

1 DIP 1

1.1 warping 和 morphing

Warping: 将图像进行空间扭曲, 只改变图像的形状, 不改变图像的内容

Morphing: 改变图像内容, 将一张图渐变到另一张图, 涉及图像的渐变融合。需要对各像素点的颜色、位置作变换。变形的起始图像和结束图像分别为两幅关键帧, 从起始形状变化到结束形状的关键在于自动地生成中间形状, 也即自动生成中间帧。

2 DIP2

超分辨率(Super-Resolution)即通过硬件或软件的方法提高原有图像的分辨率, 通过一系列低分辨率的图像来得到一幅高分辨率的图像过程就是超分辨率重建。

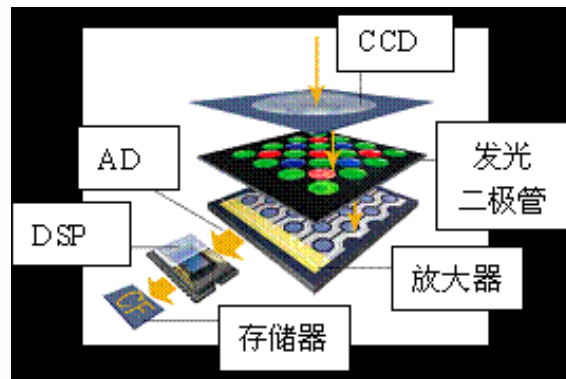
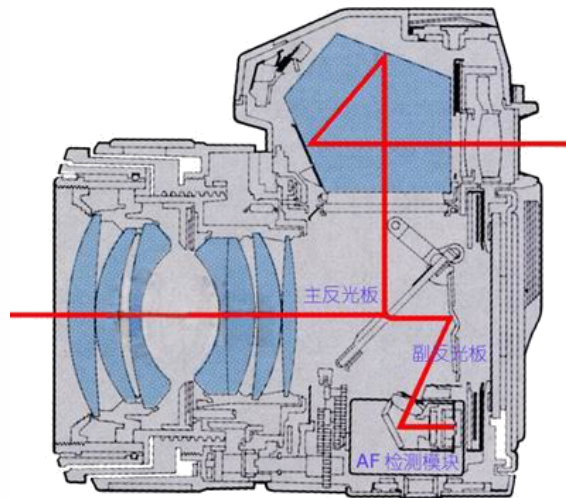
超分辨率重建的核心思想就是用时间带宽(获取同一场景的多帧图像序列)换取空间分辨率, 实现时间分辨率向空间分辨率的转换。

插值是根据一张图像进行插值, 增加分辨率; 超分辨率是对多张照片进行处理。

3 DIP3

3.1 图像的获取

3.1.1 相机结构及原理



3.1.1.1 原理

- (1) 拍照时，来自场景的光线穿过镜头并到达CCD。
- (2) 当CCD曝光时，刺激光电二极管释放电荷，并且产生电信号。
- (3) CCD控制芯片通过控制信号电路控制电流在感光元件中。CCD将收集这些电信号并将其输出到运算放大器。
- (4) 电信号经放大、滤波后到达ADC，进行ADC转换这种电信号（连续）到数字信号（离散）。数字信号的值为与电信号和电压的强度成比例。这些值是与图像的对应。
- (5) 以上数据不能直接作为图像处理，有待进一步研究由DSP（数字信号处理）处理。在DSP中，颜色校正和白平衡将执行，以获得合格的图像，并将图像编码到支持的格式和分辨率，可存储为图像文件。
- (6) 完成上述步骤后，图像文件将显示在存储卡上并可预览。

3.1.1.2 光圈 aperture

光圈越大越糊。

但也不能一直小下去：进光量太少不可；光圈太小造成衍射；小光圈难控制。

3.1.1.3 镜头

镜头的目的在于得到不同焦距镜头的画面效果，不同长短焦距的镜头造型效果不同。

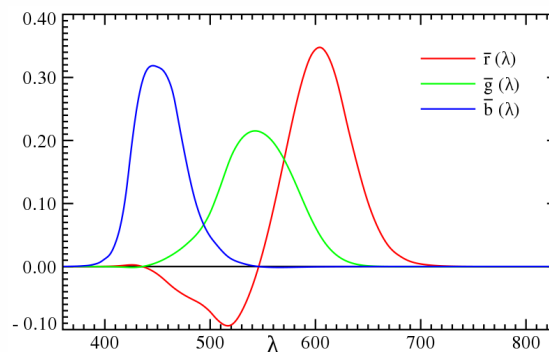
3.1.1.4 景深

光圈、镜头及拍摄物的距离是影响景深的重要因素：

- 1、光圈越大（光圈值f越小）景深越浅，光圈越小（光圈值f越大）景深越深。
- 2、镜头焦距越长景深越浅、反之景深越深。
- 3、主体越近，景深越浅，主体越远，景深越深。

浅：大、长、近

3.2 颜色物理意义



- 越红波长越长，越紫波长越短

3.3 视网膜为什么对光的变换比颜色变换感知弱

视网膜上的细胞：视杆细胞和视锥细胞。负责颜色感知的视锥细胞远远大于视杆细胞。

- 视杆细胞：~100000，对光敏感，但无法识别不同的颜色。

- 视锥细胞：~6000000-7000000，在强光下工作，但可以识别不同的颜色。

3.4 感知的priority和sensitivity

- 优先级：在相同的设置下，人类首先注意色调 (H) 的变化，然后是饱和度 (S)，然后是亮度值 (V)。
- 灵敏度：人眼对亮度变化最敏感，分辨率最好，这是眼睛HDR能力的原因。

