

图像信息处理复习笔记

复习建议

兴趣选手：如果对这门课感兴趣想要好好学，除了复习笔记，还建议浏览老师的PPT（知识密度不高，比较费时，所以只需要浏览），然后上网搜索PPT上提到的相关技术的论文精读，理解一些标志性的图像处理技术。

求过选手：每一学期的考卷都很相似，可以在CC98上搜历年卷和回忆卷针对性复习；考前一周左右开始复习：先阅读笔记，再浏览历年卷尝试回答，最后对不会的问题查漏补缺，每天2小时就绰绰有余。由于期末考不考写代码，理论题和计算题占比大概8:2，而且是闭卷考，建议实在理解不了的地方就死记硬背。

其他：宋老师是火隐忍者同好，考试可以带计算器，作业实在来不及可以期末补交，不会因为延期而扣分。

概论

- 人 > 60%的信息由视觉获取
- 图像信息处理是计算机图形学和计算机视觉的基础
- 图像采集方式有：可见光、X光、红外线（常用于黑暗环境）、超声波（B超）
- 显色原理有加色法（RGB）和减色法（CMY）；加色法的三原色RGB加起来是白色，减色法的三原色CMY加起来是黑色。CMYK常用于印刷品，因为印刷品不发光，且CMY混合的黑色不纯，要额外加K（黑色）通道
- 图像信息处理的应用：图像存储和压缩、色彩空间转换、光圈矫正、颜色平衡、图像缝合和全景图、图像变形/Warping、图像融合/Morphing、图像增强、图像分辨率提升、图像修复、图像加水印、图像去抖动、图像去雾、滤镜和风格化、一些计算机视觉技术（图像识别、场景解析）等等
- 图像在存储中是一个二维数组，横轴坐标和纵轴坐标确定一个像素：每个通道约定俗成分为255个灰度级，因此灰度图每像素占8bit，彩色图每像素占24bit，每像素按照RGB或GBR的顺序存储；带alpha通道的彩色图占32bit。
- 图像编码方式分为不压缩（BMP）、无损压缩（PNG）和有损压缩（JPEG等）

BMP格式

- BMP的文件结构分为4部分
 - BITMAP FILE HEADER：文件头
 - bf Type：文件类型，必须=="BM"来说明这是位图
 - bf Size：整个BMP文件大小（字节）
 - bf Reserved1：必须==0
 - bf Reserved2：必须==0
 - bf Offbits：从文件开始（文件头）到图像数据（BYTE）有多少个字节的偏差
 - BITMAP INFO HEADER：信息头
 - bi Size：信息头占的字节数
 - bi Height：图像的高（注，biHeight为正时图像是倒着放的）
 - bi Width：图像的宽
 - bi Planes：必须==1
 - bi BitCount：每像素的bit数，可以是1（二值图）/8（灰度图）/16（伪彩色）/24（真彩色）/32（带透明通道）等
 - bi Compression：必须==BI_RGB
 - bi SIZelImage：一般==0，有压缩时才不为0
 - bi XPelsPerMeter：x方向每米的像素值
 - bi YPelsPerMeter：y方向每米的像素值
 - RGB QUAD：相当于调色板，里面可以有N个颜色，总共占N*4 bytes（R、G、B、O每个通道占8bit），表示RGB相对强度
 - BYTE：图像数据信息（字节表示）：必须存成4的倍数字节，不足的在后面补0

成像原理和成像设备

- 成像原理：最早是小孔成像（墨子或古希腊人提出），后来根据小孔成像发明了暗室内

- 照相机由来：小孔成像+透镜成像。光圈（小孔）小了画面变暗，光圈大了画面变糊，太小也会因为发生衍射而变糊。透镜用来控制聚焦，使得前景清晰背景模糊。小光圈可以让聚焦距离更长（因为光更加平行）
- 成像器具：胶片、CCD、CMOS、人眼
- CCD（电耦合元件）：由感光二极管、放大器、AD转换器、DSP（数字信号处理器）和存储卡组成。AD转换器可以把模拟信号转换成数字信号，把连续的信号通过采样变成离散的信号。相机的ISO表示感光度（对光线的敏感程度），一般ISO越高曝光时间越短
- CCD感光的步骤：1.光到达CCD后感光二极管发出电子信号（模拟）；2.通过控制电路将电子信号传递至放大器放大；3.放大和滤波后再通过AD转换器把模拟信号转成数字信号；4.数字信号经过DSP处理（包括颜色校正和白平衡校正）编码成为图像；5.存储
- 人眼：可感知波长为390nm-780nm的光，分辨120-180种不同颜色
 - Retina（视网膜）上有两种细胞：视杆细胞和视锥细胞，视杆细胞可分辨光，视锥细胞可分辨颜色
 - 视锥细胞又分为红、绿、蓝三种，分别感知R、G、B光的三原色
 - 视杆细胞的数量多于视锥细胞，因此人眼对亮度的变化比对色彩的变化更敏感
 - 人对色相、饱和度、亮度的感知顺序为：色相->饱和度->明度
- 韦伯定律/Weber's Law：在正常灰度范围内， $\Delta I/I$ 为常数K，一般在0.01-0.02之间。这个定律还用于其他感官，一般来说，要使一个刺激的变化能够被感觉到，这个变化必须达到刺激强度的一定比例，而这个比例是一个常数，不随刺激强度的变化而变化

