Low Level: image processing, feature extraction

Mid Level: analyzing local structure, 3D reconstruction

High Level: Object recognition and detection, scene understanding, activity understanding

Canny Detector

Edge: 像素强度沿图像中的一个方向急剧变化，沿其正交方向的强度几乎没有变化

成因：深度不连续；表面取向不连续；表面颜色不连续；光照不连续

检测方案：使用 Gaussian filter 的导数进行卷积来减少噪声并完成检测

Gaussian filter: $g(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp -\frac{x^2}{2\sigma^2}$

Non-maximal Suppression: 比较中心像素与梯度方向上的邻居，若最大，则保留

Thresholding and Linking: 对于当前为 Edge 的像素检测与梯度方向垂直方向的两个邻居，对于每一个邻居，若满足以下条件将其标记为 Edge

- 梯度方向与中心像素在同一个仓内
- 梯度大小大于 minVal
- 满足 NMS 条件

Canny 表明，高斯的一阶导数近似于优化信噪比和定位的乘积的算子

Harris Detector

检测 Corner，对于平移和旋转具有不变性，在缩放上没有不变性，流程为：

- 对于图像求导，得到 I_x 和 I_y
- 对于导数求平方，得到 $I_x^2, I_y^2, I_x I_y$
- 使用矩形窗口或者 Gaussian filter（Gaussian filter 可以保持旋转不变性）
- 计算响应函数

$$\theta = g(I_x^2)g(I_y^2) - \left(g(I_x I_y)\right)^2 - \alpha\left(g(I_x^2) + g(I_y^2)\right)^2 - t$$

- 进行 Non-maximal Suppression

图形在 (x, y) 处沿着 (u, v) 方向对应的像素强度差值为

$$E_{(x,y)}(u,v) = g \star (I[x+u, y+v] - I[x,y])^2$$

$$\approx [u,v] g \star \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = R^{-1} \begin{bmatrix} \lambda_1 & \\ & \lambda_2 \end{bmatrix} R$$

如果为 Corner，那么希望 $1/k < \lambda_1/\lambda_2 < k$ 并且 $\lambda_1, \lambda_2 > b$，于是使用 θ 为

$$\theta = \frac{1}{2}(\lambda_1\lambda_2 - \alpha(\lambda_1+\lambda_2)^2) + \frac{1}{2}(\lambda_1\lambda_2 - 2t) = \lambda_1\lambda_2 - \alpha(\lambda_1+\lambda_2)^2 - t$$

$$= \det M - \alpha(\mathrm{Tr}\, M)^2 - t$$

如果为了达到缩放不变，可以对不同尺度做 Laplacian 或者使用不同尺度的 Gaussian

Least Square Method

将线写作 $ax + by = d$，于是有 $E = \sum(ax_i + by_i - d)^2 = Ah = 0$，其中 $A = UDV^{\mathbf{T}}$ 为数据构成的矩阵，h 为模型参数，那么 h 为 V 中的最后一列

RANSAC: 可以处理多种模型回归问题；容易实现，容易计算失败率；在复杂度不高的范围内只能处理适当比例的离群点；现实问题中离群点的出现概率比较高

Hough Transform: 不如 RANSAC robust；可以处理高比例的离群点

Gradient Descent

Batch Gradient Descent: 使用训练集中的全部数据标签对去计算 gradient

非常慢；容易陷入局部最小值

Stochastic Gradient Descent: 训练集中随机采样大小为 N 的 batch 计算平均梯度

快； loss 函数对于参数而言可能有很高的条件数，沿浅层维度的进展非常缓慢，而沿陡峭的方向则会出现抖动；局部最小值或鞍点处，零梯度并被卡住；梯度可能有噪音

Activate Function

ReLU: 减少了梯度消失的可能性，不仅在输入增加时不会消失，而且梯度的最大值是 1，所以在层变多时不会使得梯度快速累计下降到零；稀疏；快速

Layer

Convolution: 连接稀疏；参数共享；具有平移等变性；过滤出图像的重要特征来减少图像中的 "噪音" $\text{output} = (\text{input} + 2P - F)/\text{stride} + 1$

