

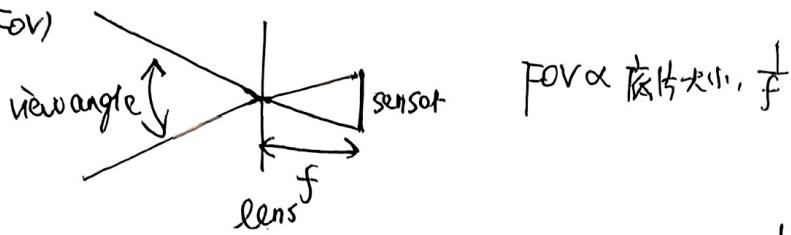
# 计算机视觉导论

## Lecture 2: Image formation.

一. 相机原理:  
Pinhole camera = 小孔成像 aperture. 光圈

$$\text{lens: } \frac{1}{i} + \frac{1}{o} = \frac{1}{f}, \quad m = \frac{hi}{ho} = -\frac{i}{o}$$

### Field of View (FOV)



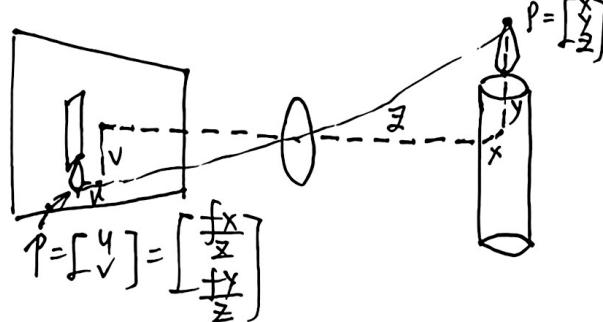
$$F\text{-数: } N = \frac{f}{D}, \quad D: \text{光圈大小.} \quad \text{失焦 (近 \(\rightarrow\) 远) 变大, } b \propto D \propto \frac{1}{N}$$

Depth of Field (DOF): 不失焦的深度 ( $b < \text{pixel size}$ )  $\text{DOF} \propto o, N, \frac{1}{f}$

\* 调节虚焦的方法:  $D \uparrow, f \uparrow, o \downarrow$ , 远离背景. \* 减小并深 DOF.  
大光圈 长焦距 近距离

## 二. 成像:

小光圈向透视投影



$$\text{齐次坐标.} \quad \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} fx \\ fy \\ fz \\ 1 \end{bmatrix} \approx \begin{bmatrix} fx \\ fy \\ fz \end{bmatrix}$$

透视投影的性质: 保持平行, 角度不变.

灭点 vanishing point: 3D 平行线在 2D 图中的交点, 代表消失点, 位置该消失线朝向, 可能在图像外或无穷远

vanishing line: vanishing point 的连线 平面朝向.



透视畸变 distortion: 角度变化

radial distortion: 由~~镜头~~造成的桶形畸变或桶形畸变(直线弯曲)  
广角镜头 长焦镜头

## 三. 光学成像(颜色,亮度).

快门: 控制曝光时间. pixel value =  $\int_{\text{exposure time}}^{\text{intensity}}$  长快门) 小光圈.

Rolling shutter(滚动快门): 逐行曝光. 快速移动相机或物体变化迅速会导致顶部与底部错位.  
解决方法: 稳定相机. 提高快门速度

颜色: RGB (0-255) 三基色

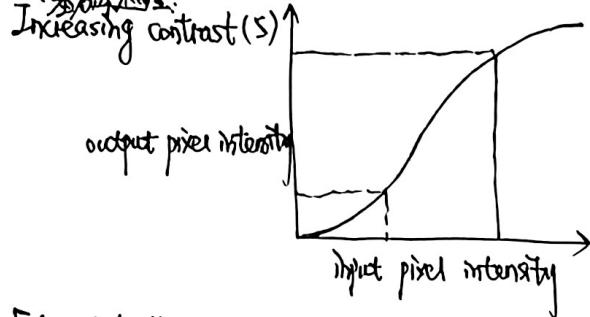
HSV (色调,饱和度,强度)