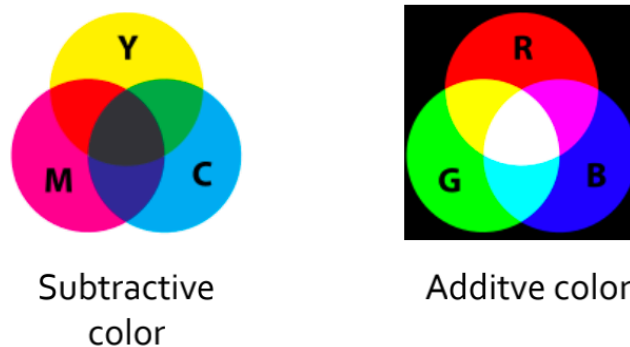
3.5 颜色空间

3.5.1 设备无关

CIE XYZ, CIE L* a* b*, CIE YUV。

3.5.2 设备相关

3.5.2.1 RGB



Red, Green, Blue

加色：(255, 255, 255) 是白色；(0, 0, 0) 是黑色。颜色越加，能量越高，越加越亮。

movies。

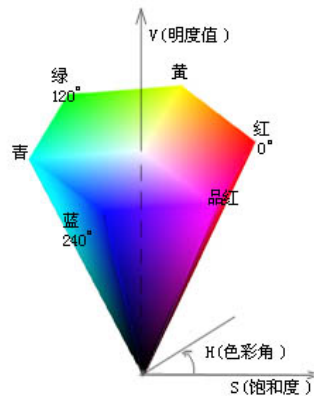
3.5.2.2 CMY

Cyan, Magenta (洋红), Yellow。

减色: (255, 255, 255) 是黑色; (0, 0, 0) 是白色。颜色越加, 能量越低, 越加越黑。

painting、printing。

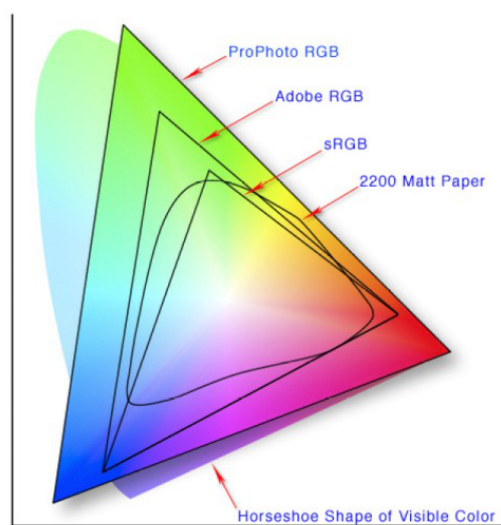
3.5.2.3 HIS / HSV



Hue, Saturation, Intensity / Value。

剥离亮度和颜色

3.5.3 颜色空间覆盖



RGB is a subset of CIE primary color space.

3.6 Jpeg

- 压缩策略：根据压缩比的要求，将信息从高频去除到低频。先从高频开始去除。
- 优点：
 - 高频信息占用更多的内存。因此，高频去除导致高压缩比；
 - 低频信息保留了物体的基本结构和颜色分布，这是图像的关键因素。
 - 适用于基于互联网的视觉媒体。

3.7 RLE

Run length encoding

	0	1	2	3	4	5	6
0							
1		■			■		
2		■	■	■	■		
3							
4							
5			■	■		■	
6							


Run length coding; the code is **{{(11144)}(214)}(52355)}**.

- 在二值图像中，游程编码仅记录属于图像中对象的区域；然后将该区域表示为列表。
 - 图像的每一行都由一个子列表描述，子列表的第一个元素是行号。
 - 后续项为坐标对；一对中的第一个元素是运行的开始，第二个元素是结束。
 - 行中可以有几个这样的序列。

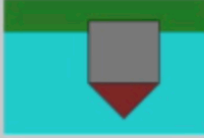
3.8 图像格式

- 无压缩的：BMP
- 有压缩的：
 - 有损压缩：JPEG、JPG、PNG
 - 无损压缩：GIF、TIFF、RAW

3.9 BMP



pixels displayed on the screen



pixels stored in .bmp-file

You do not need to turn around the rows manually. The API functions which also display the bitmap will do that for you automatically.

Another important thing is that the number of bytes in one row must always be adjusted to fit into the border of a multiple of four. You simply append zero bytes until the number of bytes in a row reaches a multiple of four

6 bytes that represent a row in the bitmap: A0 37 F2 8B 31 C4

must be saved as: A0 37 F2 8B 31 C4 00 00

- 存储和实际的显示相反，但是你只要按照正常从上到下编码就可以了，API会自动转换的
- BMP格式，每一行字节必须是4的倍数

4 二值化

4.1 获取二值化图像

1. 阈值：小于阈值取0；大于阈值取255
2. 怎么找阈值：将二值化得到的二值图像视为两部分，一部分对应前景（Foreground），另一部分对应背景（Background）。尝试找到一个合适的threshold使得到的前景和背景的内部方差最小，而它们之间的方差则最大。内部整体是一类，而两者相比有很大的区别。

- 1 Step 1: 确定原始图像中像素的最大值和最小值;
- 2 Step 2: 最小值加1作为threshold对原始图像进行二值化操作;
- 3 Step 3: 根据对应关系确定前景和背景, 分别计算当前threshold下的内部协方差和外部协方差;
- 4 Step 4: 回到Step 2直到达像素最大值;
- 5 Step 5: 找到最大外部和最小内部协方差对应的threshold.
- 6 暴力法, 全部遍历一遍

