第三讲 图像预处理

周文晖

计算机学院

3.1 图像几何变换

基本坐标变换

坐标变换的矩阵表示

- 针对二维图像的坐标变换
  - 齐次坐标表示
  - 矩阵形式表示

$$\underbrace{\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}}_{\text{变换后坐标}} = \mathbf{A} \underbrace{\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{\text{原始坐标}}$$

$$\mathbf{A} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}}_{\text{变换矩阵}}$$

平移变换

$$\begin{cases} u' = u + u_0 \\ v' = v + v_0 \end{cases} \Rightarrow \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & u_0 \\ 0 & 1 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & -u_0 \\ 0 & 1 & -v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

尺度变换

$$\begin{cases} u' = s_x u \\ v' = s_y v \end{cases} \Rightarrow \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \mathbf{S} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{S}^{-1} = \begin{bmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

旋转变换

- 定义顺时针旋转为正

$$\begin{cases} u' = \quad u\cos\theta + v\sin\theta \\ v' = -u\sin\theta + v\cos\theta \end{cases} \Rightarrow \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \mathbf{R} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{R}^{-1} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

级联变换

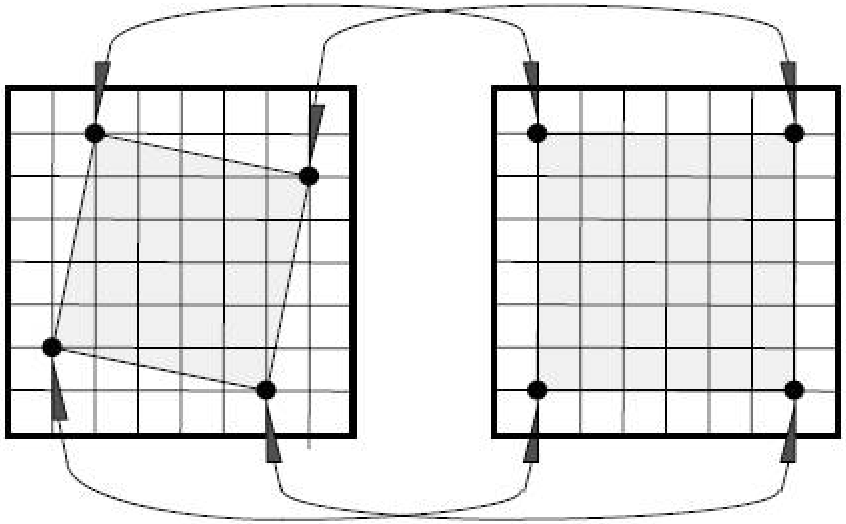
- 用矩阵乘实现级联变换
- 如图像依次进行平移、尺度和旋转变换，有

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \mathbf{R} \cdot \mathbf{S} \cdot \mathbf{T} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

3.1 图像几何变换

几何失真变换

几何失真变换



$$\begin{cases} u' = f_u(u,v) \\ v' = f_v(u,v) \end{cases}$$

- 线性几何失真变换
- 非线性几何失真变换

几何失真变换

- 线性几何失真变换

$$\begin{cases} f_u(u,v) = k_1 u + k_2 v + k_3 \\ f_v(u,v) = k_4 u + k_5 v + k_6 \end{cases} \Rightarrow \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

- 非线性几何失真变换

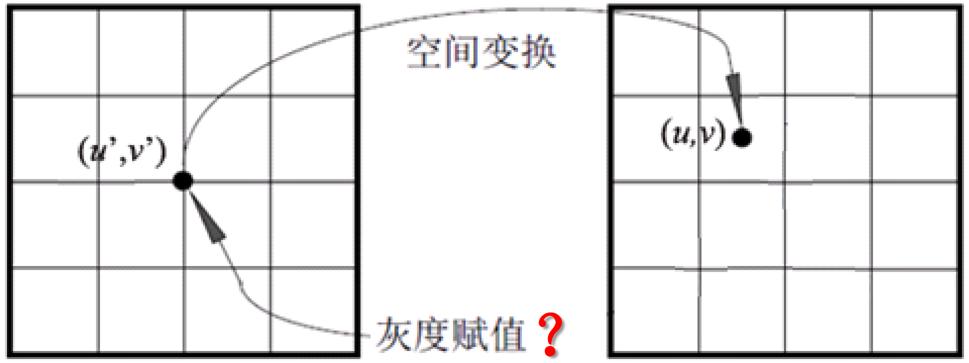
$$\begin{cases} u' = k_1 + k_2 u + k_3 v + k_4 u^2 + k_5 uv + k_6 v^2 \\ v' = k_7 + k_8 u + k_9 v + k_{10} u^2 + k_{11} uv + k_{12} v^2 \end{cases}$$

3.1 图像几何变换

灰度插值

灰度插值

- 坐标变换是像素空间域上的转换
- 灰度插值是像素值域上的转换

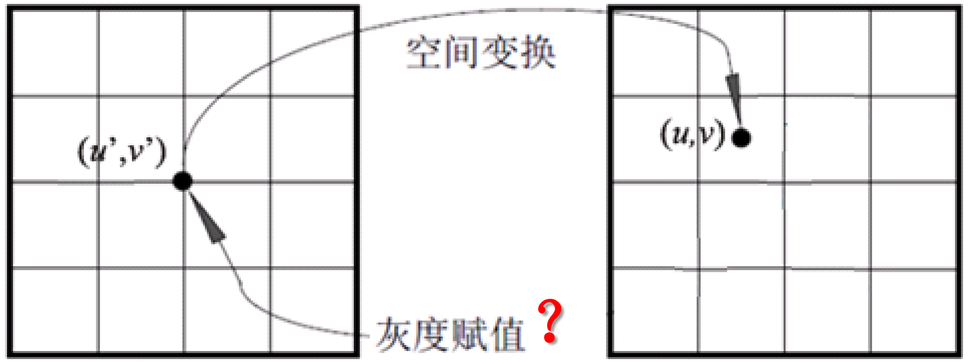


变换后图像　　　　原始图像

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

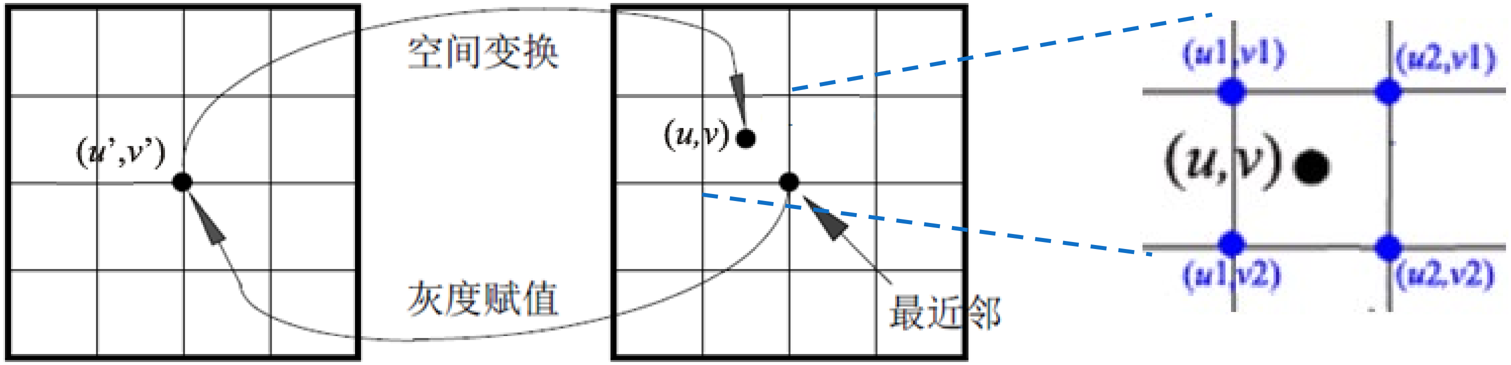
- 为保证变换后的图像像素坐标(u',v')为整数，因此通过A^{-1}反算其在原始图像中的坐标(u,v)；
- 变换后图像 (u',v')像素灰度值=原始图像(u,v)像素灰度值?

灰度插值



- (u,v)通常都是非整数值；
- 变换后图像 (u',v')像素取值由原始图像(u,v)四个邻近整数坐标的像素决定。
- 两种常用的插值算法
  - 最近邻插值
  - 双线性插值

最近邻插值

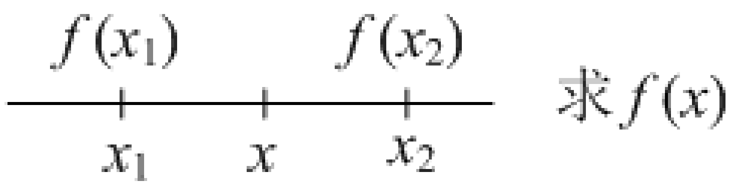


- (u,v)四个邻近整数坐标像素$(u1,v1)$, $(u2,v1)$, $(u1,v2)$, $(u2,v2)$
- 取距离(u,v)**最近**的整数坐标像素灰度值