色彩空间

- 颜色模型（色彩空间）可分为与设备有关和与设备无关两类：
 - 与设备有关：RGB、CMY、HSV
 - 与设备无关：CIE XYZ、CIE Lab、CIE YUV
- CIE Primary Color Space/CIE XYZ：它是由国际照明委员会（CIE）在1931年定义的，用三种主要颜色（红、绿、蓝）来描述人眼能感知的所有颜色。它使用了CIE xy色度图，表示颜色的色度信息，但不包含亮度信息，CIE xy色度图的边界是人眼能感知的最大色域
- RGB：光的三原色，加色法，是CIE Primary Color Space/CIE XYZ的子集
- CMY：颜色三原色，减色法，一般用于印刷品
- HSV：色相、饱和度、明度；是一种符合人的视觉感知的颜色定义；在三维上可视化是一个六棱锥；其中H用角度来表示，Red是0°，Green是120°，Blue是240°
- CIE颜色模型的作用：用于打通不同设备之间的颜色差异，减小色差
 - CIE XYZ：XYZ表示该颜色在CIE xy色度图中的坐标位置
 - CIE Lab：把亮度L独立开来，a表示从R到G，b表示从Y到B
 - CIE YUV：用于电视等视频图像处理和传输，可以由一个转换矩阵和RGB空间互转

JPEG格式

- 后缀有：JPEG、JPG、JPE、JFIF
- JPEG是一种有损压缩格式，允许用户在图像质量和大小之间权衡。它的压缩过程如下：
 - 将图像从RGB空间转换到YUV空间，利用人眼对亮度（Y）比色度（U和V）更敏感的特性对色度进行降采样，减少色度的数据量
 - 将图像分割为8x8的小块，对每个小块进行离散余弦变换（DCT），将图像的时域信息转换为频域信息，也就是不同频率的余弦函数的组合
 - 根据人眼对低频信息比高频信息更敏感的特性，对DCT的结果进行量化，也就是用一个近似的值来表示一个范围内的值，减少数据的精度，同时去除一些高频信息
 - 对量化后的数据进行编码（DCPM差分脉编码调制和RLE行程长度编码），通常使用Zig-Zag扫描和霍夫曼编码的组合，利用数据的稀疏性和重复性，进一步减少数据的位数
- JPEG的缺点是不适用于压缩线条、文本、符号等图片，会导致缺陷
- 其他有损压缩格式：
 - TIFF：常用于扫描仪、CAD、地图，高度可扩展的、灵活压缩的、相对开放的格式，由Adobe提出
 - GIF：互联网图像交换格式，可以存好几帧的数据。颜色表分为全局颜色表和局部颜色表（某一帧用到的颜色），支持嵌入文本

图像二值化和二值图像

- 二值图像：每一个像素非黑即白的图像；好处是节约内存、高效、可以迁移到灰度图；坏处是应用领域受限，无法表现物体的立体感，无法调节对比度
- 图像二值化操作：遍历灰度图的每个像素，把灰度值小于threshold的像素置成黑色，反之置成白色
- 要如何确定二值化的阈值？——大津算法/最大类间方差法
 - 核心思想：通过寻找一个阈值，将图像的像素分为前景和背景，使得两个类别之间的差异（类间方差）最大
- 最大类间方差法的执行：
 - 统计图像的灰度直方图，得到每个灰度级的像素的比例
 - 遍历所有可能的阈值（0到255），计算该阈值下得到的前景和背景的像素比例和平均灰度，从而计算类间方差
 - $\sigma_{inter}^2 = w_F(\mu_F - \mu_T)^2 + w_B(\mu_B - \mu_T)^2 = w_F * w_B * (\mu_F - \mu_B)^2$
 - w_F 表示前景像素占图像的比例， w_B 表示背景像素占图像的比例，两者相加为1
 - μ_F 表示前景像素的平均灰度， μ_B 表示背景像素的平均灰度， μ_T 表示整张图像的平均灰度，其中 $w_F * \mu_F + w_B * \mu_B = \mu_T$
 - 根据类间方差的最大值确定最佳阈值，也就是使前景和背景的平均灰度差异最大的阈值
- 二值化可以只用一个全局阈值，也可以使用一个窗口，每次只计算窗口内部分的局部阈值
- 二值图像的应用：绿幕、卫星遥感图分析、（早期）人脸检测
- 二值化后的问题：结构断裂、碎屑噪声