3. 光照变换处理 -- 局部自适应操作: 设定一个局部窗口, 在整个图像上滑动该窗口; 对于每一窗口位置, 确定针对该窗口的threshold。

4.2 二值图像有什么用

优点: 更小的内存需求; 运行速度更快; 为二值图像开发的算法往往可以用于灰度级图像; 更便宜

缺点: 应用范围毕竟有限; 更无法推广到三维空间中; 表现力欠缺, 不能表现物体内部细节; 无法控制对比度。

4.3 形态学操作

还没看

形态学操作 (腐蚀、膨胀、开操作、闭操作), 及其物理意义, 有实际例子, 让你算一下, 一维 / 二维, 把例子好好看一看, 对偶性, 啥目的 (指纹那页)

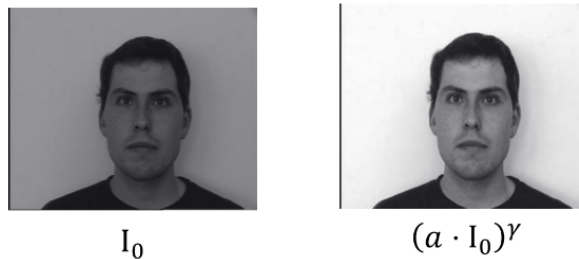
5 可视增强

5.1 Weber's Law

$$\frac{\Delta I}{I} \approx 0.01 - 0.02 \quad (1)$$

最低可被观察到的灰度区分度

5.2 γ 矫正对于图像对比度



$$I = a \cdot I_0^\gamma \quad (2)$$

姑且可以认为这样操作与调整曝光时间的效果是一样的

5.3 对数操作

$$L_d = \frac{\log(L_w + 1)}{\log(L_{max} + 1)} \quad (3)$$

L_d 是实际展示出来的灰度值， L_w 是真实的灰度值， L_{max} 是图像内灰度的最大值。

这个映射能够确保不管场景的动态范围是怎麼样的，其最大值都能映射到1（白），其他的值能够比较平滑地变化。

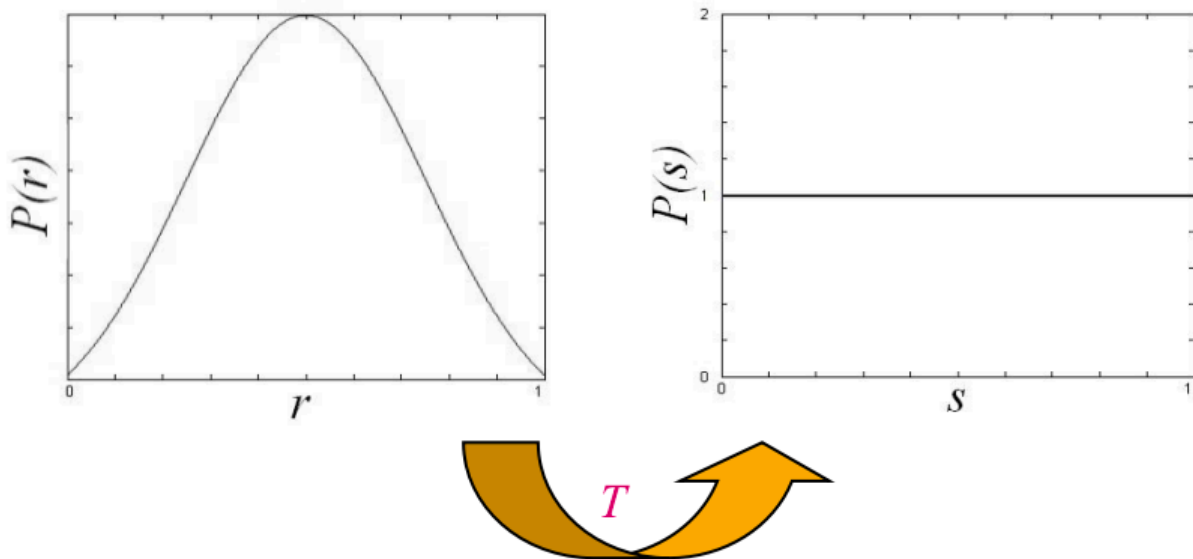
5.4 直方图均衡化

概念；直方图能反应图像颜色分配特点，但无法反应结构信息（直方图匹配的问题）。

直方图均衡化：将灰度直方图变成一个均匀分配。

5.4.1 均衡化公式

5.4.1.1 连续条件



$$s = T(r)$$

$$\int_0^r P(r)dr = \int_0^s P(s)ds = \int_0^s 1 \cdot ds = s = F(r) \quad (4)$$

$$ds = P(r)dr$$

任意一个概率密度函数的变上限积分即概率分布函数都是一个均匀分布。

5.4.1.2 离散情况

$$P(k) = \frac{n_k}{n}$$

$$s = \sum_{i=0}^k P(r_i) = \sum_{i=0}^k \frac{n_i}{n} = \frac{1}{n} \sum_{i=0}^k n_i \quad (5)$$