双线性插值

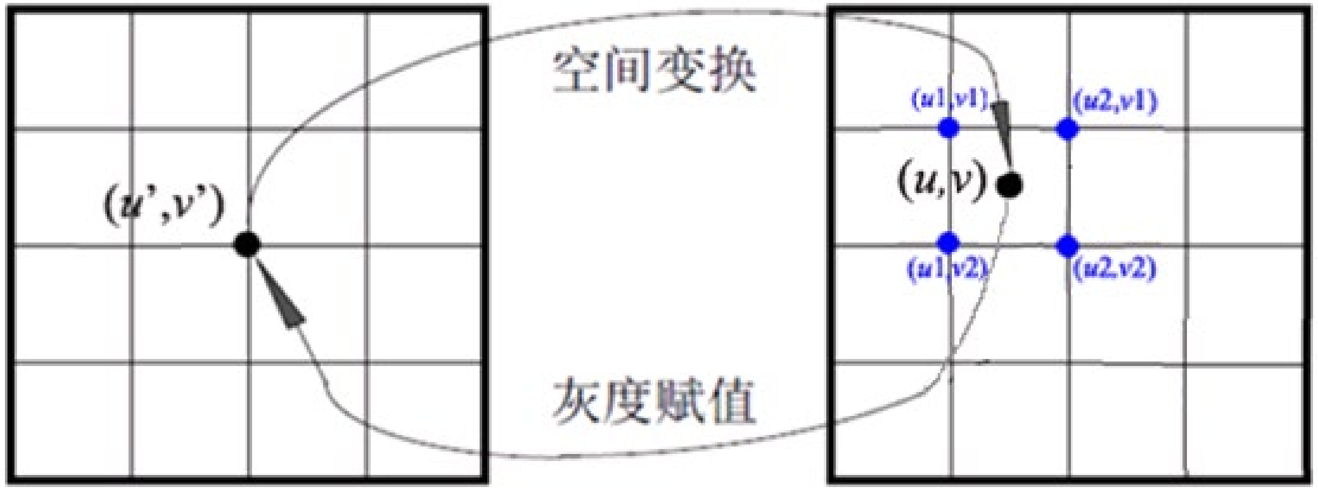
- 将一维线性插值推广到二维

- 什么是一维线性插值？



$$f(x) = \frac{x - x_1}{x_2 - x_1}\left(f(x_2) - f(x_1)\right) + f(x_1)$$

双线性插值



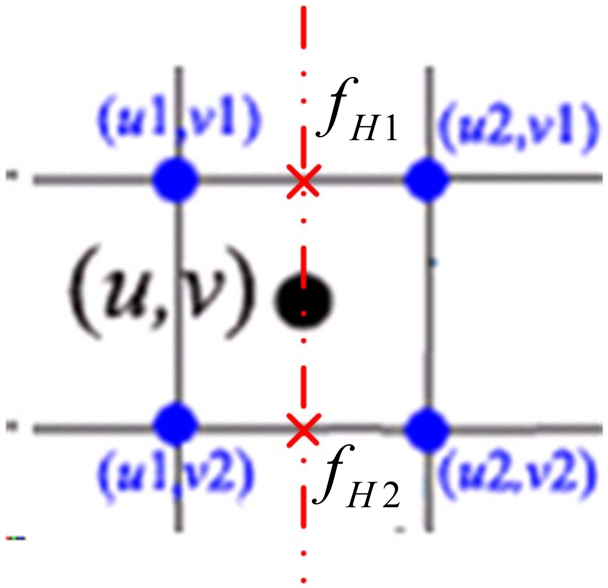
- 双线性插值：水平方向线性插值+垂直方向线性插值
- 水平方向（$v1$行和$v2$行）：$u1$列和$u2$列间插值
- 垂直方向： $v1$行和$v2$行间插值

双线性插值

- 第一步：水平方向线性插值

$$f_{H1} = \frac{u-u_1}{u_2-u_1}\left(f\left(u_2,v_1\right)-f\left(u_1,v_1\right)\right)+f\left(u_1,v_1\right)$$

$$f_{H2} = \frac{u-u_1}{u_2-u_1}\left(f\left(u_2,v_2\right)-f\left(u_1,v_2\right)\right)+f\left(u_1,v_2\right)$$



$\Downarrow$ $u2\text{-}u1 = 1$

$$f_{H1} = \left(u-u_1\right)\left(f\left(u_2,v_1\right)-f\left(u_1,v_1\right)\right)+f\left(u_1,v_1\right)$$

$$f_{H2} = \left(u-u_1\right)\left(f\left(u_2,v_2\right)-f\left(u_1,v_2\right)\right)+f\left(u_1,v_2\right)$$