Boyer 滤波: 用一块传感器记录 RGB 颜色强度 (30% 绿, 25% 红, 25% 蓝)



## Lecture 3: Image processing.

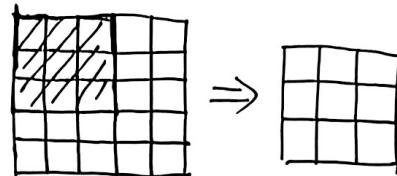
一. 基础处理:



Edge detection:

convolution, dot product for each window

padding: zero/edge value/symmetric



$$\text{Gaussian blur: } f(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{(i^2+j^2)}{2\sigma^2}}$$

$$\text{sharpen: } \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \text{ edge detection: } \begin{bmatrix} 1 & 0 & -1 \\ -1 & 0 & -2 \\ -1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & -2 & -1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

垂直边缘      水平边缘.

二. 采样:

resize 改变分辨率 / 大小

摩尔样  $\Rightarrow$  Aliasing 起样 (Noise 级) 原因: 采样频率太低导致信号在频谱的混叠. Fourier 变换:  $\sin, \cos$  信号加权逼近

卷积定理: Spatial Freq

$$g(x) = f(x) * h(x) \Leftrightarrow G(u) = F(u)H(u)$$

$$g(x) = f(x)h(x) \Leftrightarrow G(u) = F(u) * H(u)$$

(low-pass filter 低通滤波器: 让低频信号通过 例: (模糊) 高斯, 拼值

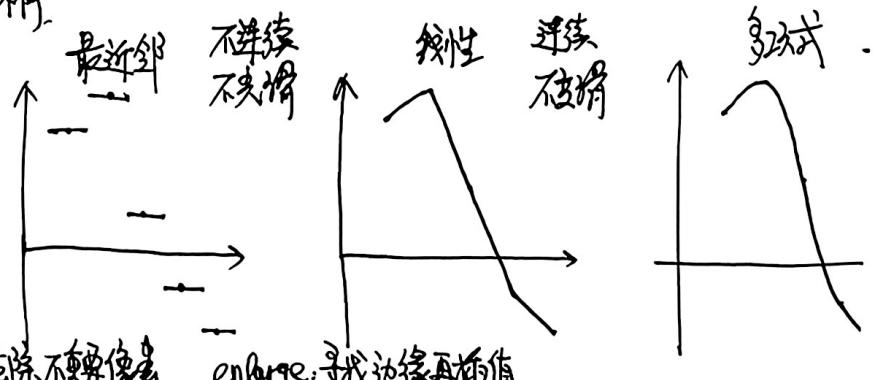
高通 边缘检测.

插样: 失去过低通滤波器去除高斯成分, 再采样  
与(如高斯滤波器)卷积.

三. 上采样:

插值: 最近邻, 线性, 多项式(三次)

三次(二次多项式插值)



改变长宽比: seam-carving: 根据梯度寻找边缘, 去除不需要像素.

enlarge: 寻找边缘再插值.



## Lecture 4: Model fitting and optimization

$$\text{MSE: } \hat{x} = \arg \min_{\mathbf{x}} \sum_i (b_i - \mathbf{a}_i^T \mathbf{x})^2, \quad \mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}. \quad F(\mathbf{x}_k + \Delta \mathbf{x}) = F(\mathbf{x}_k) + \underline{\mathbf{J}_F^T \Delta \mathbf{x}} + \frac{1}{2} \Delta \mathbf{x}^T \underline{\mathbf{H}_F} \Delta \mathbf{x}$$

$$\text{Gradient Descent: } \mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \underline{\mathbf{J}_F^T}, \quad \phi(\alpha) = F(\mathbf{x}_0 + \alpha h), \quad \Delta \phi(\alpha) \downarrow, \alpha \uparrow, \quad \phi(\alpha) > \phi(0), \alpha \downarrow$$