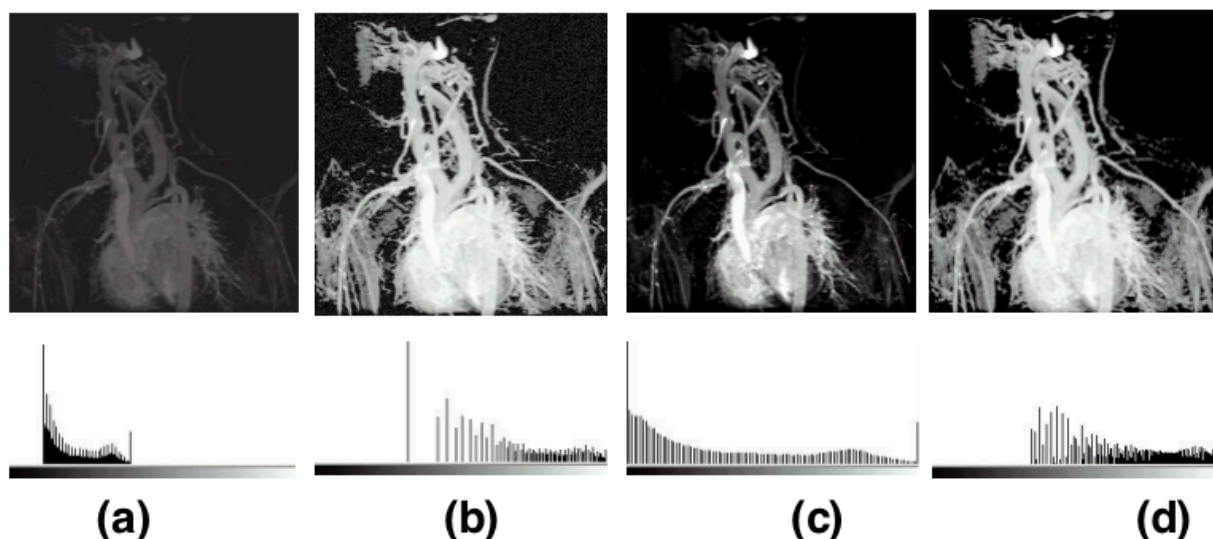
例题掌握DIP7-46，可能考计算题。

5.4.2 均衡化效果

5.4.2.1 为什么不严格按照直方图情况

步骤2中，所得的 s_k 不可能正好等于8级灰度值中的某一级，因此需要就近归入某一个灰度级中。这样，相邻的多个 s_k 就可能落入同一个灰度级，需要在步骤3时将处于同一个灰度级的像素个数累加。因此，离散灰度直方图均衡化操作以后，每个灰度级处的概率密度（或像素个数）并不完全一样。

5.4.2.2 解释



(a) Original image; (b) Histogram equalization on (a);
(c) Grayscale extension on (a); (d) Histogram equalization on (c)

解释为什么每幅图长这样：直方图均衡化会归并很多灰度域，从而达到对比度增强的效果；grayscale extension将灰度值均匀地扩展到整个灰度域中，也可以达到对比度增强的效果，由于不存在归并，所以保留了较多的图像信息。

5.5 直方图匹配的3步骤

方法简述：在步骤1和2中，分别计算获得两张表（参见直方图均衡化中的算例）。 $s = f(r)$ ； $v = g(z)$ 。

步骤3：由 $v = G(z)$ 得到 $z = G^{-1}(v)$ 。由于 s 和 v 有相同的分布，逐一取 $v = s$ ，求出与 r 对应的 $z = G^{-1}(s)$ 。

也就是从中选取一对 v_k 、 s_j ，使 $v_k = s_j$ ，并从两张表中查出对应的 z_k 、 r_j 。 $s(r)$ 图中找到 r_j ， $v(z)$ 图中找到 z_k ，使得 $s(r_j) = v(z_k)$ 。这样，原始图像中灰度级为 r_j 的所有像素都映射成灰度级 z_k ，最终得到所期望的图像。

5.6 直方图变换

比如对数变换：将直方图的灰度值非线性地映射到另一个灰度域中，使灰度值从小到大均有分布，达到了增强对比度的目的；再根据照片本身的情况，选择更多的灰度值是变亮还是变暗。

6 图像几何操作

6.1 插值

6.1.1 最近邻插值

$$(x', y') \xrightarrow[\text{assign value}]{\text{inverse transformation}} (x, y) \xrightarrow{\text{rounding operation}} (x_{int}, y_{int})$$

$$\xrightarrow{\text{assign value}} I_{new}(x', y') = I_{old}(x_{int}, y_{int})$$

为了计算几何变换后新图像中某一点P'处的像素值，可以首先计算该几何变换的逆变换，计算出P'所对应的原图像中的位置P。通常情况下，P的位置不可能正好处在原图像的某一个像素位置上（即P点的坐标通常都不会正好是整数）。寻找与P点最接近的像素Q，把Q点的像素值作为新图像中P'点的像素值。

6.1.2 双线性插值

- 定义双线性方程 $g(x,y)=ax+by+cxy+d$ 。分别将A、B、C、D四点的位置和灰度代入方程，得到方程组。解方程组，解出a、b、c、d四个系数。将P点的位置代入方程，得到P点的灰度。

6.2 RBF

比如拖动图像中的某一个像素，那个控制点的像素变化距离是最明显的，其他像素的移动距离是和该控制点像素的相距距离有关的。

ϕ 函数常用的有**高斯函数**。可能有很多个控制点共同影响。

$$G(x) = \sum_{i=1}^n w_i G(c_i)$$

$$w_i = \frac{\phi(|x - c_i|)}{\sum_{i=1}^n \phi(|x - c_i|)} \quad (6)$$

6.3 Morph

- 大小相同的两幅图的转换作静态变换。从一幅图a逐渐变化成第二幅图b。
- 原理：让图a中每个像素的颜色，逐渐变成图b相同位置像素的颜色。
- 方法：根据变换的快慢，设置相应的步长，将图a每一点的RGB逐渐变成图b相同位置像素的RGB。可以选择等比或等差的方式，或其它方式让：ra-->rb。
- 对于灰度图像，可以直接用等比或等差级数定义步长，使颜色从原图变到目标图。

6.4 表情比例图

Input: Images A A' B

Step1: Mark feature points

Step2: For each feature point v_b in B , warp it:

$$v_{b'} = v_b + v_{a'} - v_a$$

Let B_g be the warped image of B

Step3: Align A , A' with B_g by image warping.