形态学与二值图像运算操作

- 形态学/Morphology：形态学的核心思想是检测并提取出具有某一形状的结构元素，并剔除不想要的结构元素
- 形态学的四种操作：
 - 膨胀：扩大前景/白色区域，遍历原图每个像素并放上结构元，只要结构元和原图的白色部分有交集，新图中该位置的像素就是白；膨胀的作用是把背景中和前景相连的部分归入前景，并填上前景中的孔洞
 - 腐蚀：缩小前景/白色区域，遍历原图每个像素并放上结构元，只有结构元完全被白色部分包裹，新图中该位置的像素才是白
 - 腐蚀操作可用于二值图像边缘提取：A-erode(A)即可得到边缘
 - 开运算：先腐蚀后膨胀，可以去除细小物体（白色噪点），把藕断丝连的物体分离，并平滑大的物体的边缘并保留面积
 - 闭运算：先膨胀后腐蚀，可以填补小的孔洞，使藕断丝连的物体连接，平滑边缘并保留面积
- 应用：指纹处理是先开运算去噪，然后用闭运算加粗指纹部分

图像对数增强和伽马矫正

- HDR（高动态范围）图像的压缩：把人眼感知到的自然界中 10^8 量级的亮度压缩到 2^8 个灰度级，同时保留细节。处理HDR图像可以用伽马矫正和图像对数增强的方法，让LDR设备可以正常显示HDR图像
- 伽马矫正：根据人眼对亮度的非线性感知，调整图像的亮度和对比度，从而增强图像中暗部的细节，公式为 $V_{out} = AV_{in}^\gamma$ ，其中A是一个常数，通常为1，而 γ 为解码伽马值和加码伽马值，前者通常为2.2，后者通常为-2.2
- 图像对数增强：让图像整体变亮，对于单个像素的处理公式是 $y = c \log(1 + x)$ ；一般 $c = 1/\log(L_{max} + 1)$ ，遍历每个像素应用该公式可以让各像素值的灰度值更平滑

灰度直方图和均衡化操作

- 直方图/Histogram：横坐标是灰度值，纵坐标是每种灰度值像素个数占图像像素总个数的比例
- 直方图越均衡（灰度分布越均匀），图像看起来越尖锐
- 直方图无法表现图像内容，不同图像可能有相同的直方图
- 直方图均衡化：通过改变图像的灰度分布直方图来增强图像的对比度（均衡化前后亮度的变化取决于图像的原始直方图分布。如果直方图集中在较低的灰度区域，那么均衡化后图像的亮度会增加，因为均衡化会扩展图像的灰度动态范围，使得较暗的像素变得更亮，反之亦然）。直方图均衡化的目的是使得图像的直方图分布更加均匀，从而扩展图像的灰度动态范围，使图像中的细节更加清晰
- 公式：

- 若灰度是连续的: $T(r) = L \int_0^r P(r)dr$, 最后得到的直方图是没有起伏的矩形, 若灰度要标准化, 则不用乘上L
- 若灰度是离散的 (绝大部分情况): $S_k = T(r_k) = \frac{1}{n} \sum_{i=1}^k n_i$
 - $S_0 = P(r_0)$, $S_1 = \frac{n_0+n_1}{n} = S_0 + P(r_1)$, 因此, 一级一级灰度算下去可以算出所有的 S_k
 - 再把分数 S_k 量化 (四舍五入) 成最近的灰度级
- 由于离散的灰度直方图在算完之后还要量化, 因此最终得到的直方图不是没有起伏的矩形, 并且概率低的灰度级容易被归并到邻近的概率高的灰度级
- 直方图均衡化还可以做直方图匹配: 简单来说就是把图像A的直方图弄成和图像B的直方图一样
 - 分别得到A和B的直方图均衡化函数 $T_A(r)$ 和 $T_B(r)$, 这个函数可以对图像应用直方图均衡化
 - 先对A应用A的均衡化函数 $T_A(r)$, 得到均衡化后的图像, 再对这个图像运用B的反均衡化函数 $T_B^{-1}(r)$ 即可; 因此 $T_{A \rightarrow B}(r) = T_B^{-1}(T_A(r))$; 如果图像A和B的灰度是离散的, 分别计算两个映射表格再对应即可
- 直方图的其他操作: 直方图量化减少细节、直方图线性变换 ($k < 1$ 对比度下降, 反之上升)、直方图对数变换、直方图几何变换

简单几何变换

- 一般用变换矩阵实现图像的简单几何变换。
 - Translation: 平移变换的公式为 $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, 其中 (x, y) 是原图像的坐标, (x', y') 是变换后的坐标
 - Rotation: 旋转变换的公式为 $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, 其中 (x, y) 是原图像的坐标, (x', y') 是变换后的坐标, θ 是旋转的角度。如果旋转的中心不是原点, 那么需要先将图像平移到原点, 再进行旋转, 然后再平移回原来的位置。这可以通过矩阵的乘法来实现,