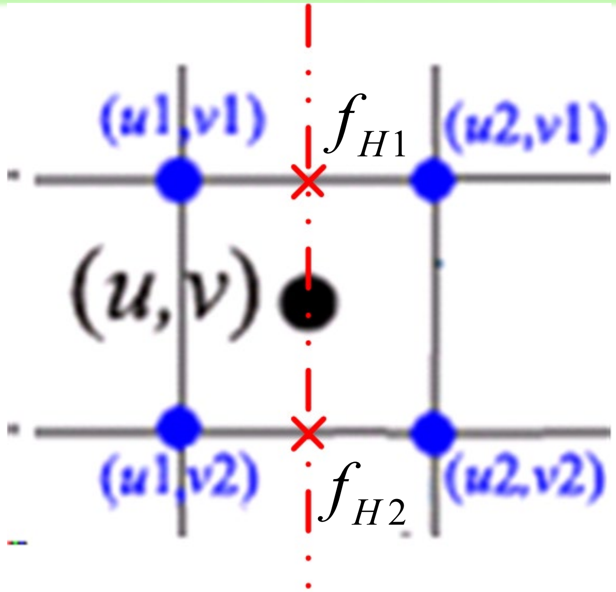
双线性插值

- 第二步：垂直方向线性插值

$$f_{H1} = \frac{v - v_1}{v_2 - v_1}\left(f_{H2} - f_{H1}\right) + f_{H1}$$

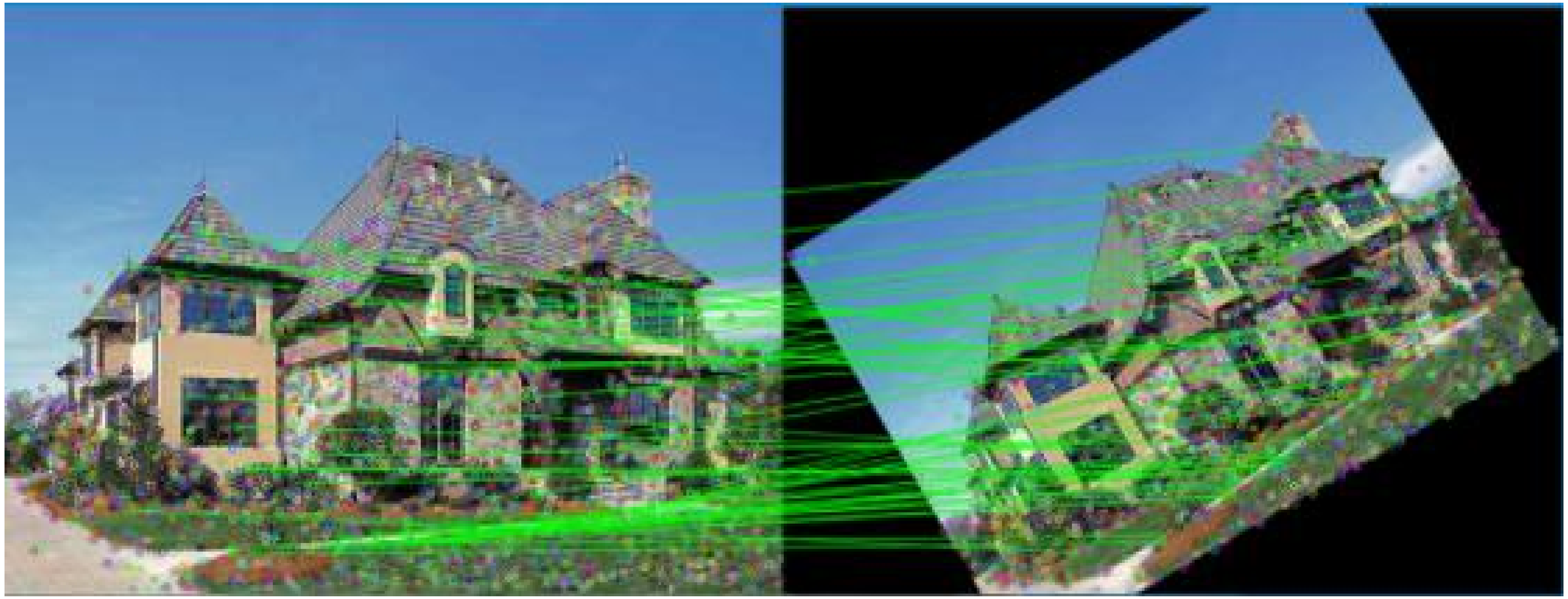
$v2\text{-}v1 = 1$

$$f_{H1} = \left(v - v_1\right)\left(f_{H2} - f_{H1}\right) + f_{H1}$$



计算空间变换

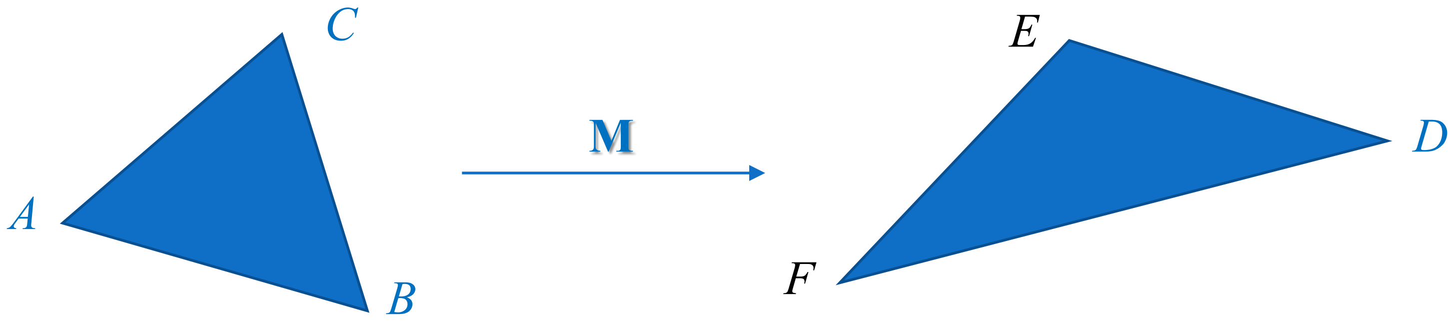
- **两个步骤：**
  - 计算空间变换函数 **M**
  - 插值填充



问题：变换矩阵未知，其参数需通过若干已知的匹配对应点后求解得到

计算空间变换

- 若已知两个三角形 ABC和DEF的映射关系：A变换为D，B变换为E，C变换为F，应如何确定两者间的变换矩阵？

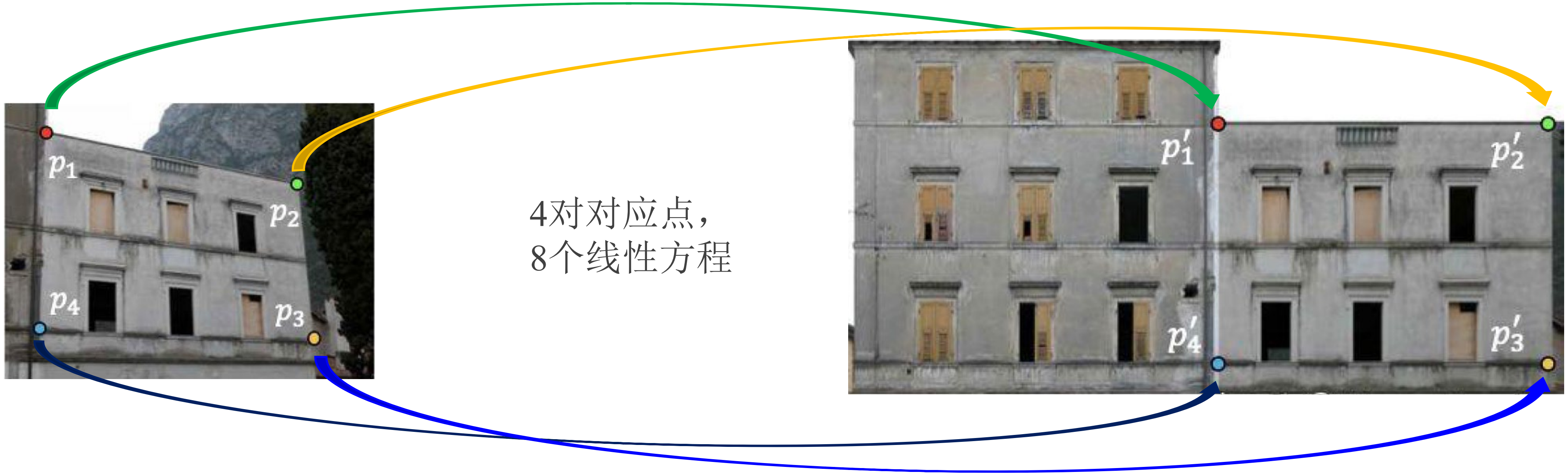


$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \mathbf{M} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} & \vdots & & & \vdots & \\ u & v & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & u & v & 1 \\ & \vdots & & & \vdots & \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix} = \begin{bmatrix} \vdots \\ u' \\ v' \\ \vdots \end{bmatrix}$$

计算空间变换例

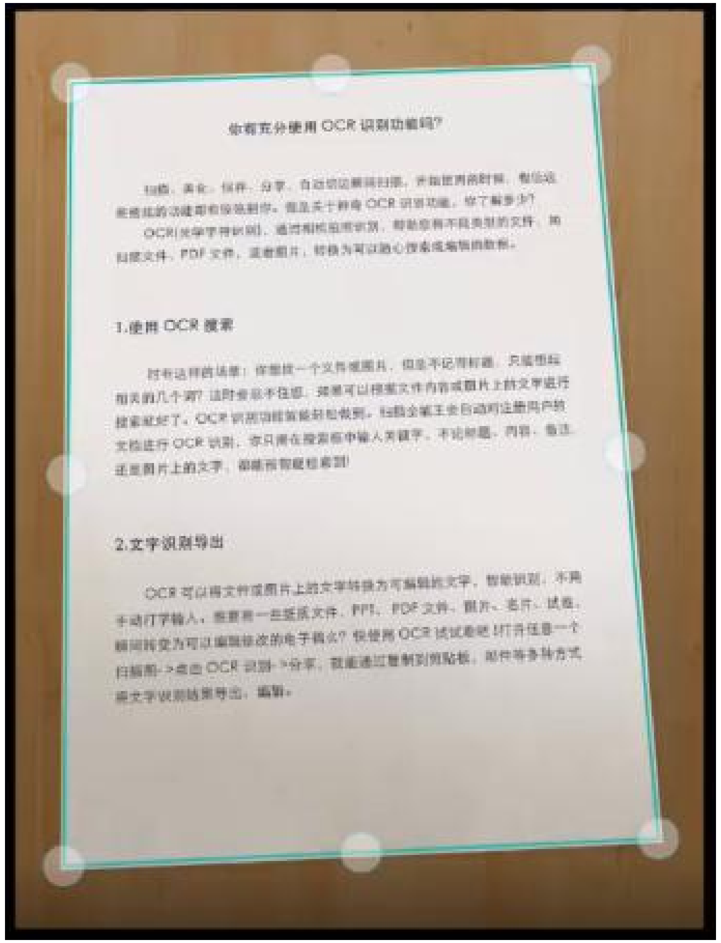
图像校正

和拼接

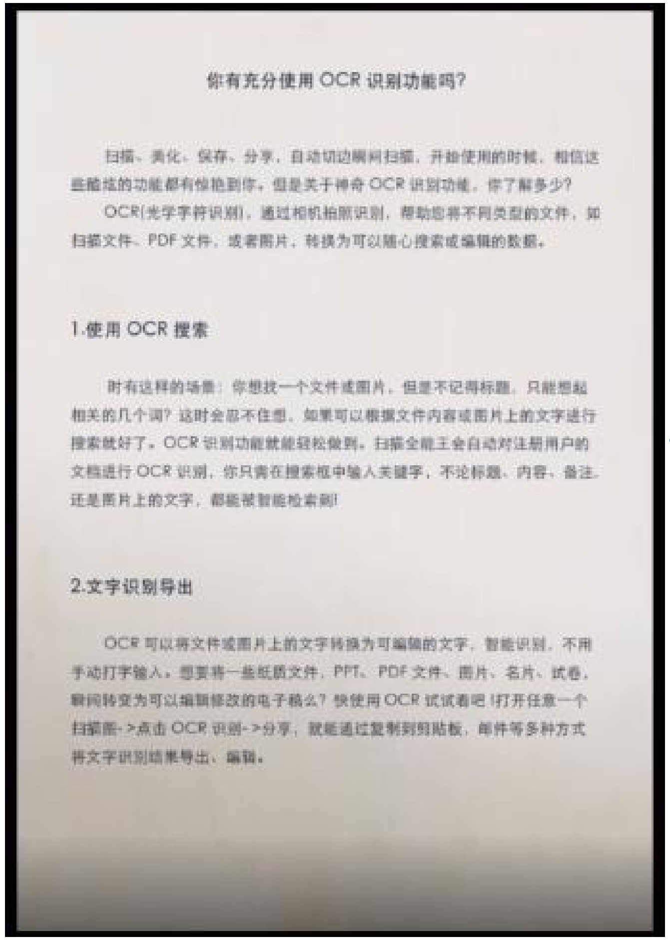


4对对应点，
8个线性方程

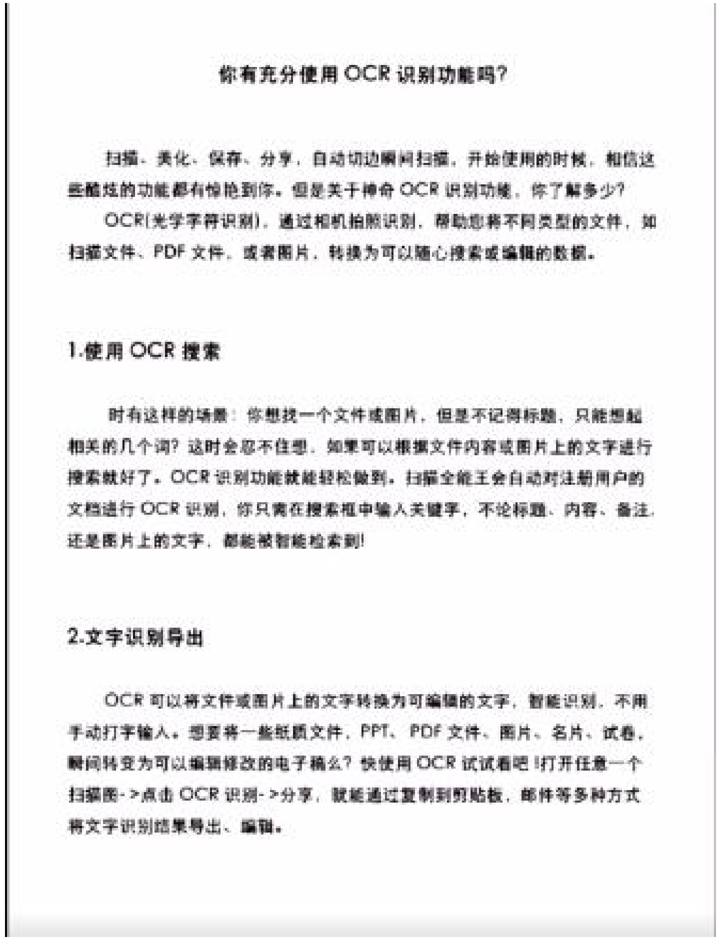
例：OCR倾斜校正



倾斜校正



二值化



你有充分使用 OCR 识别功能吗?

扫描、美化、保存、分享，自动切边瞬间扫描，开始使用的时候，相信这些酷炫的功能都有惊艳到你。但是关于神奇 OCR 识别功能，你了解多少?