$$\text{Newton: } \mathbf{x}_{k+1} = \mathbf{x}_k - \underline{\mathbf{H}_F^{-1} \mathbf{J}_F^T}, \quad \text{fast convergence, high computation}$$

$$\text{Gauss-Newton: } R(\mathbf{x}) = b - f(\mathbf{x}), \quad \Delta \mathbf{x} = -(\mathbf{J}_R^T \mathbf{J}_R)^{-1} \mathbf{J}_R^T R(\mathbf{x}_k), \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \underline{(\mathbf{J}_R^T \mathbf{J}_R)^{-1} \mathbf{J}_R^T}$$

$$\text{LM: } \mathbf{x}_{k+1} = \mathbf{x}_k - \underline{(\mathbf{J}_R^T \mathbf{J}_R + \lambda I)^{-1} \mathbf{J}_R^T} \quad \text{estimate } \mathbf{H}_F \text{ as } \underline{\mathbf{J}_R^T \mathbf{J}_R} \text{ (must be non-singular)}$$

robust estimate, 去除 outlier.

1. 置信度 loss func:  $\ell_1$  loss / Huber loss

2. RANSAC: 找出直线:

Iterate:

1. 随机采样两个点拟合一条直线
2. 投票(计算直线一定范围内点数)
3. 选择票数最高的模型。

对称估计与不对称估计:  $\ell_1/\ell_2$  正则化。

## Lecture 5: Image Matching and Motion Estimation.

### 1. 图像匹配.

1. 检测: feature points.

Harris corner detector: (1). 计算各点的协差矩阵.

(PCA) (2). 计算特征值  $\lambda_1, \lambda_2$

(3). 分类  $\lambda_1, \lambda_2$  都小  $\rightarrow$  平坦;  $\lambda_1$  远大于  $\lambda_2 \rightarrow$  边缘;  $\lambda_1, \lambda_2$  都大  $\rightarrow$  角点.  $f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$

性质: 高度偏移不变, 缩放不变, 平移, 旋转: 不变, 错误: 低噪 (检测到坏点: 选择合适的尺度)

尝试不同的窗口大小, 选择响应最大的窗口。

Blob detector: 5 Harris 有同样的编程实现问题.

$$\log: \nabla f * g = f \nabla^2 g, \quad \log(\log \text{响应}): \nabla^2 G \approx G_{xx} - G_{yy}.$$

### 2. 描述子: 特征点的描述向量

SIFT (Scale-Invariant Feature Transform): 梯度方向直方图

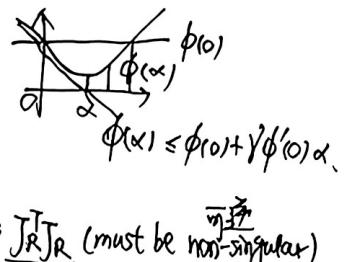
旋转不变性: 通过归一化; 角度不变性: 相比较 Harris detector.

### 3. 匹配:

最近邻:  $\|f_1 - f_2\|$ .

重叠度量 (去除歧义匹配): ratio test:  $\frac{\|f_1 - f_2\|}{\|f_1 - f_3\|}$  best match  
second best  
越大, 异常性越强.

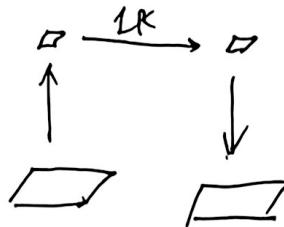
mutual nearest neighbor:  $f_2$  is the nearest neighbor of  $f_1$  in  $I_2$ .  
 $f_1$  —————  $f_2$  in  $I_1$ .



## 二、运动估计

LK算子：假设 (1) 小运动 (2) 灰度一致 (3) 空间相关  
优点：冗余问题 边缘无关计算.