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \text{ 即}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_c \\ 0 & 1 & y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \text{ 其中 } (x_c, y_c) \text{ 是旋转的中心坐标}$$
 - Scale: 缩放变换的公式为 $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, 其中 (x, y) 是原图像的坐标, (x', y') 是变换后的坐标, f_x 和 f_y 是水平和垂直方向的缩放因子
 - Shear: 错切是将图像沿着水平或垂直方向拉伸或压缩, 改变图像的形状, 但不改变图像的面积。错切变换可以用一个 3×3 的矩阵来表示, 其中第一行和第二行分别表示水平和垂直方向的错切系数, 第三行为 $[0, 0, 1]$ 。错切变换的公式为 $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & k_x & 0 \\ k_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, 其中 (x, y) 是原图像的坐标, (x', y') 是变换后的坐标, k_x 和 k_y 是水平和垂直方向的错切系数。
 - Mirror: 镜像变换的公式为 $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$, 其中 (x, y) 是原图像的坐标, (x', y') 是变换后的坐标, 如果是水平方向的镜像, 那么第一行的翻转系数为-1, 第二行的翻转系数为1; 如果是垂直方向的镜像, 那么第一行的翻转系数为1, 第二行的翻转系数为-1
- Rotation中出现的空洞问题的解决方法: interpolation/插值
 - 邻近插值法: 将空洞的像素值直接设置为离它在原图中的位置最近的已知像素值 (用round函数四舍五入得到的整数坐标就是最近的邻居的坐标), 也可通过几个邻居平均得到。优点是计算简单, 缺点是会造成图像的锯齿和马赛克
 - 线性插值法: 用一条直线来连接两个已知像素值, 然后根据直线方程来计算空洞的像素值, 也就是说, 使用空洞在原图中的位置附近的2个邻居的像素值的加权平均来填充空洞。这种方法的优点是比邻近插值法更平滑, 缺点是会造成图像的模糊或失真。

- 双线性插值法：先在水平方向和垂直方向分别进行线性插值，然后再进行一次线性插值，也就是说，使用4个邻居的像素值的加权平均来填充空洞。这种方法的优点是比线性插值法更平滑。这种方法用到4个已知像素，它们组成的田字格包含了空洞像素在原图中的位置对应的像素方块)
- 径向基函数法：用一个函数来拟合所有已知像素值，然后根据函数值来计算空洞的像素值，也就是说，使用所有邻居的像素值的加权平均来填充空洞。径向基函数填补空洞的公式可以表示为：

$$f(\mathbf{x}) = \sum_{i=1}^n w_i \phi(|\mathbf{x} - \mathbf{x}_i|)$$

其中， $f(\mathbf{x})$ 是插值后的像素值， n 是已知像素值的个数， \mathbf{x}_i 是已知像素值的坐标， w_i 为权重， ϕ 是径向基函数，可以取高斯函数、线性函数、薄板样条函数等。一般权重 $w_i = G(c_i) / \sum \phi(|x - c_i|)$

Morph/图像融合

- Image morph可以将一幅图像平滑地过渡成另一幅图像，基本步骤是：
 - 建立图像之间的对应关系，即确定哪些像素点或区域在不同图像中是相似的
 - 根据对应关系，对图像进行变形，使得相似的像素点或区域在空间位置上重合
 - 根据变形后的图像，进行颜色插值，使得相似的像素点或区域在颜色上也相近
- 根据建立对应关系的方式，分为基于线段的morph和基于网格的morph：
 - 基于线段的morph：在图像中选取一些有代表性的线段对作为图像之间的特征，然后根据线段对之间的变换矩阵，计算其他像素点的对应关系和变形方式。由于线段对不止有一对，对源图像中的每个像素要计算它与每个线段对的距离作为影响权重，然后根据权重和变换矩阵，计算它在目标图像中的对应位置
 - 基于网格的morph：将图像划分为一些小的网格单元，然后根据网格单元之间的位置和形状变化，计算其他像素点的对应关系和变形方式
- 基于兰伯特光照模型的人脸表情Morph：在人脸图像中选取一些关键点，如眼睛、鼻子、嘴巴等，作为图像之间的特征，然后根据关键点之间的位置和角度变化，计算其他像素点的对应关系和变形方式。根据变形后的图像，进行颜色插值，使得相似的像素点或区域在颜色上也相近。同时，根据兰伯特光照模型，考虑光源的方向和强度，对变形后的图像进行光照调整，使得人脸表面的明暗变化更加自然和真实