OCR(光学字符识别)，通过相机拍照识别，帮助您将不同类型的文件，如扫描文件、PDF 文件、或者图片，转换为可以随心搜索或编辑的数据。

1.使用 OCR 搜索

时有这样的场景：你想找一个文件或图片，但是不记得标题，只能想起相关的几个词? 这时会忍不住想，如果可以根据文件内容或图片上的文字进行搜索就好了。OCR 识别功能就能轻松做到。扫描全能王会自动对注册用户的文档进行 OCR 识别，你只需在搜索框中输入关键字，不论标题、内容、备注、还是图片上的文字，都能被智能检索到!

2.文字识别导出

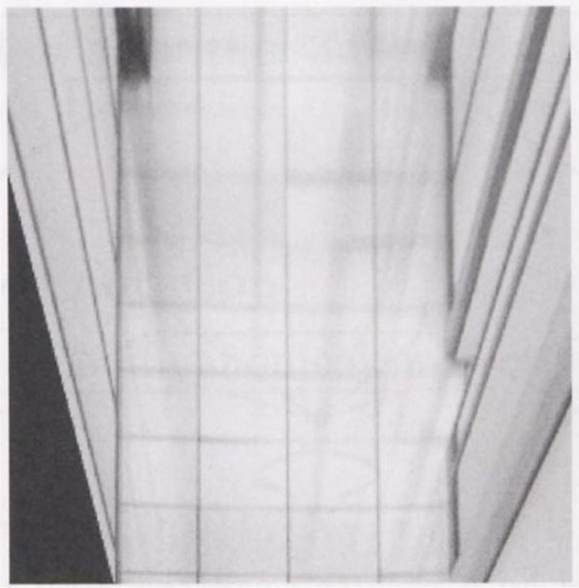
OCR 可以将文件或图片上的文字转换为可编辑的文字，智能识别，不用手动打字输入。想要将一些纸质文件、PPT、PDF 文件、图片、名片、试卷，瞬间转变为可以编辑修改的电子稿么? 快使用 OCR 试试看吧 !打开任意一个扫描图->点击 OCR 识别->分享，就能通过复制到剪贴板，邮件等多种方式将文字识别结果导出、编辑。