Step4: Compute ratio image: $\mathfrak{R} = \frac{A'}{A}$

Step5: $B' = \mathfrak{R} \cdot B_g$

- Step1：找面部关键点
- Step2：根据A图，对B图进行warp，得到warp后的图像，进行初步的表情更改，但只是限于关键点的移动，并未对细节做深入刻画。
- Step3：将三者对齐
- Step4：计算表情比例
- Step5：将warp后的图像乘以表情比例，得到真正的结果图像

7 卷积

7.1 卷积定义

7.1.1 连续

$$g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x - \tau)d\tau \quad (7)$$

7.1.2 离散

$$g(x) = f(x) * h(x) = \frac{1}{M} \sum_{\tau=0}^{M-1} f(\tau)h(x - \tau) \quad (8)$$

- 第一步先褶，把h(t)翻过去
- 第二步，移动x距离
- 移动后，对每一个样本点t都计算f(t)·h(x-t)，全部累加，再除以M（样本点总数）
- 得到的卷积函数是一个关于滑动距离x的函数

7.2 常用卷积滤波器

7.2.1 均值滤波（各向同性）

Linear smoothing filter— —example

Simple mean, pixels in the mask window contribute equally to the final result.

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

Weighted mean, pixels in the mask window contribute unequally to the final result.

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

Two 3×3 mean filter, each filter's factor equals to the sum of all the coefficients in order to obtain the mean value.

- 有简单平均，也有加权平均
- 各向同性：我们希望滤波器有旋转不敏感性，即对旋转是不敏感的，这样无论图像是朝什么方向的，得到的结果不会改变

7.2.2 中值滤波

- 用领域内的中位数值来代表滤波器中心的数值
- 注意是根据原始图像中的像素点计算中位数，不是根据变换后的
- 对处理脉冲噪声（也称为椒盐噪声）非常有效，因为这种噪声是以黑白点叠加在图像上的。保持锐度，又能去噪。
- 偶数窗口能不能做中值滤波，应该也能。

7.2.3 拉普拉斯做锐化

$$|\nabla f| = [G_x^2 + G_y^2]^{\frac{1}{2}} = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{\frac{1}{2}} \approx |G_x| + |G_y| \quad (9)$$

- 微分算子可以做锐化

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{If the center element of the mask is negative} \\ f(x, y) + \nabla^2 f(x, y) & \text{If the center element of the mask is positive} \end{cases}$$

8 双边滤波

是希望实现图像平滑

8.1 Basic Idea

图像有两个主要特征：

空间域S，是图像中可能的位置集。这与分辨率有关，即图像中的行数和列数。

强度域R，它是一组可能的像素值。用于表示像素值的位数可能会有所不同。常用的像素表示法是无符号字节（0到255）和浮点。

每个样本都被其相邻样本的加权平均值替换，

这些重量反映了两种力：

相邻样本和中心样本的距离有多近，因此权重越大，样本越近。

相邻样本和中心样本的相似程度-相似样本的权重更大。

所有权重均应标准化，以保持局部平均值。

双边滤波不是普遍采用高斯滤波（图像全部都平均），而是增加了一个周围像素的变化强度的影响量。

$$BF[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_t}(|I_p - I_q|) I_q \quad (10)$$

W_p 是一个归一化因子； $G_{\sigma_t}(I_p - I_q)$ 是新增的，关于强度影响项； p 是选择的这个点， q 是周围的点。

如果周围的强度很相近的话，那么得到的值就很近似，达到了该平滑的地方平衡；如果周围有近的像素强度剧烈变化，那么 $G_{\sigma_t}(I_p - I_q)$ 就会比较小，也就是受它影响比较小。关键就是不会把周围的像素一股脑地都接受（只根据位置接收-高斯滤波），而是把距离近、像素近的元素值强烈接收，距离近、像素远的少接受。以此达到了保边的效果。

σ_s 和 σ_t 怎么影响：越大越矮胖，接收范围越广，平滑效果越好

- σ_t 无穷大，就相当于这个因子不起作用，退化成高斯滤波

如果不是灰度图像，那么 $G_{\sigma_t}(|I_p - I_q|)$ 要换成 $G_{\sigma_t}(|C_p - C_q|)$ 计算像素点值之间差的欧式距离。