非小运动 (降低分辨率) coarse-to-fine  
其它运动假设 (梯度).



feature tracking: 跟踪每次的特征点  
optical flow, 检测每帧像素的变化.

## Lecture 6: 图像拼接

求解变换: (1). 仿射变换:  $\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \approx \begin{pmatrix} a & b & tx \\ c & d & ty \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$

自由度: 6 {平移: 2, 缩放: 4, 需要 4 对匹配}

(2). 单应变换:  $\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} \approx \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}, [h_{00} \dots h_{22}] = 1$

自由度: 8 需要 8 对匹配.

拼接过程: 1. 输入 选择相近一组匹配  
2. 特征匹配 计算匹配对数  
3. 用 RANSAC 计算变换  
4. inverse warping

Inverse warping:  $(x, y) = T^{-1}(x, y)$  in  $f(x, y)$ . 利用变换逆矩阵, 求目标图像在原图的对应位置, 插值取颜色 → 处理非整数映射, 高质量插值. 相机图像拼接 (相机旋转 → 因柱上高平移)

## Lecture 7: SfM.

### 一、相机模型

1. 世界变换 (世界  $\rightarrow$  相机):  $X_w = \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \Rightarrow X_c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$

相机的外参: 相机在世界坐标的位置  $C_w$ , 流转  $R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$

$$X_c = R(X_w - C_w) = RX_w + t \Rightarrow \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

$$\tilde{X}_c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = M_{ext} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

### 2. 透镜投影

$$X_c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \Rightarrow X_i = f \cdot \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \\ 1 \end{bmatrix} = f \cdot \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}$$

### 3. 2D 到像素的映射

$m_x, m_y$  分别是  $x, y$  3D 像素密度 (pixels/mm). 由像素坐标  $u = m_x X_c = m_x f \frac{X_c}{Z_c}, v = m_y Y_c = m_y f \frac{Y_c}{Z_c}$ .

$$\begin{bmatrix} u \\ v \end{bmatrix} \approx \begin{bmatrix} f_x & 0 & C_x & 0 \\ 0 & f_y & C_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}, f_x = m_x f, f_y = m_y f.$$

$$= M_{int} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

相机内部参数.

$$P = M_{int} \cdot M_{ext}.$$



一、相机标定 (求解内参数).

标定板匹配.  $X_w = \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} \Rightarrow \underline{\lambda}_i = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$ . 列方程  $\underline{\lambda} = P X_w$  求解  $P$ . 因为  $P = M_{int} \cdot M_{ext}$ .

只说定位问题: 通过图像定位相机位置

利用2D点与3D点的对应求解相机参数.

若已知  $M_{int}$  求解  $M_{ext}$ . 称之为  $P_nP_0$  问题 (Perspective-n-Point)

$P_3P$  算法: 4对对应点.

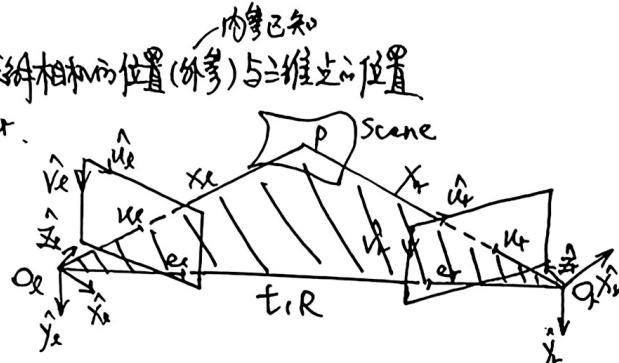
三、SfM.

定义: 给定多张图, 求解相机位置 (外参) 与三维点的位置

1. 对称及  $n$  点 epipolar.

Given 2D 匹配,

求解  $R$  和  $t$ .



2. 相机位姿 (pose) 求解.