例：视角变换

- 正视图变换为俯视图或侧视图

正视图

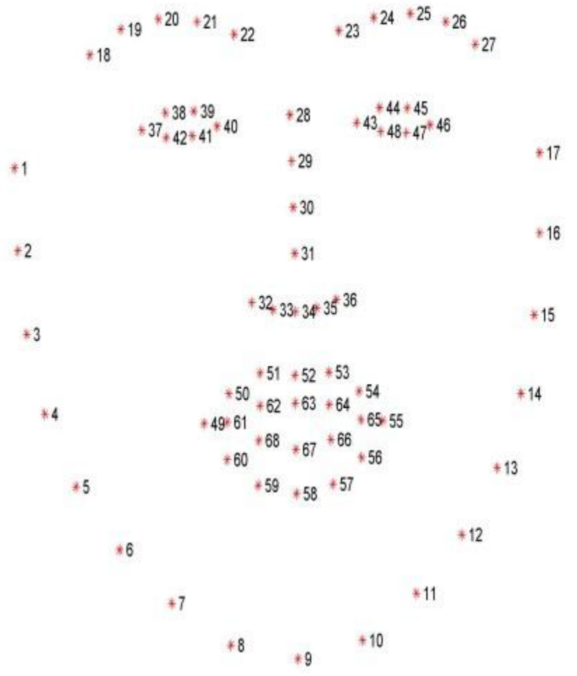
合成的俯视图

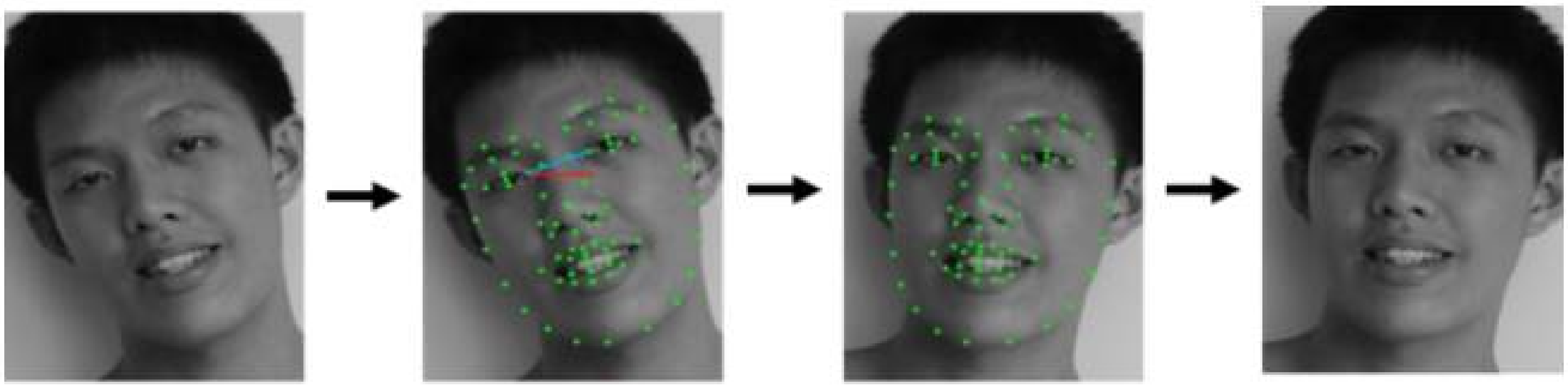


合成的侧视图

例：人脸对齐

- 检测人脸关键点，然后根据这些关键点对人脸做对齐校准（ Face Alignment ）。使用仿射变换将人脸统一“摆正”，尽量去消除姿势不同带来的误差。





检测人脸关键点： dlib，openpose，alphapose，face++ api

例：自动驾驶BEV特征融合

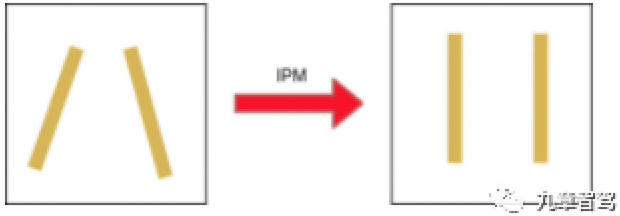
在BEV空间中的融合优势

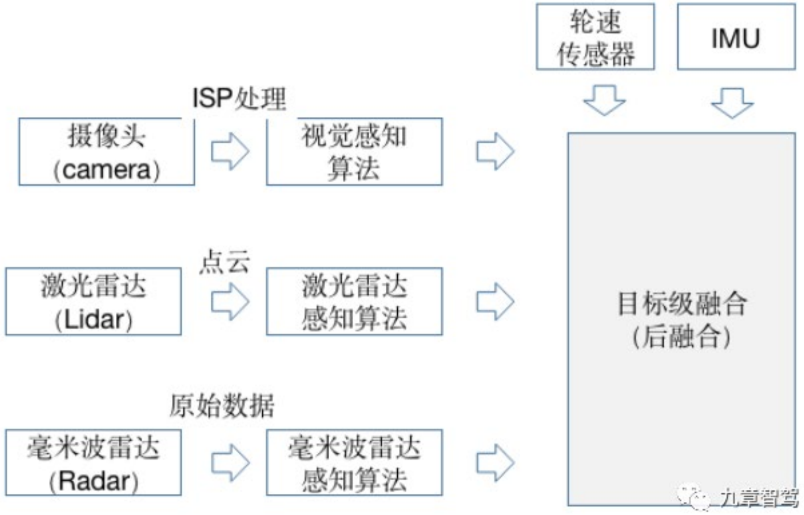
跨摄像头融合和多模融合更易实现

时序融合更易实现

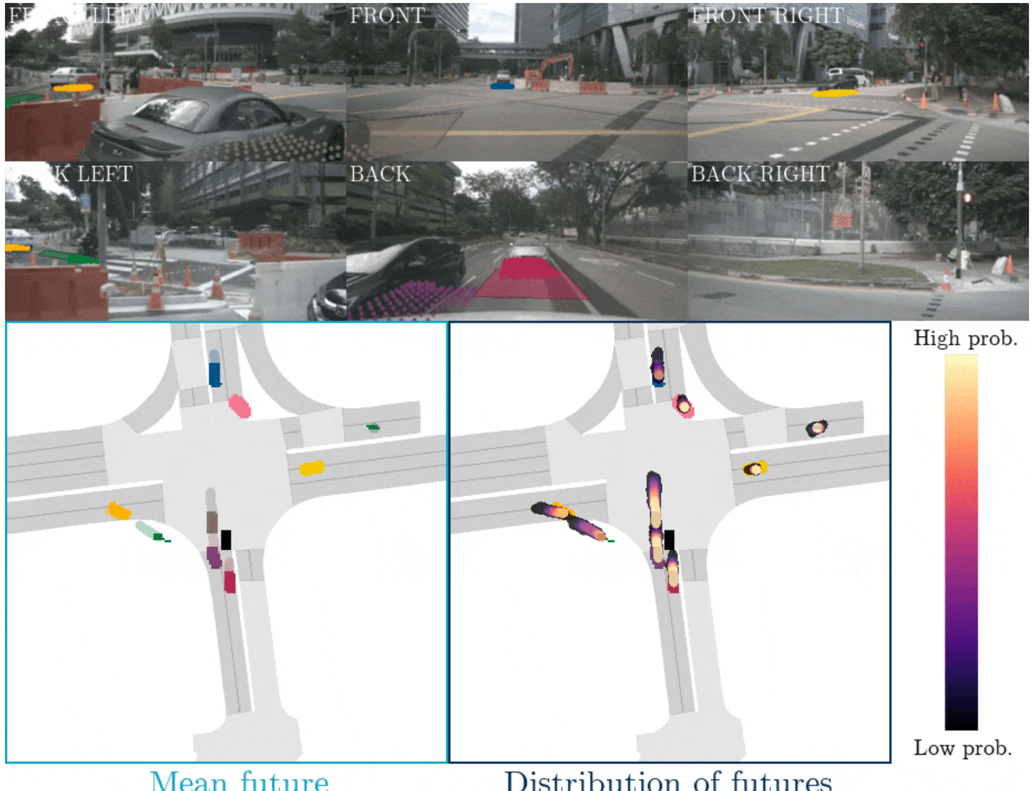
可预测出被遮挡区域的目标

端到端优化更方便



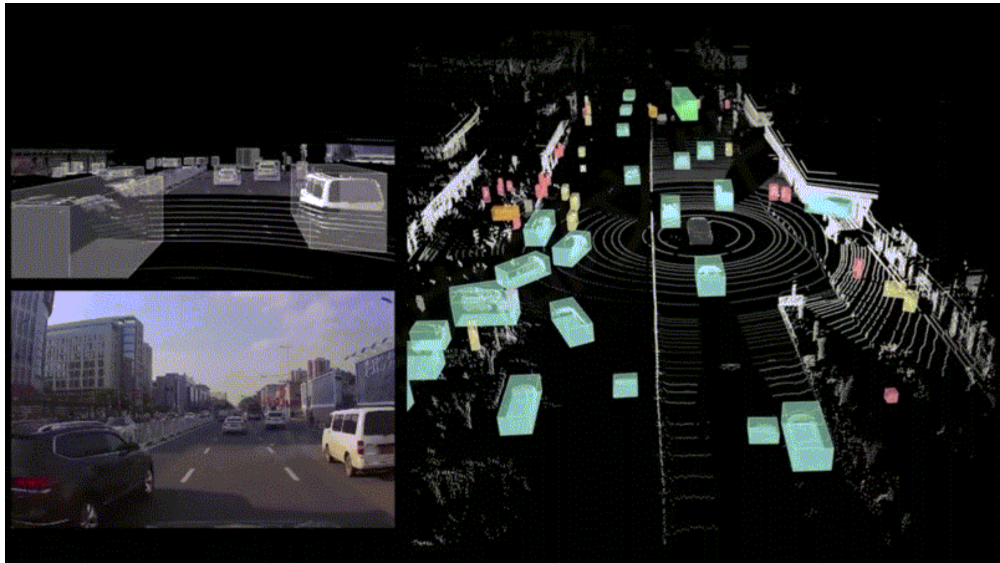


- 英国的自动驾驶创业公司Wayve和剑桥大学合作提出的FIERY网络



- 相机和激光雷达融合

视觉数据是2D图像，激光雷达点云是3D空间。融合时，将点云投影到图像空间，为像素提供深度信息；或在点云坐标系里，为点云赋值颜色信息。

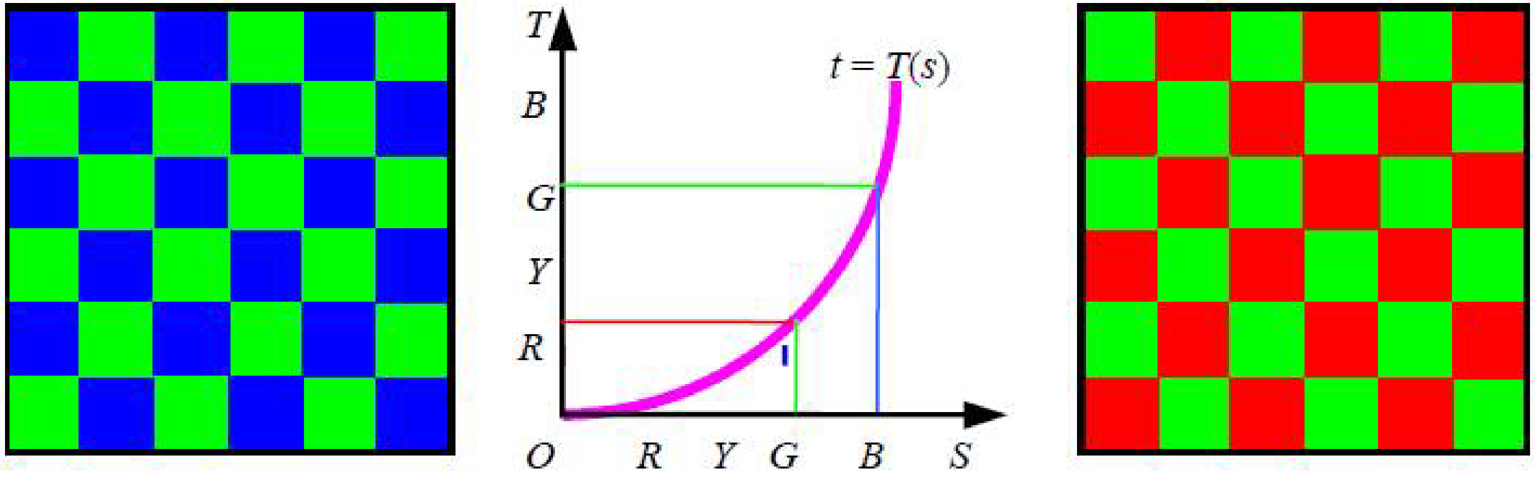


3.2 灰度映射

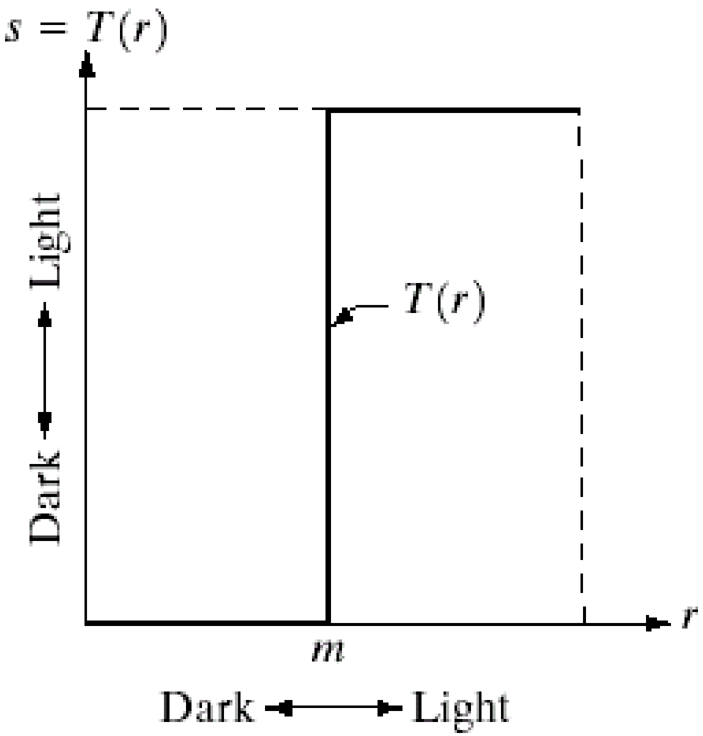
灰度映射原理和应用

灰度映射原理

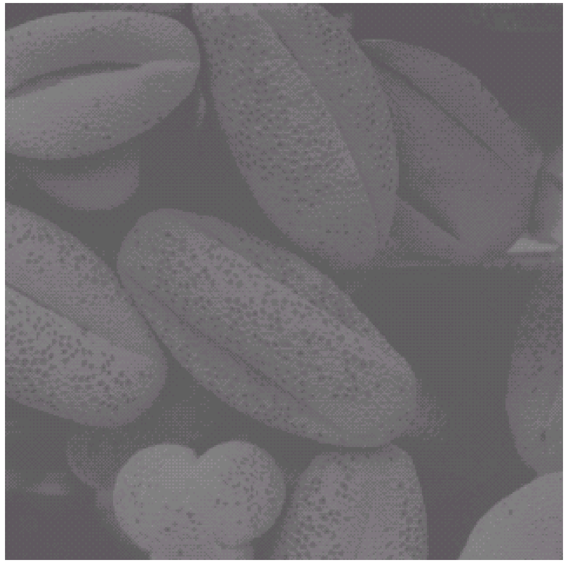
- 基于图像像素的点操作
- 映射函数 $t = T(s)$
- 灰度映射的关键是根据增强要求设计映射函数



