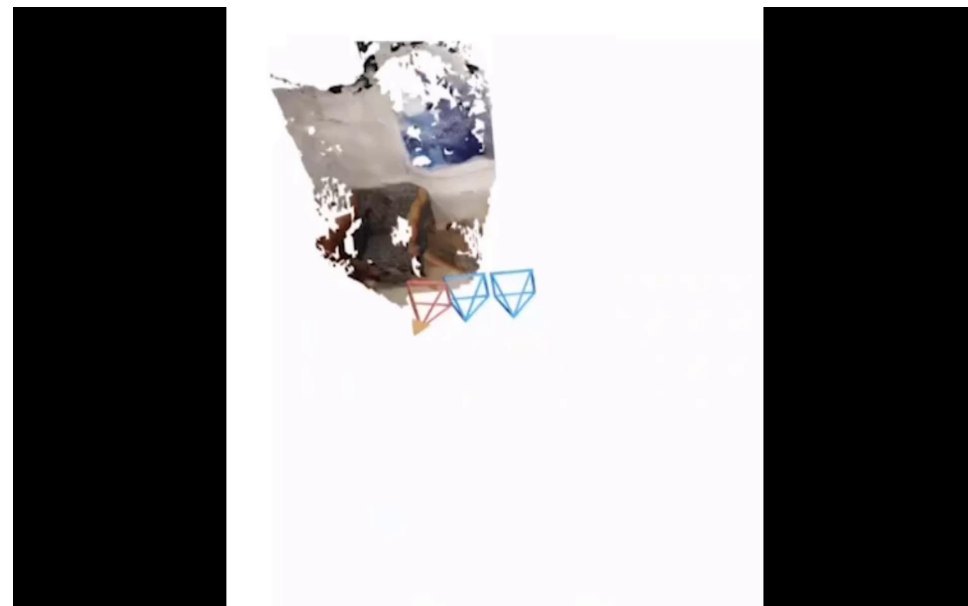
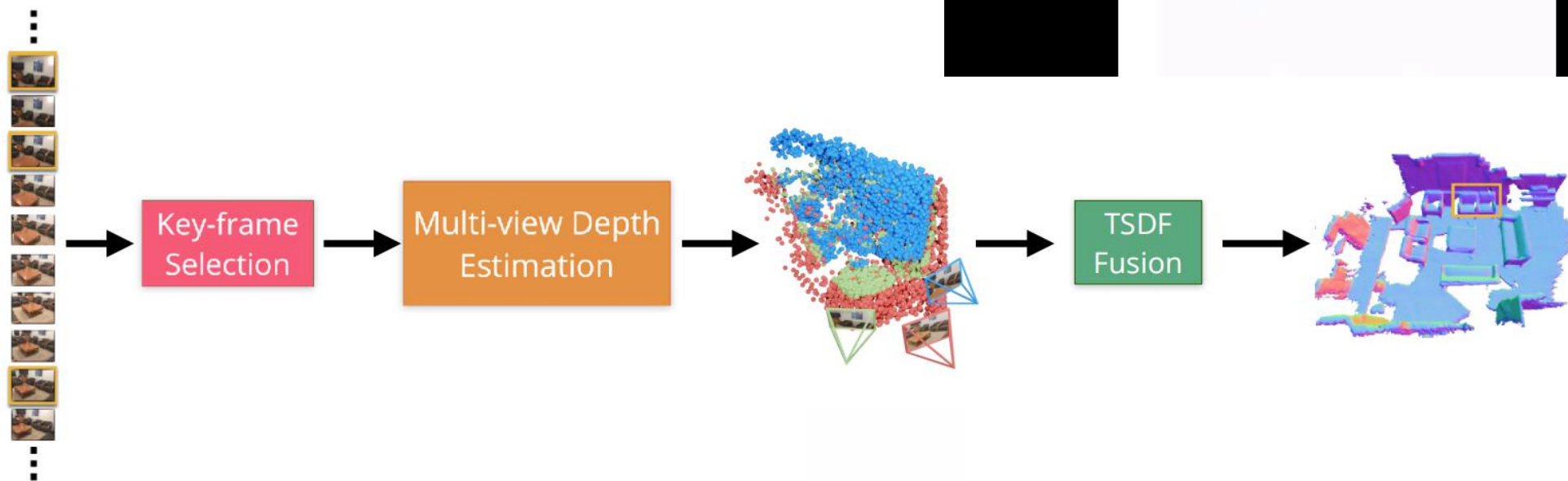