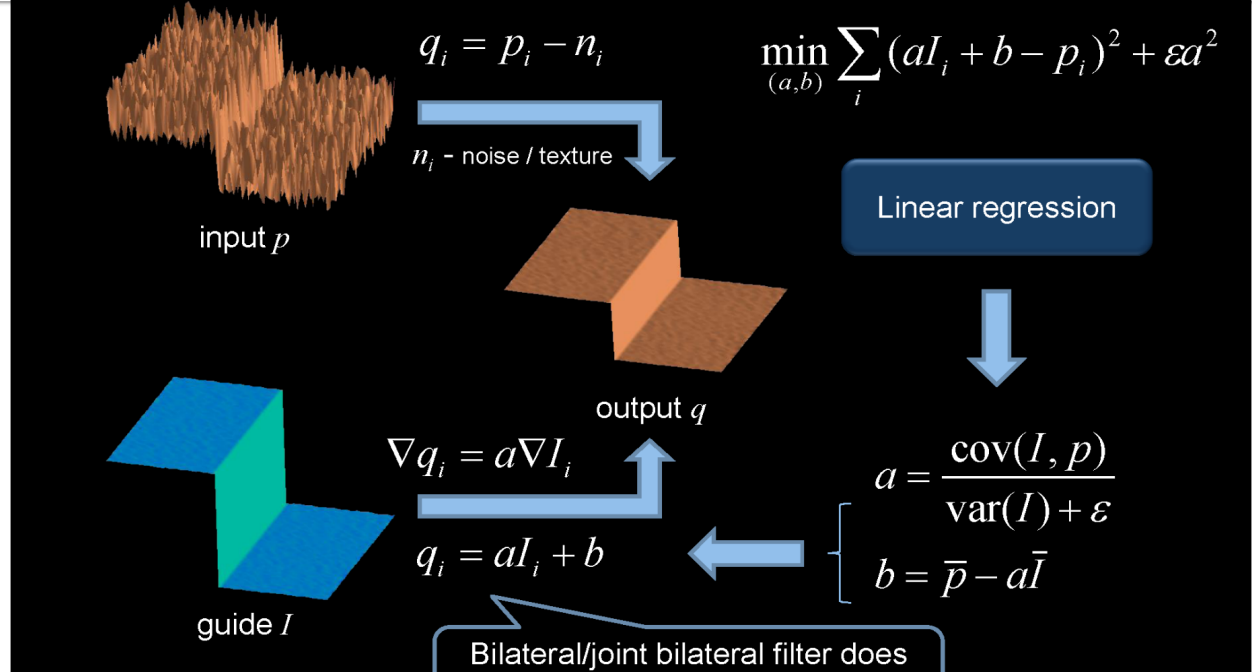
加速版（二维傅立叶，但是感觉不太会考）

9 引导滤波

	共同点	差异
双边滤波	都可以保边	不保梯度
引导滤波		保梯度

- 主要就是输入一张 p ，有一张引导的 I ，输出平滑 p 后的 q 。关键就是求 a 和 b 。
- 这张引导图像也可以是自己，也可以是其他的单独图像。

Guided filter



- 输入是 p ，是带噪声的，希望输出 q 是不带噪声的
- guide image也是不带噪声，两者的梯度有差别，保梯度

$$q_i = aI_i + b \quad (11)$$

求梯度： $\nabla q_i = a \nabla I_i$

- 噪声是求二范数

$$\text{噪声： } \min \sum (aI_i + b - p_i)^2 + \epsilon a^2 \quad (\text{修正项}) \quad (12)$$

- 求解用拉格朗日乘数法：右下角的公式。 a 中的 ϵ 是为了 a 不要太大，是个约束。 a 好像相关系数乘以 p 的方差。

10 稀疏范式滤波

$$\min \sum_{j \in N_i} |I_i^{new} - I_j|^p \quad (13)$$

$p=2$ （均值滤波）、 $p=1$ （中值滤波）、 $p<1$ （稀疏范式，保边能力强）

11 傅立叶变换

11.1 定义

傅立叶变换是复数域的傅立叶级数在极限区间内的推广，可以分成低频和高频成分

正弦波和余弦波具有无限长的特点，这是傅立叶分析的一个缺点，这也解释了小波分析更适合分析某些信号

11.2 去条纹

对一个有噪音的输入函数，我们可以对其采用傅立叶分析，提取出低频和高频成分，再将高频成分去除（将高频成分的傅立叶系数设为0），仅保留低频成分，可以得到去噪后的函数。

图像也是由低频成分和高频成分组成的。低频成分对应了变化较慢的信息，比如连续的面，而高频成分对应了变化较快的信息，比如边等。因此对图像采用类似的傅立叶分析，去除一定的高频成分信息，可以去掉一部分的噪音，达到去条纹的效果。

11.3 FFT

- 将原始的N点序列依次分解为一系列短序列；求出这些短序列的离散傅立叶变换；组合出所需的变换值；计算量（乘除法）： $2N^2 \rightarrow 2N \log_2 N$
- 对一个长度为N的序列进行傅立叶变换可以通过将其分成两半计算，对第一部分的计算需要通过计算两个长度为N/2长度序列的傅立叶变换式进行，然后利用这两个长度为N/2的序列可以得到第二部分的值。
- 可以大概再看一下FFT的过程

傅立叶变换中相位比幅值更重要

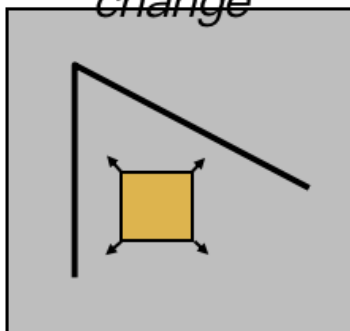
12 图像检测

12.1 Corner

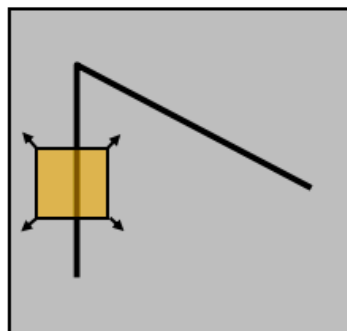
Feature detection

Local measure of feature uniqueness

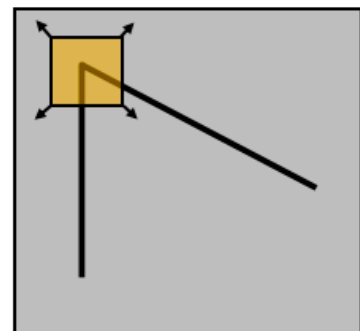
- How does the window change when you shift it?
- Shifting the window in *any direction* causes a *big change*