卷积和滤波器

- 连续一维卷积/Continuous 1D Convolution：两个一维函数 $f(x)$ 和 $h(x)$ 的卷积 $g(x) = \int_{-\infty}^{+\infty} f(t)h(x-t)dt$ ，其中 x 是任一给定的位置
- 离散一维卷积/Discrete 1D Convolution：把积分改成求和， $g(x) = \frac{1}{M} \sum_{t=0}^{M-1} f(t)h(x-t)$
- 离散二维卷积/Discrete 2D Convolution：用一个窗口/卷积核在图像上平移，框出原图内的若干像素，并求和窗口内的值和像素值的乘积作为卷积结果。常用的卷积核有3x3, 5x5等，一般来说卷积核的宽高为奇数。2D卷积核的一般形式： $g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)f(x+s, y+t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$ ，卷积核的宽是2a+1，高是2b+1。
- 卷积运算符合交换律、分配律、结合律
- 低通滤波器：是一个卷积核，用于图像模糊。常见的是线性平滑滤波器/Mean Filter，是一个3x3的窗口，里面的每一个值都是1/9；可以去除细小物体，核的大小取决于要去除的物体的大小
- 统计滤波器/Statistical Filter：又叫中值滤波器，对窗口内像素的灰度值进行排序，用中位数代替窗口中心的值，可以去噪（去除面积 $< n^2/2$ 的噪点），相比于低通滤波器会减少模糊
- 图像锐化滤波器：目的是突出图像中的边缘和细节，增强图像对比度和清晰度。原理是利用图像的一阶微分或二阶微分来检测图像中的灰度变化，然后根据灰度变化的大小和方向来增强或抑制像素值。
 - 一阶微分： $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right]$ ，其中 $\frac{\partial f}{\partial x}$ 和 $\frac{\partial f}{\partial y}$ 是图像在水平方向和垂直方向的一阶偏导数，也称为水平梯度和垂直梯度。一阶微分滤波器的输出图像可以用原图和滤波后的结果叠加得到，即：

$$M(x, y) = f(x, y) + |\nabla f| = f(x, y) + \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$
 而滤波结果则用梯度的模长来表示。为了提升计算速度用曼哈顿距离来代替欧氏距离，并用差分近似导数：

$$M(x, y) \approx f(x, y) + (|\frac{\partial f}{\partial x}| + |\frac{\partial f}{\partial y}|) * k = f(x, y) + (|f(x+1, y) - f(x)| + |f(x, y+1) - f(x)|) * k$$
 (k为锐化系数)

- 在图像的恒定灰度区域，一阶微分值为零，叠加值为零，因此输出图像不产生效果；在图像的灰度变化的台阶或斜坡处，一阶微分非零，叠加了一个与梯度模长成正比的值，产生锐化效果。
- 一阶微分滤波器的常用算子有Sobel算子、Prewitt算子、Roberts算子等，它们都是用3×3或2×2的模板来近似计算图像的一阶微分。罗伯特算子/Roberts Cross Gradient需要用2×2的模板来近似计算图像的一阶导数，模板的形式如下：

$$G_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

其中， G_x 和 G_y 分别表示水平方向和垂直方向的一阶微分模板。它可以处理具有陡峭边缘的低噪声图像，当图像边缘接近于正45度或负45度时，该算法处理效果理想。但它对边缘的定位不太准确，提取的边缘线条粗，且对噪声敏感，容易产生虚假的边缘。

- 二阶微分： $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$ ，其中 $\frac{\partial^2 f}{\partial x^2}$ 和 $\frac{\partial^2 f}{\partial y^2}$ 是图像在水平方向和垂直方向的二阶偏导数，也称为水平曲率和垂直曲率。二阶微分滤波器的输出图像可以用拉普拉斯算子的值来表示（在图像处理中，因为像素之间最小差距为1，用差分近似导数）：

$$M(x, y) = f(x, y) \pm \nabla^2 f(x, y) = f(x, y) \pm [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)]$$
- 拉普拉斯算子除了考虑上下左右的像素，还可以扩展，考虑4个对角的像素：

$$\nabla^2 f(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 f(x+i, y+j) - 9f(x, y)$$