三维重建算法

✓ 传统任务流程

✎ 输入序列选择，深度估计，点云，融合

✎ 你能确定每一步都做的准嘛？



三维重建算法

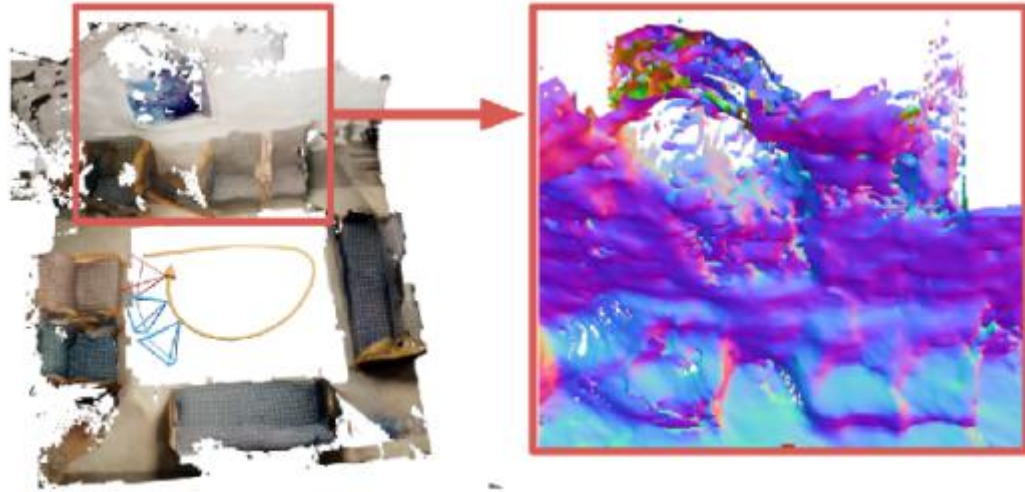
✓ 可能遇到的问题

✎ 深度估计结果尺度不一致：

✎ 各自为政，想玩到一块不容易

✎ 重复的计算非常多，输入序列中肯定很多位置重复了

✎ 每个位置都要取计算它的深度信息，速度大打折扣



三维重建算法

✓ NeuralRecon要做的事

✎ 一句话总结就是：擒贼先擒王，省略掉中间过程，直接预测想要的

✎ 中间过程就相当于深度信息，点云信息等统统不需要，直接输出结果

✎ 那中间的事谁管呢？爱谁谁吧，交给神经网络就得了

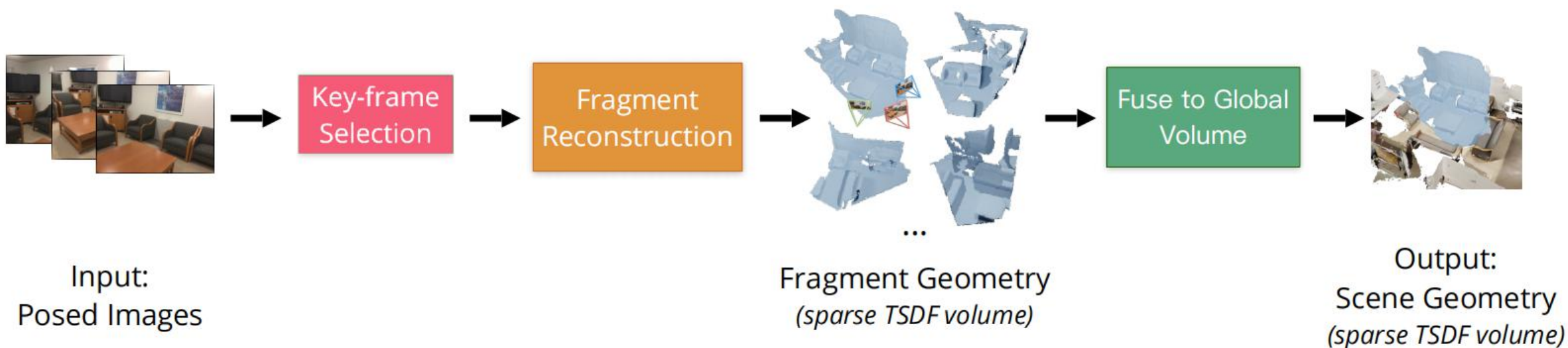
✎ 既不用高端设备，也不计算中间结果，直接End2End的一个框架

三维重建算法

✓ 整体框架