Pooling: 对小的平移和旋转具有不变性；大幅减少进入下一层的图像的大小；增强某些特征，（取最大值可以突出图像的边缘），减少大小同时仍然保持图像的主体内容

Batch Normalization: 插在全连接层或卷积层之后，在激活函数之前；scale 和 shift 是可训练的参数，这样每个 batchnorm 层都能为自己找到最佳参数，从而可以移动和缩放归一化的值，得到最佳预测；使得每一维数据对结果的影响是相同的，并平稳地进行到最小值，对初始化更加稳健；对于像 Sigmoid 激活函数而言，可以适当解决梯度消失问题；在 mini-batch 中计算均值方差，因此会带来一些较小的噪声，在神经网络中添加随机噪声可以带来正则化的效果

Loss

Sigmoid: $y = 1/(1+e^{-x})$ $y(1-y)$

NLL: $-\sum y^i \log h_\theta(x^i) + (1-y^i)\log\left(1 - h_\theta(x^i)\right)$

Cross Entropy: $-\sum P(x)\log Q(x)$

Initialization

Xavier: $\mathrm{var}\, w_i = 1/D_{\mathrm{in}}$ 保证输入和输出的方差不变

He: $\mathrm{var}\, w_i = 2/D_{\mathrm{in}}$ 使用 ReLU 激活函数时舍弃了 < 0 部分的值

Optimization

learning rate: batch size 增大 N 倍时，初始 learning rate 也要对应增大 N 倍

Linear Warmup: 过高的初始学习率会使得 loss 炸掉，可以在初期线性地增大

Data Augmentation: Scaling, Cropping, Flipping,

Padding, Rotation, Translation, Affine transformation, Brightness, Contrast, Saturation, Hue; 减少数据的过度拟合和产生数据的可变性；提高模型的泛化能力；帮助解决分类中的类不平衡问题

Dropout: 必须对每个神经元的激活进行缩放；迫使网络有一个冗余的表示，防止特征的共同适应

Bottleneck Residual Block: 减少了参数和矩阵乘法的数量；残余块尽可能薄，以增加深度，并有更少的参数

Skip link: 使得从输入到输出成为捷径；协助最后的分割；避免高层网络记忆过多内容

Bottleneck: 不需要记忆整个图像，只提供全局背景；大的感受野并提供全局背景；摆脱多余的信息；降低计算成本

U-Net: 随着编码器的深度变深，原始图像的高层次特征被提取出来，而相应的解码器块刚刚开始恢复；另一方面，编码器最早的块提取了低层次的特征，但连接的匹配解码器块是最接近预测的块。换句话说，U-Net 连接的低级特征靠近预测层，而连接的高级特征远离预测层。这是一套单一的编码器–解码器架构不可避免的限制

Calibrate Camera

对于一个相机而言，成像过程为

$$\boldsymbol{P}' = \begin{bmatrix} \alpha & -\alpha\cot\theta & c_x & 0 \\ 0 & \beta/\sin\theta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \boldsymbol{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{M}\boldsymbol{P}_w$$

其中 $\mathbf{R}_{3\times3} = \mathbf{R}_x(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_z(\gamma) = [\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{r}_3]^{\mathbf{T}}, \boldsymbol{T}_{3\times1} = [T_x \quad T_y \quad T_z]^T, \mathbf{M} = [\boldsymbol{m}_1 \quad \boldsymbol{m}_2 \quad \boldsymbol{m}_3]^{\mathbf{T}}$

由已知信息可以得到对于每一个点的方程 $u_i(\boldsymbol{m}_3\boldsymbol{P}_i) - \boldsymbol{m}_1\boldsymbol{P}_i = 0, v_i(\boldsymbol{m}_3\boldsymbol{P}_i) - \boldsymbol{m}_2\boldsymbol{P}_i = 0$