双边滤波

- 图像平滑：对大规模特征平滑得到结构/structure和模糊效果/blur；对小规模特征平滑得到细节/texture和光晕/halo
- 图像的两大特征：
 - 时域/Space Domain：是指图像在二维平面坐标系上的表示，它由图像的像素值组成，反映了图像的灰度或颜色分布。图像的时域处理是指直接对图像的像素值进行操作
 - 值域/Intensity Domain：是指图像的灰度或颜色的范围，它反映了图像的亮度或饱和度。图像的intensity domain可以用灰度直方图来表示，例如，8位灰度图像的可以用x轴为[0, 255]的直方图表示
- 双边滤波（三大保边滤波之一）：是一种非线性的图像滤波方法，它可以在去除噪声的同时保持图像的边缘和细节。它的原理是结合图像的空间邻近度和像素值相似度，用两个高斯函数来构造滤波器的权重，从而实现对图像的空域和值域的同时考虑。
 - 空间核（像素域）/Space Kernel算出空间权重： $G_s(x) = \exp(-\frac{|x_i-x|^2}{2\sigma_s^2})$ ，在灰度平坦区域Gr很小，因此Gs占主导
 - 范围核（灰度域）/Range Kernel算出值权重： $G_r(x) = \exp(-\frac{|I(x_i)-I(x)|^2}{2\sigma_r^2})$ ，在边缘区域，同侧的Gr相近且远大于另一侧，因此能明显区分边缘，起到保边作用
 - 滤波后的结果为： $I_{\text{filtered}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) * G_s(x) * G_r(x)$ ，其中x是当前像素的坐标，Ω是当前像素的邻域， W_p 是归一化因子。 σ_r 和 σ_s 是两个高斯函数的标准差，用来控制值域和空域的影响程度。归一化因子 W_p 用来保证滤波后的图像的灰度值不超过1，它是权重和 $W_p = \sum_{x_i \in \Omega} G_s(x)G_r(x)$
- 双边滤波器的应用：色调映射、重新着色、材质修改等。色调映射/Tone Mapping是指把HDR照片用于LDR媒介，由于相片等媒介的亮度范围不足以表现真实世界，简单先行压缩会丢失细节。因此采用LargeScale（双边滤波得到）+Detail（residual得到）+Color的叠加得到处理后的图片
- 由于蛮力法实现双边滤波速度很慢，可以用信号处理方法快速逼近，简单来说是用一个线性滤波器对值域进行逼近处理（用增维方法实现值域滤波，将灰度值作为一个新的维度，然后在这个新的空间中进行滤波，这样可以将非线性的问题转化为线性的问题）

导向滤波

- 导向滤波/Guided Filter：也是三大保边滤波器之一。输入一个图像P，通过引导图像I（也可以是A本身），经过滤波后得到图像Q
- 导向滤波的公式可以用最小二乘法推导出来：
 - 基本的思想是，对于每个像素点i，输出图像q的值是一个加权平均值： $q_i = \sum_j W_{ij}(I)p_j$ ，其中p是输入图像，I是导向图像， $W_{ij}(I)$ 是只和I相关的滤波核。
 - 导向滤波假设输出图像q和导向图像I在局部窗口 ω_k 内有线性关系： $q_i = a_k I_i + b_k, \forall i \in \omega_k$ ，其中， a_k 和 b_k 是常数系数，k是窗口的中心点。

- 为了求解这些系数，导向滤波使用了带有正则化项的线性岭回归：

$$E(a_k, b_k) = \sum_{i \in \omega_k} ((a_k I_i + b_k - p_i)^2 + \epsilon a_k^2)$$
，其中 ϵ 是一个很小的正数，用来防止 a_k 过大。通过最小化这个损失函数，可以得到 a_k 和 b_k 的闭式解： $a_k = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon}$ ， $b_k = \bar{p}_k - a_k \mu_k$ ，其中 μ_k 和 σ_k^2 是窗口内导向图像 I 的均值和方差， \bar{p}_k 是窗口内输入图像 p 的均值， $|\omega|$ 是窗口的大小。
- 最后，对于每个像素点 i ，它可能属于多个窗口，所以需要对所有窗口的输出取平均，得到最终的输出图像 q ：

$$q_i = \frac{1}{|\omega|} \sum_{k: i \in \omega_k} (a_k I_i + b_k)$$
。
- 导向滤波相比于双边滤波，具有O(1)时间复杂度，不存在梯度反转问题
- 应用：毛发遮罩/Feathering、去雾/Haze Removal

傅里叶展开

- 任何一个复杂函数都可以写成多个简单函数之和，如泰勒展开
- 傅里叶展开是一种将周期函数用三角函数的和来表示的方法；傅里叶变换是一种将非周期函数用连续的正弦波和余弦波的积分来表示的方法。它们都可以用来分析函数的频率特性（函数包含哪些频率分量），以及不同频率分量的幅度和相位
- 傅里叶展开的基本形式是： $f(t) = a_0 + \sum_{n=1}^{\infty} (a_n \cos(n\omega t) + b_n \sin(n\omega t))$ ，其中 $f(t)$ 是周期为 T 的周期函数， $\omega = \frac{2\pi}{T}$ 是基波角频率， a_0, a_n, b_n 是傅里叶系数，可以通过如下公式求得： $a_0 = \frac{2}{T} \int_0^T f(t) dt$ ， $a_n = \frac{1}{T} \int_{-T}^T f(t) \cos(n\omega t) dt$ ， $b_n = \frac{2}{T} \int_0^T f(t) \sin(n\omega t) dt$ 。对于一个非周期函数，可以在 $[-\pi, \pi]$ 区间内用傅里叶展开去近似
- 除此之外，根据欧拉公式： $e^{i\theta} = \cos(\theta) + i \sin(\theta)$ ，傅里叶展开还能写成指数形式： $f(t) = \sum_{-\infty}^{\infty} c_n e^{in\omega t}$ ，其中 c_n 是傅里叶系数： $c_n = \frac{1}{T} \int_0^T f(t) e^{-in\omega t} dt$
- 傅里叶变换针对非周期函数，而一个非周期函数可以看做周期无限大的函数。傅里叶变换和逆变换的公式是一种将时域函数和频域函数相互转换的方法。傅里叶变换（e的指数有负号）： $F(i\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$ ；傅里叶逆变换（e的指数没有负号）： $f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(i\omega) e^{i\omega t} d\omega$ ，其中， $f(t)$ 是时域函数， $F(n\omega)$ 是频域函数， i 是虚数单位， ω 是角频率。
- 离散傅里叶变换可用于图像等离散信号，把积分换成求和可得： $F(f(x)) = \sum_{x=0}^{N-1} f(x) e^{-\frac{2\pi i x y}{N}}$ ，离散傅里叶变换的逆变换： $F^{-1}(f(x)) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{\frac{2\pi i x y}{N}}$ ， $f(x)$ 分别是时域函数和频域函数， i 是虚数单位， ω 是角频率。
- 快速傅里叶变换/FFT是一种高效的算法，它可以快速计算序列的离散傅里叶变换/DFT或其逆变换，把时间复杂度从 $O(N^2)$ 变为 $O(N \log N)$ 。FFT的思想是利用DFT的一些数学性质，将一个长度为N的序列的DFT分解为两个长度为N/2的序列的DFT（拆分成一个奇函数和一个偶函数），基于分治法的策略，递归地将子序列分成更短的子序列，然后用旋转因子/twiddle factor进行组合。
- 二维图像上的离散傅里叶变换： $F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi i (\frac{u}{M} x + \frac{v}{N} y)}$ ，这里水平和垂直方向的角频率分别是 u/M 和 v/N