✎ 1.关键帧选择; 2.输入序列(9张图); 3.片段重建; 4.全局合成

✎ 核心就是中间的网络, 如何利用图像信息估计TSDF值

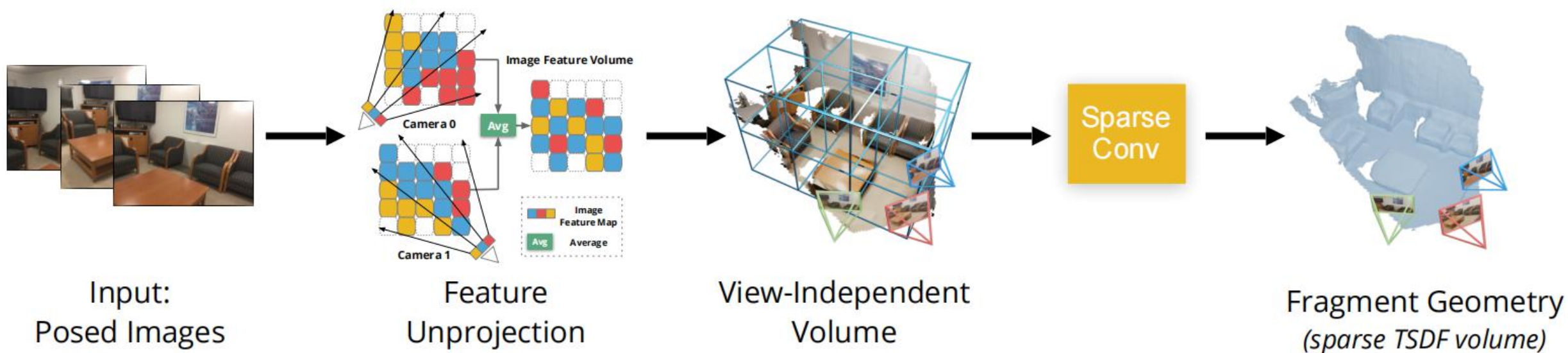


三维重建算法

✓ NeuralRecon框架

✎ 核心就是特征映射，如何将图像中的特征映射到重建的结果上

✎ 其实就是通过相机的内外参将像素坐标系与世界坐标系对应



三维重建算法

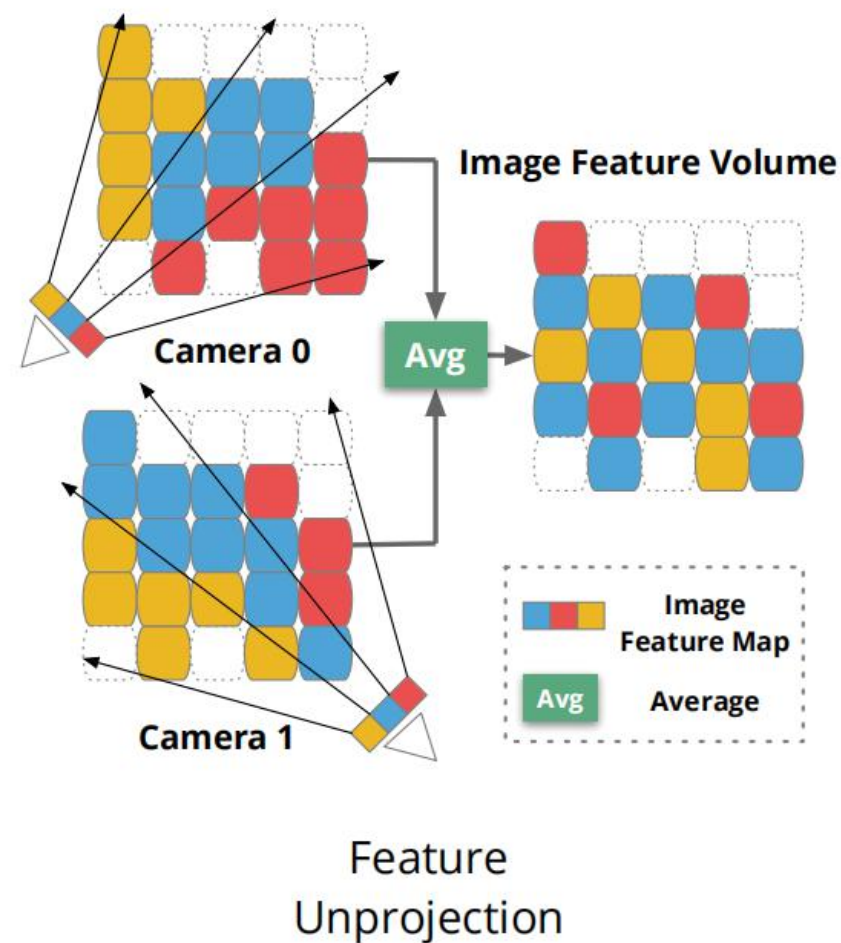
✓ 映射细节

✎ 例如输入特征图：40*40*80

✎ 体素一共有1W个小方块（可以自己设置）

✎ 映射方法：
$$P' = M P_w = \underbrace{K}_{\text{Internal parameters}} \underbrace{\begin{bmatrix} R & T \end{bmatrix}}_{\text{External parameters}} P_w$$