通过 SVD 分解可以得到结果为 $\mathbf{M} = [\mathbf{A} \quad \boldsymbol{b}]$，其中 $\mathbf{A} = [\boldsymbol{a}_1 \quad \boldsymbol{a}_2 \quad \boldsymbol{a}_3]^{\mathbf{T}}$，得到相机参数

$$\rho = \pm\frac{1}{|\boldsymbol{a}_3|}, \quad c_x = \rho^2(\boldsymbol{a}_1\cdot\boldsymbol{a}_3), \quad c_y = \frac{(\boldsymbol{a}_1\times\boldsymbol{a}_3)\cdot(\boldsymbol{a}_2\times\boldsymbol{a}_3)}{|\boldsymbol{a}_1\times\boldsymbol{a}_3||\boldsymbol{a}_2\times\boldsymbol{a}_3|},$$

$$\alpha = \rho^2|\boldsymbol{a}_1\times\boldsymbol{a}_3|\sin\theta$$

$$\beta = \rho^2|\boldsymbol{a}_2\times\boldsymbol{a}_3|\sin\theta, \quad \boldsymbol{r}_1 = \frac{\boldsymbol{a}_2\times\boldsymbol{a}_3}{|\boldsymbol{a}_2\times\boldsymbol{a}_3|}, \quad \boldsymbol{r}_3 = \pm\frac{\boldsymbol{a}_3}{|\boldsymbol{a}_3|},$$

$$\boldsymbol{r}_2 = \boldsymbol{r}_3\times\boldsymbol{r}_1, \quad \boldsymbol{T} = \rho\mathbf{K}^{-1}\boldsymbol{b}$$

对于双目视点而言，视差为 $u - u' = Bf/z$，B 为相机位置之间的距离，f 为焦距

挑战: Occlusions, Fore shortening, Brightness, Homogeneous regions, Repetitive patterns

CONSIDERATIONS	STEREO VISION	STRUCTURED-LIGHT	TIME-OF-FLIGHT (TOF)
Software Complexity	High	Medium	Low
Material Cost	Low	High	Medium
Compactness	Low	High	Low
Response Time	Medium	Slow	Fast
Depth Accuracy	Low	High	Medium Quickly improving!
Low-Light Performance	Weak	Good	Good
Bright-Light Performance	Good	Weak	Good
Power Consumption	Low	Medium	Scalable
Range	Limited	Scalable	Scalable

Point Cloud

Uniform Sampling: 在 $[0,1]$ 内对于 a_1, a_2 均匀采样，如果 $a_1 + a_2 \le 1$，$\boldsymbol{x} = a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + (1-a_1-a_2)\boldsymbol{v}_3$，否则 $\boldsymbol{x} = (1-a_1)\boldsymbol{v}_1 + (1-a_2)\boldsymbol{v}_2 + (a_1+a_2-1)\boldsymbol{v}_3$

Normal Sampling: 从 $U(0,1)$ 中采样 r_1, r_2，$\boldsymbol{x} = (1-\sqrt{r_1})\boldsymbol{v}_1 + \sqrt{r_1}(1-r_2)\boldsymbol{v}_2 + \sqrt{r_1}r_2\boldsymbol{v}_3$

Iterative Furthest Point Sampling: 通过快速方法对形状进行过采样；迭代选择一个与所选点距离最大的粒子

Chamfer distance: $d_{CD}(S_1,S_2) = \sum_{x\in S_1}\min_{y\in S_2}\|x-y\|_2 + \sum_{y\in S_2}\min_{x\in S_1}\|x-y\|_2$；最接近的距离之和；对采样不敏感

Earth Mover's distance: $d_{EMD}(S_1,S_2) = \min_{\phi:S_1\to S_2}\sum_{x\in S_1}\|x-\phi(x)\|_2$；匹配的最接近距离的总和；对采样很敏感

Local Embedding: 点的特征和全局特征组合在一起；更轻量和迅速；对于点的缺失更加 robust

PointNet: 没有点附近的局部上下文；全局特征依赖于绝对坐标，难以对于未见过的场景泛化

PointNet++: 在局部区域递归应用 PointNet（最远点采样+分组+PointNet）；分层特征学习；局部平移不变性；轮换不变性

Voxel

Sparse Conv:

+: kernel 为空间各向异性的

+: 对索引和邻接查询更有效

+: 适用于大规模场景

+: 平移等变性

−: 有限的分辨率

Point cloud networks:

+: 分辨率高

+: 更容易使用，可以作为快速尝试的第一选择

−: 性能略低

−: 使用最远点采样和邻域球查询的时候会更慢