特征对齐

- 特征匹配/Feature Matching是制作全景图的关键：检测两张图中的特征点（flat、edge、coner），找特征点对并对齐。常用的检测方法有基于特征向量和特征值的特征检测：
 - 可以用像素值的平方和差来判断表示某个点的特征： $E(u, v) = \sum_{u, v \in w} [I(x+u, y+v) - I(u, v)]^2$ 。由于

$$I(x+u, y+v) \approx I(x, y) + \frac{\delta I}{\delta x} u + \frac{\delta I}{\delta y} v = I(x, y) + [I'_x I'_y][uv]^T$$
，因此

$$E(u, v) \approx \sum_{u, v \in w} ([I'_x I'_y][uv]^T)^2$$
，把矩阵 $H = \begin{bmatrix} I_x'^2 & I_x' I_y' \\ I_y' I_x' & I_y'^2 \end{bmatrix}$ 的特征向量 X 和特征值 λ 作为该像素点的特征向量和特征值，分别表示朝某个方向变化最大和变化的幅度
 - X_{max} 是朝着这个方向变化最大的方向， X_{min} 反之，用 λ_{max} 和 λ_{min} 可以画出一个特征椭圆，若椭圆很大很圆，说明该点是一个角/corner，朝各个方向的变化幅度都很大，若椭圆很扁，说明该点位于一条边/edge
 - 特征检测的步骤：
 - 计算点(x,y)在水平方向和垂直方向的一阶导数
 - 用两个一阶导数值创建特征矩阵H
 - 计算特征向量和特征值
 - 找到特征值较大的所有点（ $\lambda > threshold$ ）作为该图像的特征点
 - 基于特征向量和特征值的特征检测具有旋转不变性，但不具有缩放不变性

- 除了特征向量和特征值方法，还有一种哈里斯操作/Harris Operator可以用来检测特征：这是一种用于提取图像中的角点的方法，利用一个窗口在图像上滑动，计算窗口内的灰度变化量，如果窗口在任意方向上移动都会引起较大的灰度变化，那么窗口中心就是一个角点。Harris Operator特征检测具有旋转不变性和光照不变性，但不具有尺度不变性