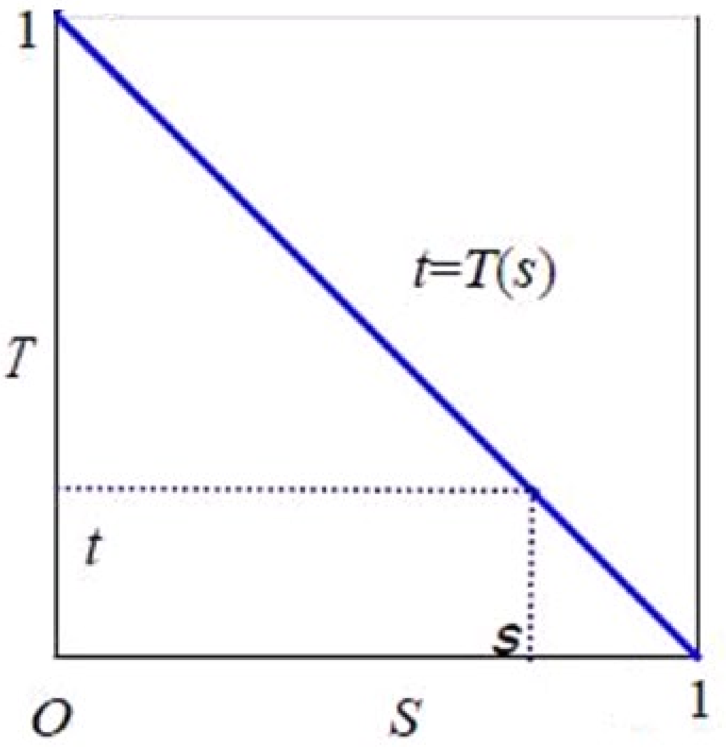
图像二值化



$$T_r = \begin{cases} 1 & r \geq m \\ 0 & r < m \end{cases}$$

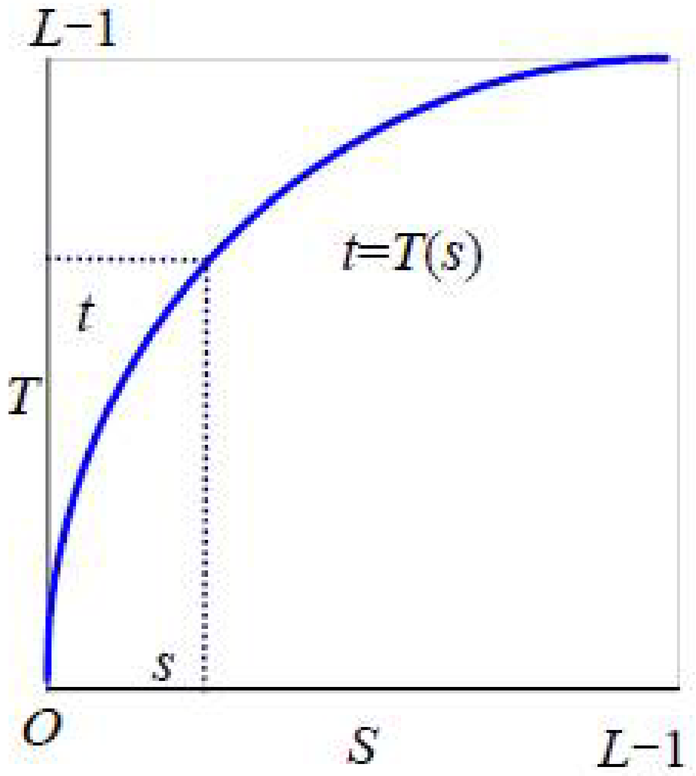


灰度映射：图像反色



$$T_r = 1 - s$$

灰度映射：动态范围压缩



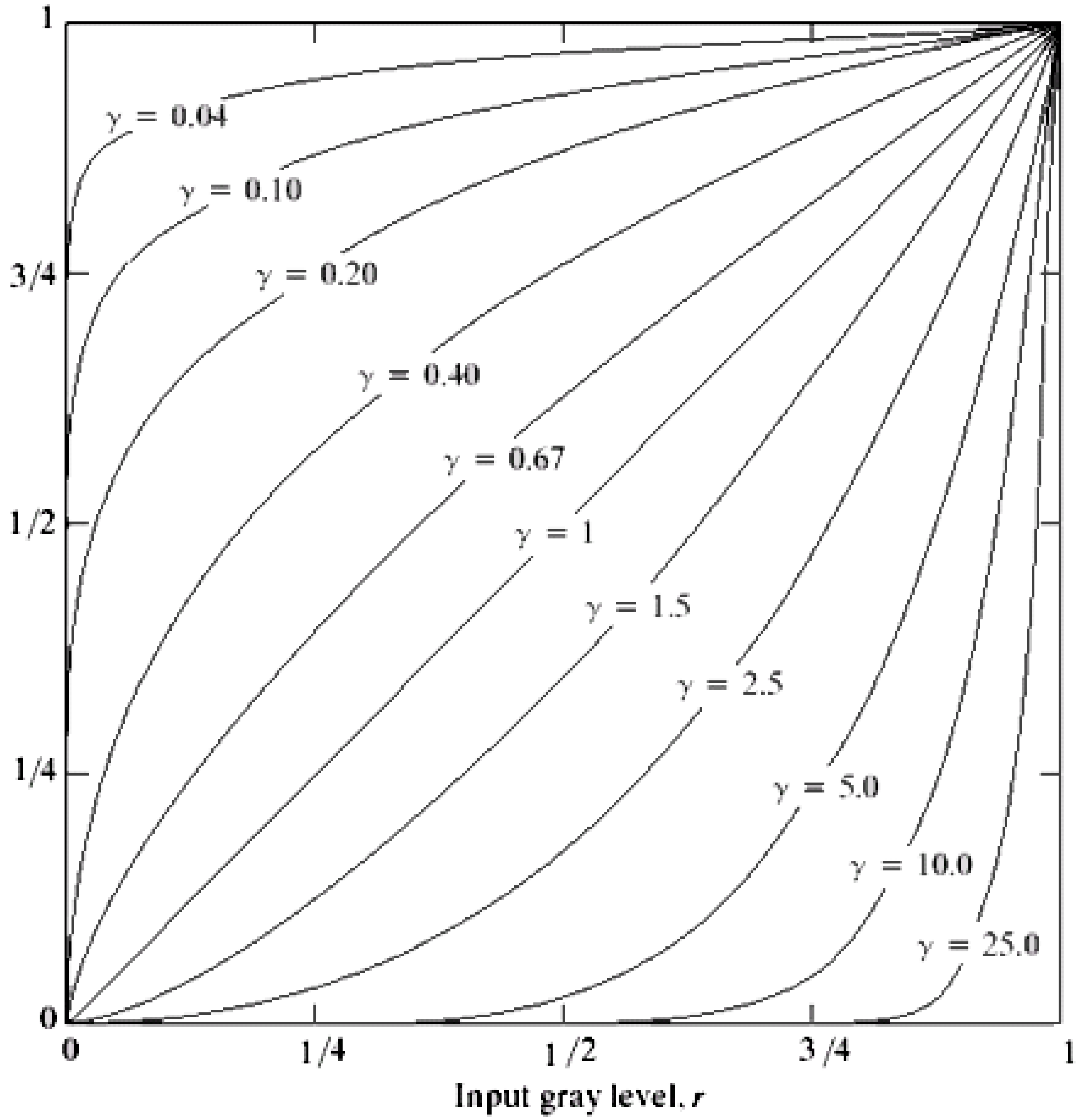
$T(s)$为幂次变换

灰度映射：动态范围压缩

- 幂次变换

$$T_r = r^{\gamma}$$

许多用于图像获取、打印和显示的设备是根据幂次规律响应的。

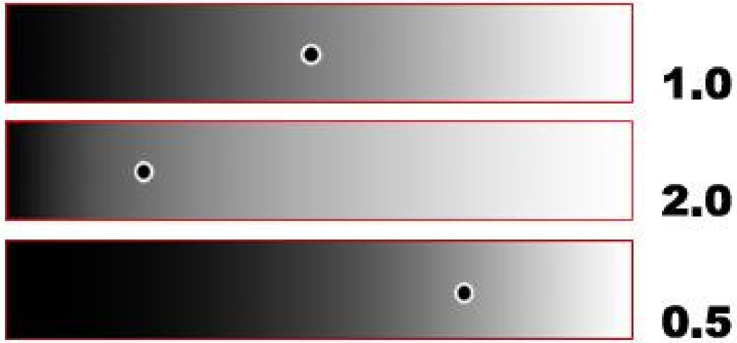


显卡的颜色校正设置

即插即用监视器 和 Mobile Intel(R)...
常规
适配器
监视器
疑难解答
颜色管理
Intel® Graphics Media Accelerator Driver for mobile
显示设备
显示设置
颜色校正
快捷键
笔记本电脑
颜色：
全部
红色
绿色
蓝色
伽马：1.0
1.0
5.0
亮度：0
-60
100
对比度：50
40
100
恢复默认值
确定
取消
应用
3D 设置
图式选项
请选择：