步骤: 1. 通过 SIFT 找到至少 8 对匹配点

2. 利用对极约束列方程

3. 求解  $F$

4. 利用内参求解上

5. 分解得到  $R, t$ .

3. Triangulation: 利用 2D 位姿和相机位姿通过光束求解重建 3D 点云.

4. 多图 SfM: 1. 先使用两张图

2. 对每张增加图:

利用已有 3D 点计算新相机位姿

利用三角化求解更多 3D 点

3. Bundle adjustment

5. Bundle adjustment: 对相机参数和 3D 点坐标优化. 重建 3D 点重投影到相机图像上.

目标函数:  $E(P_{proj}, P) = \sum_i \sum_j \| \underline{\lambda}_i^{(i)} - P_{proj}^{(i)} P_j \|_2^2$ . 所有点的重投影误差.

相机 3D 点 相机图像上点的原始位置

相机坐标系原点在另一台相机视场内称为 Epipole.  $e_L, e_R$ . 从相机看去, 另一相机位置的成像点.

Epipolar plane: 由相机原点  $(O_L, O_R)$ , 对极点  $(e_L, e_R)$  与场景点  $P$  组成的平面

Epipolar constraint:  $[x_L \ y_L \ z_L] E \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = 0, E = T \times R$

$x_L^T E x_r = 0$  fundamental matrix

$\Leftrightarrow [\underline{\lambda}_L \ \underline{\lambda}_R] K_L^{-1} E K_R^{-1} \begin{bmatrix} u_r \\ v_r \\ 1 \end{bmatrix} = 0, F = K_L^T E K_R^{-1}$

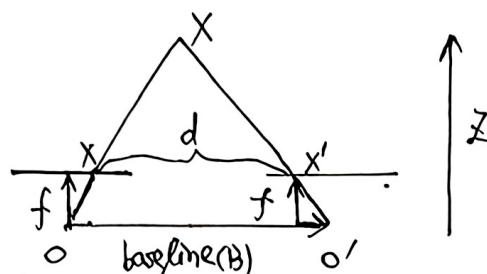


## Lecture 8: 3D Reconstruction.

(条纹)

- 一. 深度估计
  - 深度图  $\{$  active: LiDAR (based on time of flight ToF: 发射接收差)  
passive: 双目立体视觉 (视差)

depth from disparity:



步骤: 1. 相机标定

2. 矫正图像

3. 特征匹配计算视差

4. 求解深度.

$$\text{disparity} = X - X' = \frac{B \cdot f}{z}, \quad \text{depth} = z.$$

Rectification校正: 调整图像使两张图像的 epipolar lines 对齐 (将图像平面重投影到一个公共平面)

baseline: 大baseline精度高, 但过大导致搜索困难, 小baseline会导致较大的深度误差.

## 2. Multi-View Stereo (MVS).

目标: 利用多张图像计算某图像的深度图 error

方法: 对每帧尝试不同的深度值, 找到误差最小的深度.

PatchMatch. 高效匹配

## 3. 3D重建:

步骤: 1. SfM 得到各相机参数

2. 利用MVS计算每张图的深度图.

3. 反投影得到3D点云.

4. 利用泊松重建得到体素表达

5. Marching cubes 抽取网格.

3D表示: 点云  
Poisson重建  $\downarrow$  体素  $\{$  signed occupancy  
Marching cubes  $\{$  (volume) 有向距离场. signed distance  
网格.

## Lecture 9: Deep Learning.

ML: learn a model from data.