“flat” region:
no change in all
directions



“edge”:
no change along the
edge direction



“corner”:
significant change in
all directions

Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

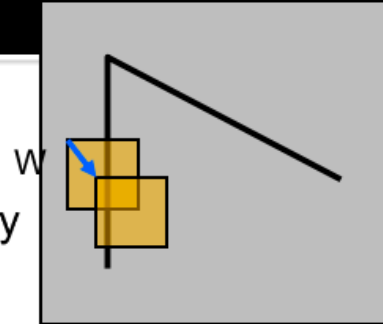
- 选角是很敏感的，任意一点移动都会产生变化

12.2 Harris operator

Feature detection: the math

Consider shifting the window W by (u, v)

- how do the pixels in W change?
- compare each pixel before and after by summing up the squared differences
- this defines an “error” of $E(u, v)$:

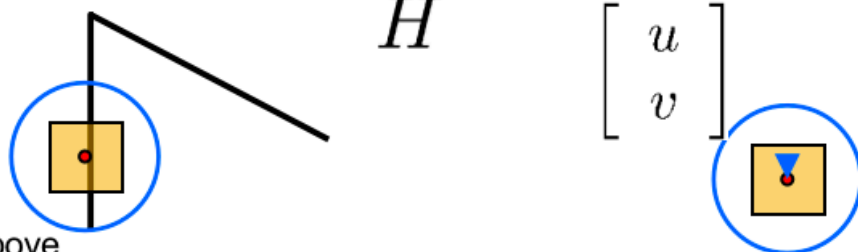


$$\begin{aligned}
 E(u, v) &= \sum_{(x, y) \in W} [I(x + u, y + v) - I(x, y)]^2 \\
 &\approx \sum_{(x, y) \in W} [I(x, y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} - I(x, y)]^2 \\
 &\approx \sum_{(x, y) \in W} \left[[I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} \right]^2
 \end{aligned}$$

Feature detection: the math

This can be rewritten:

$$E(u, v) = \sum_{(x,y) \in W} [u \ v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_H \begin{bmatrix} u \\ v \end{bmatrix}$$



For the example above

- You can move the center of the yellow window to anywhere on the blue unit circle
- Which directions will result in the largest and smallest E values?
- We can find these directions by looking at the eigenvectors of H

- H可以写成特征向量和特征值的矩阵相乘形式，对于一个2*2的矩阵，有两个特征值和特征向量
- 算出来有两个特征值，大的叫 λ_+ ，小的叫 λ_- ，常比较小那个特征值的情况（与阈值比较）
- 这两个特征值能给出一个特征椭圆。
- Harris提出了一个算子，比计算特征向量的时间少很多

$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} = \frac{\det(H)}{\text{tr}(H)} \quad (14)$$

12.3 性质

旋转不变性：旋转后，计算得到的特征值还是一样的，相当于只对椭圆进行旋转，形状一模一样。

intensity change：相对大小不会改变，仍能确定极值点

图像大小：会影响

12.4 尺度不变检测器

- 首先先找多尺度的Harris corner。做好空间上的分辨。
 - 会出现某个尺度上找不到了Harris corner，这就说明这个点是不稳定的。
- 寻找好Harris corner之后，用lap来筛选不同的几个尺度上的lap响应值。取lap响应值最大的那个作为合适的尺度。

这个lap是怎么筛选的，是怎么响应的？

13 SIFT

13.1 原理

1. 生成高斯差分金字塔（DOG金字塔），尺度空间构建
2. 空间极值点检测（关键点的初步查探）

在二维图像空间，中心点与它 3×3 邻域内的8个点做比较，在同一组内的尺度空间上，中心点和上下相邻的两层图像的 2×9 个点作比较，如此可以保证检测到的关键点在尺度空间和二维图像空间上都是局部极值点。

3. 稳定关键点的精确定位

去除不稳定和错误检测出的极值点

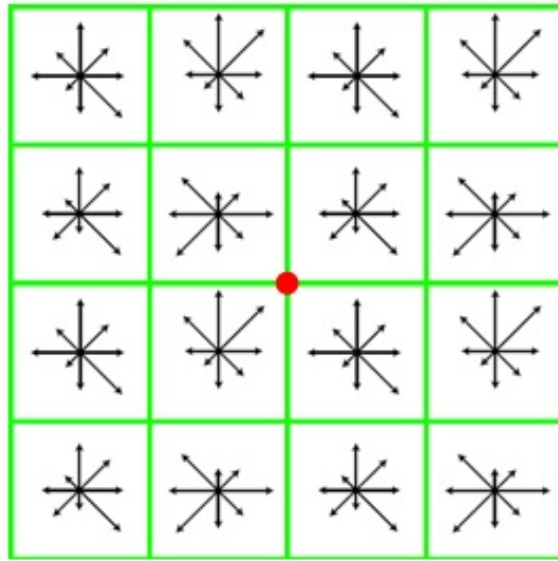
4. 稳定关键点方向信息分配

对关键点求梯度：有数值和方向

5. 关键点描述

对关键点周围像素区域分块，计算块内梯度直方图，生成具有独特性的向量，这个向量是该区域图像信息的一种抽象表述。一块里面16个像素，有16个梯度、16个幅值，每个块都计算八维的向量

如下图，对于 4×4 块，每块的所有像素点的做高斯加权，每块最终取8个方向。（问：这八个方向是固定了的吗？）即可以生成 $4 \times 4 \times 8$ 维度的向量，以这 $4 \times 4 \times 8$ 维向量作为中心关键点的数学描述。