γ 校正

- γ 校正是用于补偿不同输出设备存在的颜色显示差异，从而使图像在不同输出设备上呈现出相同的效果。

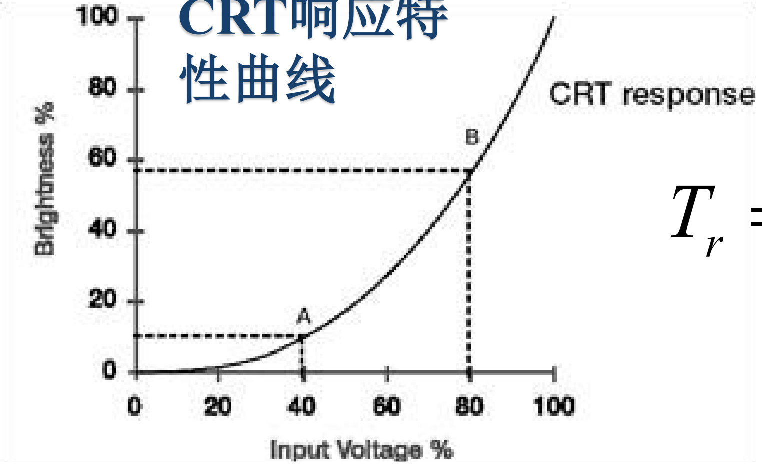


γ=1.0为“理想”显示器；具有从白色-灰色-黑色的连续线性渐变的完美显示效果。

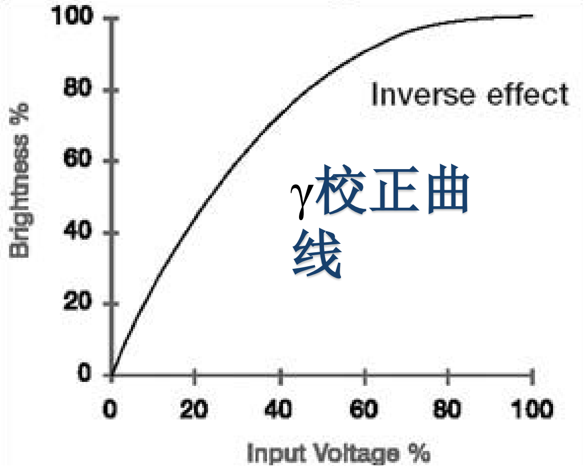
但理想的显示设备是不存在的。通常显示设备都是“非线性”的设备。

NTSC 视频的标准γ=2.2。电脑显示器的γ值一般在 1.5 到 2.0 之间。

γ校正

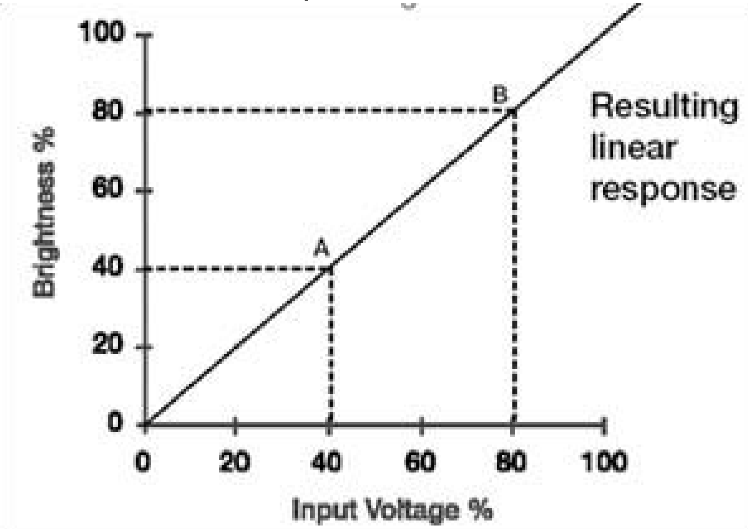


$$T_r = r^{2.5}$$



$$T_r = r^{1/2.5}$$

$$T_r = r^{1}$$



理想状态下的曲线

γ校正

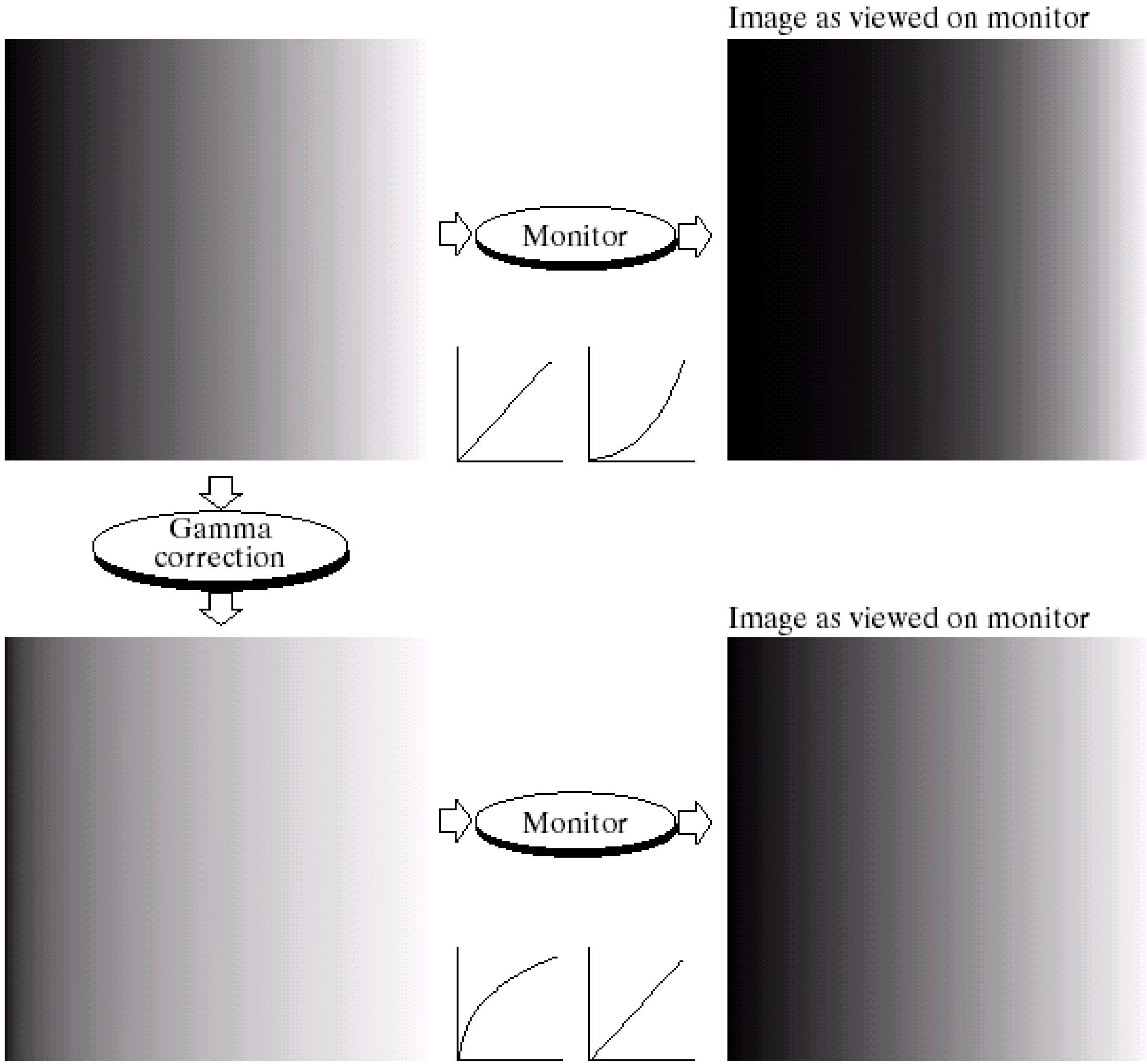
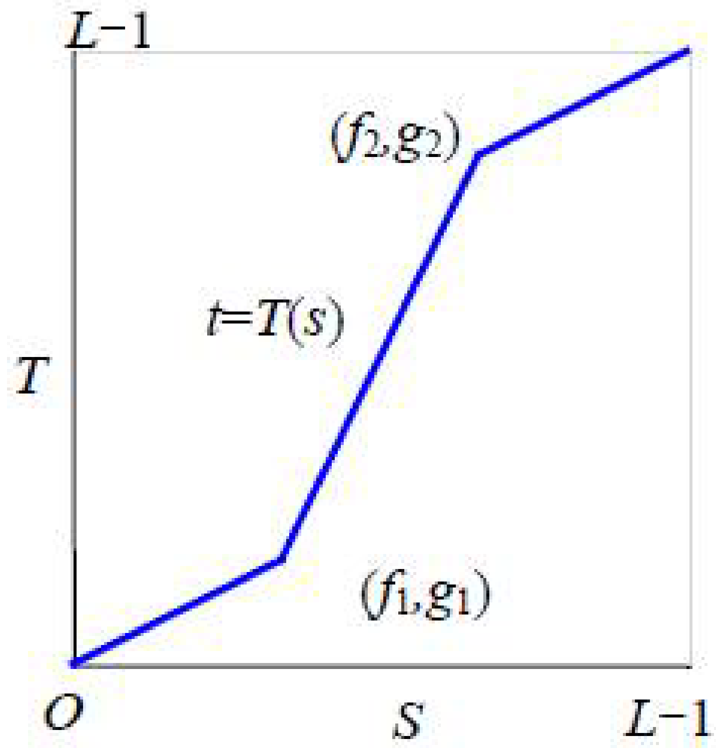


Image as viewed on monitor
Monitor
Gamma correction
Image as viewed on monitor
Monitor

灰度映射：对比度增强

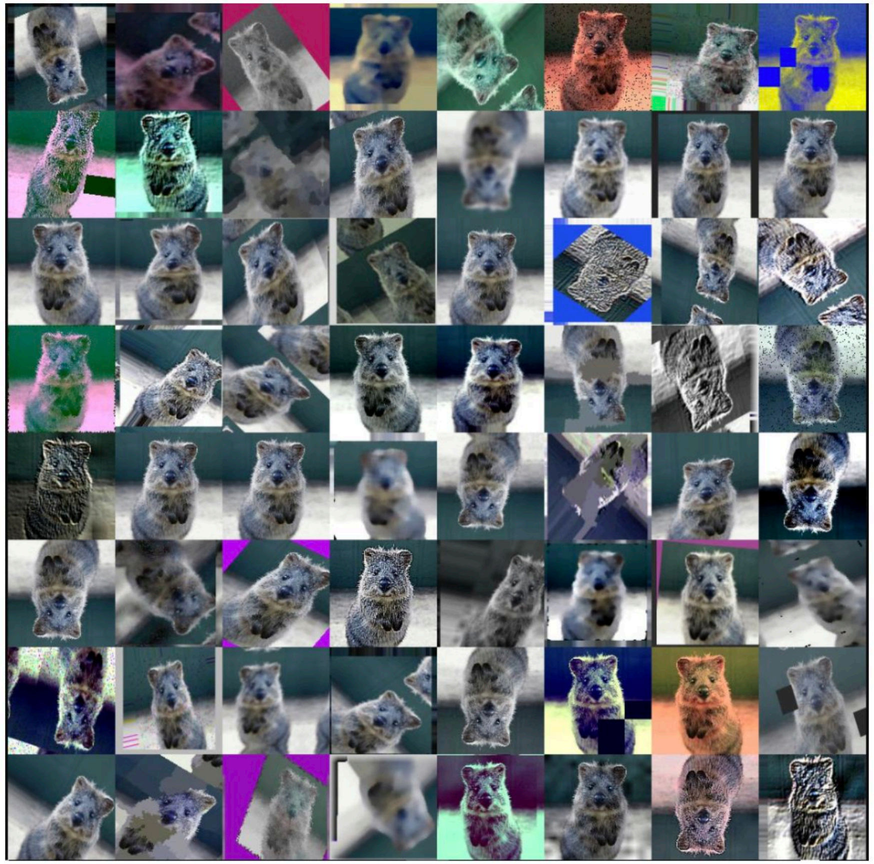


$T(s)$为分段线性函数

例：数据增强

- https://github.com/CrazyVertigo/awesome-data-augmentation

https://github.com/aleju/imgaug



包括：
加噪声
仿射变换
裁剪
翻转
旋转
…

albumentations-team/albumentations

包括
颜色变换
亮度调整
模糊
压缩
黑白
…

3.3 直方图

直方图均衡

直方图


MENU
OK
Nikon
100/289
NIKON D300
A 1/500 F8
200
55mm
PREd-1, A5, G2 AdobeRGB LS H
402ND300 _TR46951. JPG RAW+BASIC
2012/02/02 16:59:13 2144x1424
AE-L
AF-L
AF-ON
L
CF