✎ 平均的意思就是1W的小块中可能每一个对应不同图像（例如9个）的特征，那就求个平均

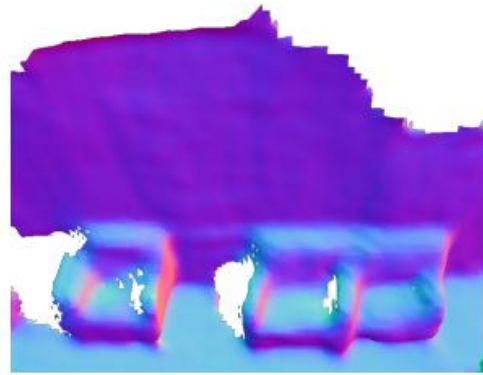


三维重建算法

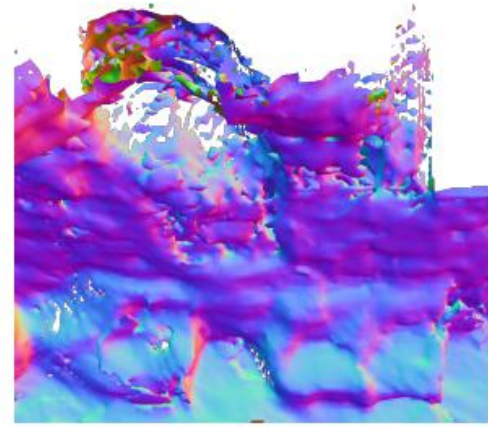
✓ 直接预测的优势

✎ 不需要中间结果

✎ 看来更平滑稳定



Volume-based



Depth-based

✎ 更大的优势就是简单方便，一步搞定；（要把大象放冰箱里，第一步。。。）