Supervised learning: labelled signal

linear classifier for image:  $MSE \times$  cross-entropy + softmax (score  $\rightarrow$  probability)

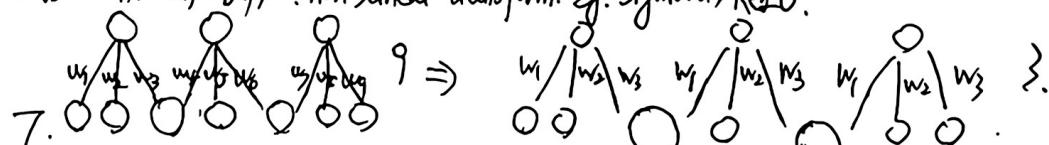
Multi-layer perception:  $f(x) = \sigma(W_1 \sigma(W_2 x + b_1) + b_2)$ ,  $\sigma$ : non-linear transform. Eg. sigmoid, ReLU.

CNN: Locally connected layer

Convolve / weight sharing

$$y = \sigma(X \otimes w + b)$$

padding, stride, side-channel



$32 \times 32 \times 3$  image + 6  $5 \times 5 \times 3$  filter  $\xrightarrow{\text{ReLU}}$   $28 \times 28 \times 6$  feature maps + 10  $5 \times 5 \times 6$  filters  $\xrightarrow{\text{ReLU}}$   $24 \times 24 \times 12 + \dots \rightarrow$

Receptive field: 感受野, 输出的每个元素包含  $K \times K$  个像素的信息 (kernel size =  $K$ )

Fully-connected layer

1 层  $\Rightarrow$  感受野边长:  $1 + L \times (K-1)$ .

Pooling: Max/Average. 聚合信息输出特征图.

Backpropagation, SGD, data split

Regularization, dropout, data augmentation  $\Rightarrow$  prevent overfitting. batch normalization  $\Rightarrow$  faster convergence, prevent gradient problem

b.



## Lecture 10: Recognition.

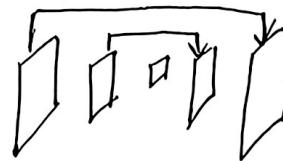
### 一. 像素分割.

图像像素划分(分类)

Fully CNN with per-pixel cross-entropy.

Pooling, stride then unpooling (interpolation)

U-Net: skip-connection



Deeplab: Fully connected network (FCN) + Conditional random field (CRF), SIM

### 二. 物体检测 / 目标检测

输出: 一系列 bounding boxes (label, location, size)      单阶段法: 直接分类每个物体 (大候选区域)

R-CNN, Fast R-CNN, Faster R-CNN, YOLO (You only look once)

(Region of interest)      ~~先找框再分类, 双阶段法~~      ~~FPN: Region proposal~~

三. 实例分割.

物体检测 + 像素分割.

Mask-R-CNN

全景分割: 对背景也作区分.

### ④. 人类姿态估计.

关节点  $\Rightarrow$  热力图表示

多人  $\left\{ \begin{array}{l} \text{自底向上: 失识别人 (Mask R-CNN)} \\ \text{自顶向下: 失识别关节点再组成人 (OpenPose)} \end{array} \right.$

## Lecture 12: Image Synthesis (Computational Photography)

### 一. HDR (High Dynamic Range Imaging).

Exposure = Gain  $\times$  Irradiance  $\times$  Time  $\leftarrow$  shutter speed, aperture, ISO (感光度)      高动态范围可在黑暗环境拍摄但引入噪点.

Dynamic Range 对比度

HDR: 拍摄多张不同曝光下得 LDR 再合并, 合并过程:

1. 寻找合法像素

2. 分配权重

3. weighted average 计算新像素

### 二. Deblurring.

blurring 成因: 物体不在焦平面; 物体运动; 相机不稳.

解决方法: 高取样, 短快门 (大光圈, 高ISO), 硬件 (三角架), 曝光

Non-Blind (已知卷积核) image deconvolution (NBID):  $G = F \otimes H^T \xrightarrow{\text{kernel}} F = \text{IFFT}(\text{FFT}(G) \div \text{FFT}(H))$

Weiner filter: 控制不稳定性

$L_1$  regularization,  $\lambda \|\nabla f\|_1 \Rightarrow$  反模糊.

BID:  $\lambda_2 \|Hf\|_1$ ,

7.