6. 特征点匹配

13.2 优点

在比例、旋转、照明等变化中保持不变性的期望特性。

在局部斑块中具有高度的独特性和描述性。

在刚性对象表示中特别有效。

13.3 缺点

提取耗时：对于大小为400 x 400的图像，平均约为1秒。

非刚性对象的性能较差，如人脸、动物等。

可能无法在严重的仿射失真中工作。

局部面片是一个圆，而不是根据仿射变形调整的椭圆。

14 SURF

14.1 算法描述

- 兴趣点检测器：
 - 计算积分图像：从左上角积出来，积分图像允许我们对过滤器进行上采样，而不是对图像进行下采样。
 - 对图像应用二阶导数（近似）过滤器：二阶导真正的滤波器是13-14左半边的图，为了简便运算，取成右半边的图。13-16，过滤器是以6为幅度递增的，原因是保留二阶导滤波器的完整性。
 - 非最大抑制（在 (x, y, σ) 空间中查找局部极大值）：sigma可以理解成图像的大小。一个像素和上下空间以及自己周围的26个点进行比较。
 - 二次插值：对于局部极大值，要进行二次插值以为了之后的子窗口运算。（问：插哪里？子窗口里面插嘛？）
 - 在兴趣点周围建立4*4的子窗口，每个子窗口内利用哈尔小波计算5*5的dx和dy，每个子窗口得到4个向量。

Interest point *descriptor*:

- Divide window into 4x4 (16 subwindows)
- Compute Haar wavelet outputs
- Within each subwindow, compute

$$v_{subregion} = \left[\sum dx, \sum dy, \sum |dx|, \sum |dy| \right]$$

- This yields a 64–element descriptor

(Only implement USURF – no rotation)

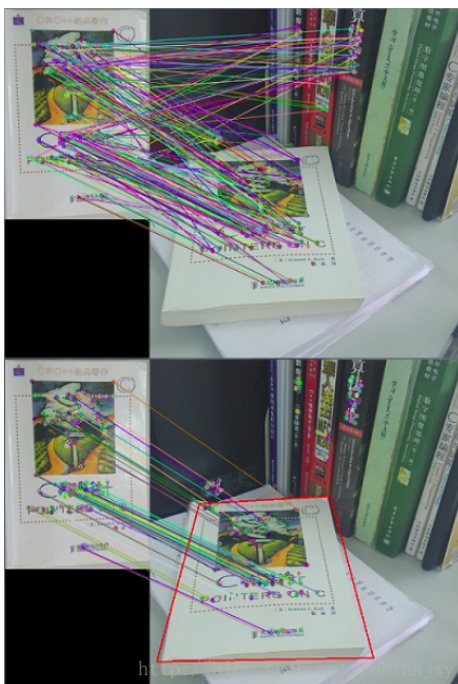
14.2 RANSEC

RANSEC实际上就是找到一批点都尽量满足在一条线上，这样就可以拼接图像

14.2.1 RANSAC步骤

- 随机选择一组点作为变换估计的基础（例如，一组匹配）
- 从种子组计算变换
- 查找此转换的内联点
- 如果内联点的数量足够大，则重新计算所有内联点上的最小二乘估计

保持具有最多内点。可以理解为，匹配的时候可能会出现误匹配，我们可以把每次匹配的结果都转化成一个参数点，如果有一批点他们都满足一个模型内（比如这些点都在一条直线上），那么我就认为这些参数点都是我预估这个模型的正确值，其他的都是噪音点无用的。把这些正确点拿出来就好。



- 图像拼接最少要几个点：4个样本

14.3 Blending 融合

图像融合步骤：检测关键点；构建SIFT描述符；匹配筛选描述符；拟合变换；RANSEC；image blending。

通过高斯金字塔、拉普拉斯金字塔分别提取出颜色和条纹细节特征。lap的level越小，越能提取出细节。再分别取两张图像的融合边的两边的不同level的金字塔值，进行叠加，比如在比较低的范围内，主色调互相融合羽化，在比较高的level内，细节值取得比较精细，这样大致一看没有什么大的裂缝，仔细一看细节也能很好保留。他不是完全的中间一刀切，实际上一一种羽化的操作，左边会渗一点右边的基础色调，右边也会渗一点左边的基础色调。

15 神经网络

15.1 BP神经网络

BP神经网络是一种按误差反向传播(简称误差反传)训练的多层前馈网络，其算法称为BP算法，它的基本思想是梯度下降法，利用梯度搜索技术，以期使网络的实际输出值和期望输出值的误差均方差为最小。

基本BP算法包括信号的前向传播和误差的反向传播两个过程。即计算误差输出时按从输入到输出的方向进行，而调整权值和阈值则从输出到输入的方向进行。

正向传播时，输入信号通过隐含层作用于输出节点，经过非线性变换，产生输出信号，若实际输出与期望输出不相符，则转入误差的反向传播过程。

误差反传是将输出误差通过隐含层向输入层逐层反传，并将误差分摊给各层所有单元，以从各层获得的误差信号作为调整各单元权值的依据。通过调整输入节点与隐层节点的联接强度（权值）和隐层节点与输出节点的联接强度（权值）以及阈值，使误差沿梯度方向下降。

权重大的边，分担大的误差，权重小的边，分担小误差；在下一轮会调整边的权重。不停的迭代，误差就会越来越小，直到设定的一个阈值。最开始的权重是随机分配的，有加一个eps（称学习率），作为调整的值，一般较小。

经过反复学习训练，确定与最小误差相对应的网络参数(权值和阈值)，训练即告停止。此时经过训练的神经网络即能对类似样本的输入信息，自行处理输出误差最小的经过非线性转换的信息。

15.2 为什么要卷积

原始神经网络中边很多，每个边都有一个权重，那么权值的数量级就很大，反向传播十分耗时。但图像中不同part的参数可能是相同的，因此采用多个不同大小的卷积核，对图像进行卷积操作。对每个卷积核，正反向传播时也只需要修改卷积核内的参数来改变权值，这样就大大减少了计算量。

15.3 池化层

降维操作