SIFT和SURF特征检测

- SIFT/Scale Invariance Feature Transform特征检测：具有尺度不变性和旋转不变性，可以在不同的图像尺度和角度下匹配相同的物体。SIFT特征检测的主要步骤如下：
 - 尺度空间极值检测：通过使用高斯差分函数来计算并搜索所有尺度上的图像位置（在Space和Scale维度都运用高斯函数），用于识别对尺度和方向不变的潜在关键点。具体来说是用一个16x16的窗口去计算每个像素的边缘朝向，并扔掉弱边缘
 - 关键点定位和方向匹配：通过一个拟合精细的模型在每个候选位置上确定位置和尺度，关键点的选择依赖于它们的稳定程度。创建幸存边缘的朝向直方图，在一个窗口4x4的范围内进行投票，得票最高的朝向就是该窗口的朝向（一个或多个），后续所有对图像数据的操作都是相对于关键点的方向、尺度和位置进行变换，从而提供这些特征的不变性
 - 关键点描述：在每个关键点周围的区域内，以选定的比例计算局部图像梯度，这些梯度被变换成一种表示用于描述关键点。最终输出4x4（每个cell的格子数）x8（八个方向向量的权重向量）=128维向量作为特征
 - 优点是提取的特征具有旋转、缩放、亮度不变性，局部特征具有高度特异性和可解释性，对刚体的表示效果好；缺点是速度慢，对非刚体的表现效果差，在严重变形时容易检测不到特征
- SURF/Speeded Up Robust Features特征检测：SIFT升级版，具有尺度不变性和旋转不变性，可以在不同的图像尺度和角度下匹配相同的物体。SIFT特征检测的主要步骤如下：
 - 计算积分图（该像素点之前的所有像素点的灰度值之和），某个像素的积分图是在其左上方的所有像素的灰度值之和： $S(x, y) = I(x, y) + S(x-1, y) + S(x, y-1) - S(x-1, y-1)$
 - 对积分图应用二阶导数滤波器，找到灰度变化剧烈的地方， $D_{xx} = \begin{bmatrix} 1 & & \\ -2 & & \\ 1 & & \end{bmatrix}$, $D_{xy} = \begin{bmatrix} 1 & -1 & \\ -1 & 1 & \end{bmatrix}$, D_{yy} 同理
 - 利用二阶导构建Hessian矩阵近似： $H(x, y, \sigma) = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix}$ ，这一步是在图像空间和尺度空间中寻找Hessian矩阵行列式的极大值点，作为候选的特征点
 - 用非极大抑制/NMS扔掉一些不明显的特征（比三层中9+8+9个邻居都大的点才保留下来）
 - 根据上面得到的一阶导和二阶导，插值 $H(x) = H + \frac{\delta H^T}{\delta X} X + \frac{1}{2} X^T \frac{\delta^2 H}{\delta X^2} X$ ，精确定位特征点的位置和尺度，以及去除低对比度和边缘响应的特征点
 - 计算特征点的主方向，实现旋转不变性。通过统计特征点邻域内的Haar小波特征的方向和强度，找到最显著的方向
 - 计算特征点的描述子，将特征点邻域划分为4x4的子区域，计算每个子区域内的Haar小波特征的和和绝对值的和，得到64维向量作为特征点的描述子（共输出4x4（格子数）x4（ $\sum dx$ 、 $\sum |dx|$ 、 $\sum dy$ 、 $\sum |dy|$ ）个值）
- 图像特征检测和拼接的流程：
 - 检测关键点
 - 建立SIFT或SURF解释器
 - 匹配解释器（特征差异小于某个阈值时判定为一对特征点）
 - 根据特征点的变换得到变换矩阵，对原图应用几何变换
- 随机样本一致性/RANSAC：先随机选择四个特征点构成变换矩阵，用这个矩阵来计算其他特征点是否对齐（用一个阈值区分内点和外点），用内点和外点的个数作为评分，不断迭代变换矩阵以提高评分

人工智能在特征检测中的应用

- 机器学习中传统的识别方法是用人工指定的特征（非训练、学习而得）+训练而得的分类器（如SVM\HMM等）
- 为什么需要可训练的特征提取器？
 - 为特定的识别任务找到更合适的特征，提高识别效率
 - 应用于一些不知道如何人工设计特征的新领域
 - 用算法批量提取特征，节省时间，可用于大规模数据集
- 一些用于特征检测的机器学习模型
 - 浅层模型：SVM、GMM、Perception、Decision Tree、RBM、Auto-Encoder
 - 深层模型：Deep Auto-Encode、CNN、DBN、DBM

- 神经网络：由多个层级组成，每个层级有一个或多个神经元。神经网络的拟合通过前向传播和反向传播（梯度下降法）实现。
- 反向传播核心是利用链式求导法则把损失函数的梯度从输出层依次向输入层传递，然后根据梯度方向和大小调整每一层的权重：
 - 随机设定权重后把输入的数据输送到网络，得到第一次输出（前向传播）
 - 用损失函数计算输出结果和正确结果的误差
 - 用链式法则对损失函数进行求导得到梯度
 - 根据梯度和设定的学习率从后往前更新每一层的权重参数
- 卷积神经网络是一种用于提取图像中的特征的深度学习模型。它的利用一些小的矩阵（卷积核）在图像上滑动，计算卷积核和图像局部区域的内积，从而得到一个新的矩阵（称为特征图或激活图）。这个过程可以提取出图像中的边缘，纹理，形状等低层次的特征。通过多层的卷积操作，可以逐渐提取出图像中的高层次的特征，如物体的部件，类别等。卷积神经网络的一般结构包括以下几种层：
 - 输入层：对原始图像数据进行预处理，如去均值，归一化，中心化等
 - 卷积层：对输入数据进行卷积操作，提取局部特征
 - 激活层：使用激活函数对卷积层的输出进行非线性变换，如ReLU, tanh等
 - 池化层：对激活层的输出进行下采样，减少数据量和参数量，增强特征的不变性
 - 全连接层：该层每个节点都与下一层的每个节点相连，负责对池化层的输出进行分类或回归，输出最终的结果
 - 输出层：负责输出最终的特征向量，交由损失函数处理