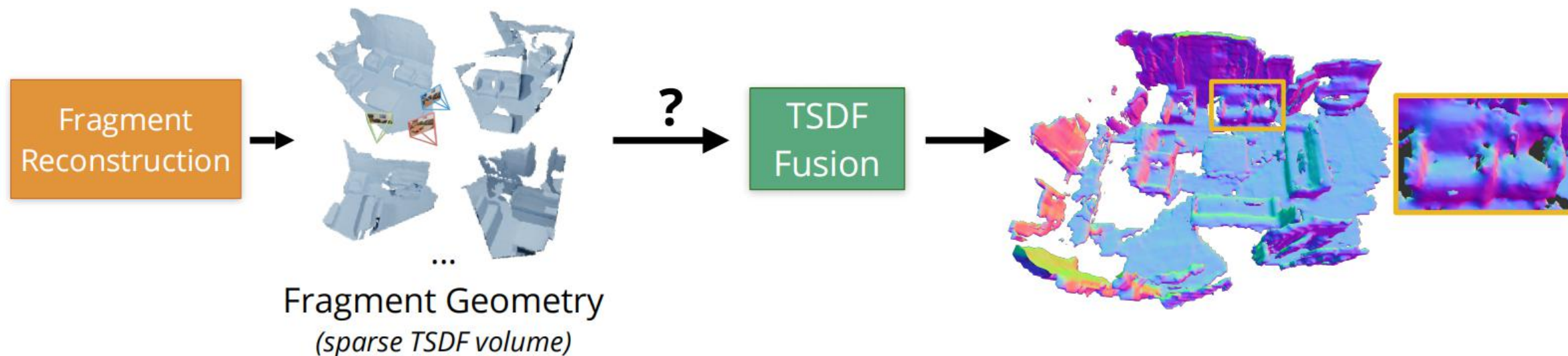
✎ 但是现在咱们只说了一个片段该怎么做，那实际场景中多个片段呢？

三维重建算法

✓ 片段合并

✎ 直接合并到一起，每个片段都挺好，但是整体来说还有挺多瑕疵

✎ 因为每个片段都单独做的，他们没交流啊，所以配合不咋地



三维重建算法

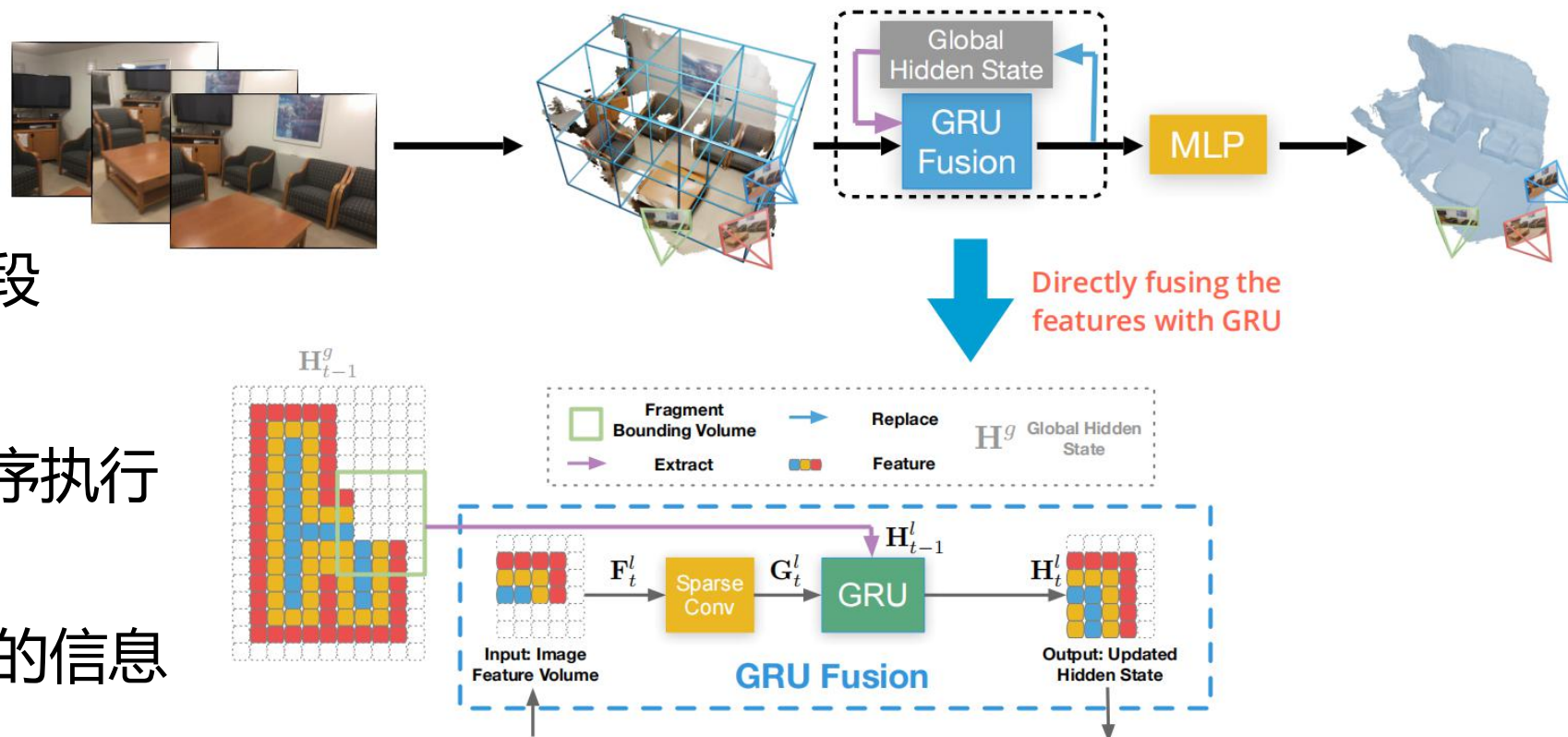
✓ 整体融合思路:

✎ 类似RNN的思想

✎ 迭代式融合输入片段

✎ 一个个搞定，按顺序执行

✎ GRU就是利用前面的信息



三维重建算法

✓ 效果分析

✎ 全局的效果更好

✎ 传统的是窗口加权

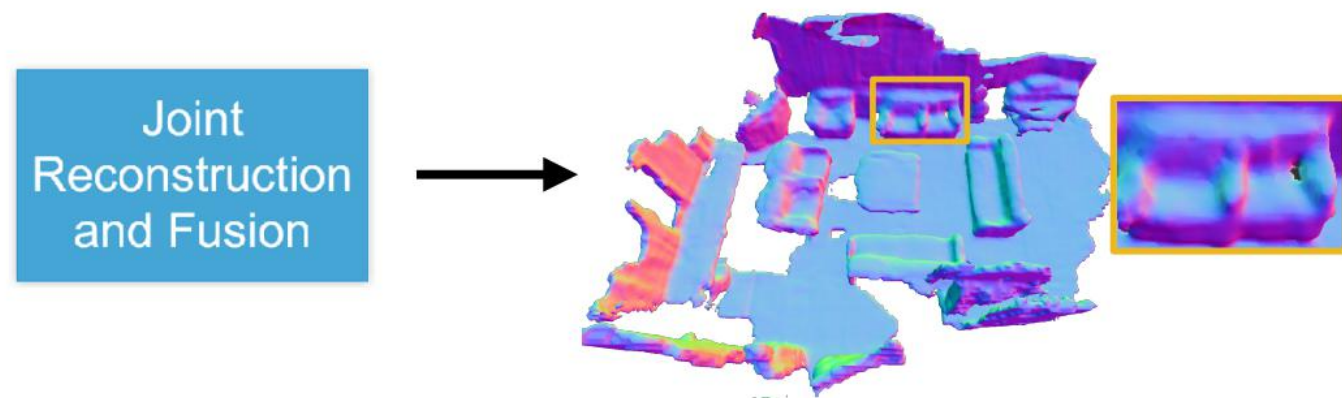
✎ 现在是GRU自动分配

✎ 手动挡换自动挡

Joint reconstruction and fusion



VS



三维重建算法

✓ 整体架构

✎ 粗到细的过程

✎ 这招挺眼熟的

✎ GRU不断融合

✎ 输出最终结果

NeuralRecon Coarse-to-fine architecture