灰度直方图

- 灰度直方图：图像中各灰度级出现频数分布的统计图表，1D离散函数。
- 设图像总像素个数为n，共有L级灰度，r_k 是第 k 级灰度, n_k 是图像中灰度级为 r_k 的像素数。

  直方图表示：$h(r_k)=n_k$， $k=1, 2, \ldots, L$。

$$\sum_k h(r_k) = n$$

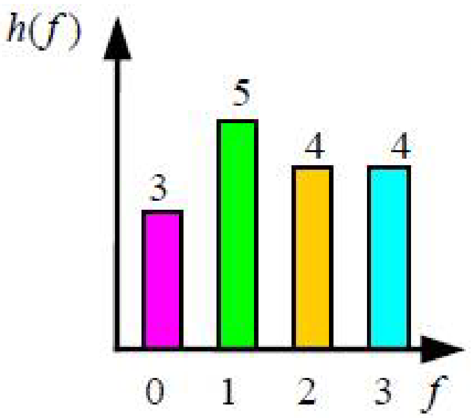
- 直方图归一化

$$\tilde{h}(r_k) = n_k / n$$

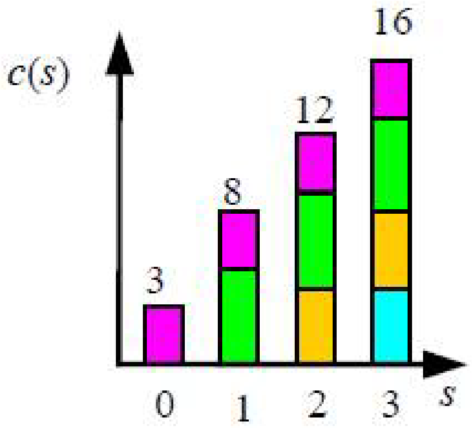
$$\sum_k \tilde{h}(r_k) = 1$$

灰度直方图的意义

0	1	2	3
1	2	3	1
2	3	1	0
3	1	0	2



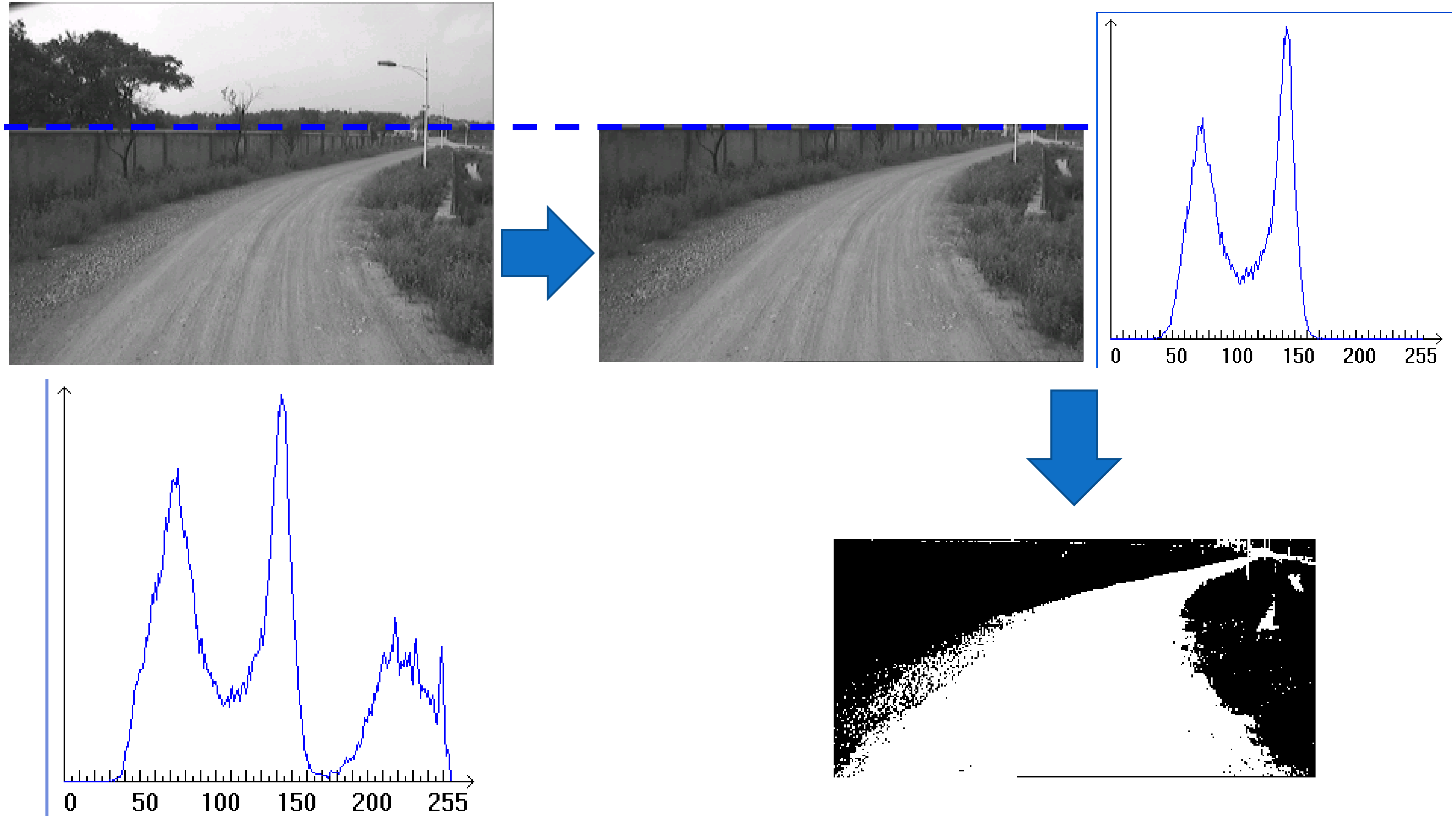
灰度直方图



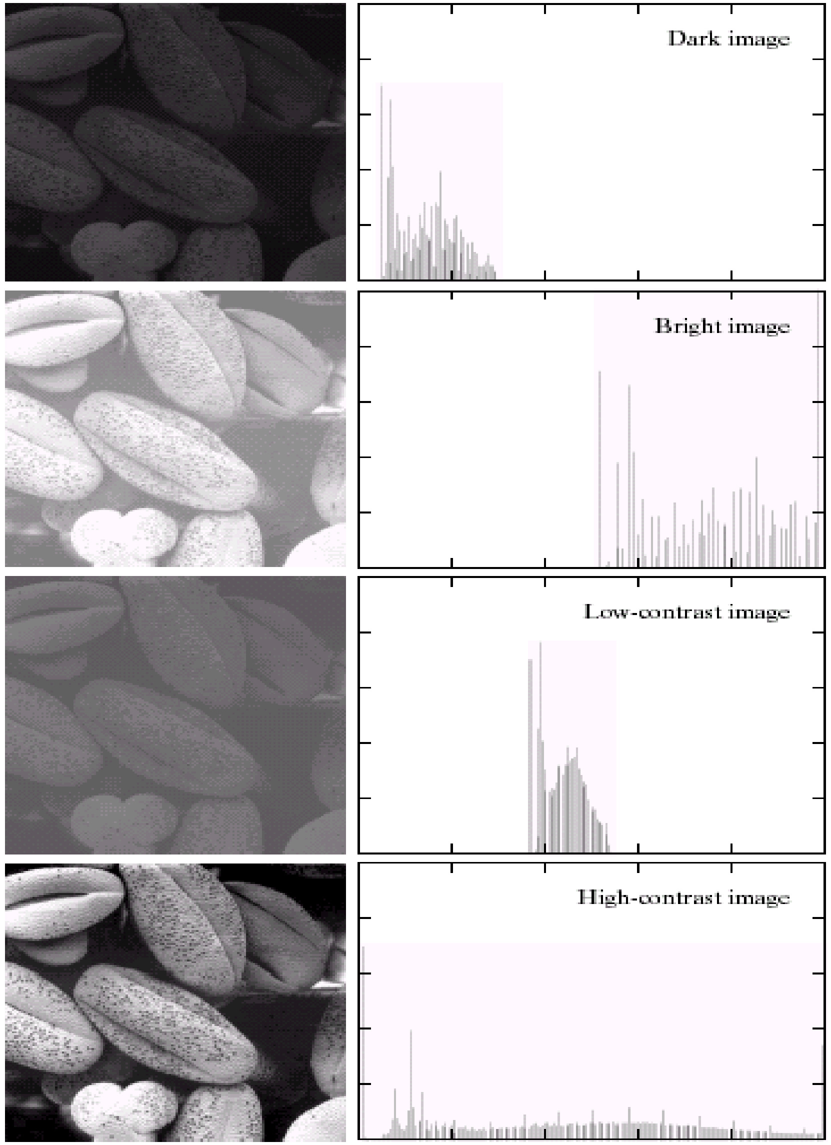
灰度累积直方图

- 反映各灰度级出现频数的分布情况，进而反映图像对比(清晰)度，但不反映各灰度级的空间位置分布。
- **直方图归一化：概率质量函数**
- **累积直方图归一化：累积分布函数**

直方图意义：场景分类



意义：图像质量



- 图像曝光不足，直方图集中在灰度级低的一侧。
- 图像曝光过渡，直方图集中在灰度级高的一侧
- 图像对比度不够，像素只占了整个直方图区域中的很小范围。
- 高质量图像，像素占全部可能的灰度级并分布均匀。

直方图均衡

- 基本思想:
  - 将原始图直方图变换为在整个灰度范围内均匀分布。
- 目的：
  - 增加像素灰度值的动态范围，增强图像整体对比度。
- 思路：
  - 寻找灰度映射函数$T(\cdot)$，有 $s_k = T(r_k)$
  - 要求 $\tilde{h}(s_k)$ 为均匀分布。

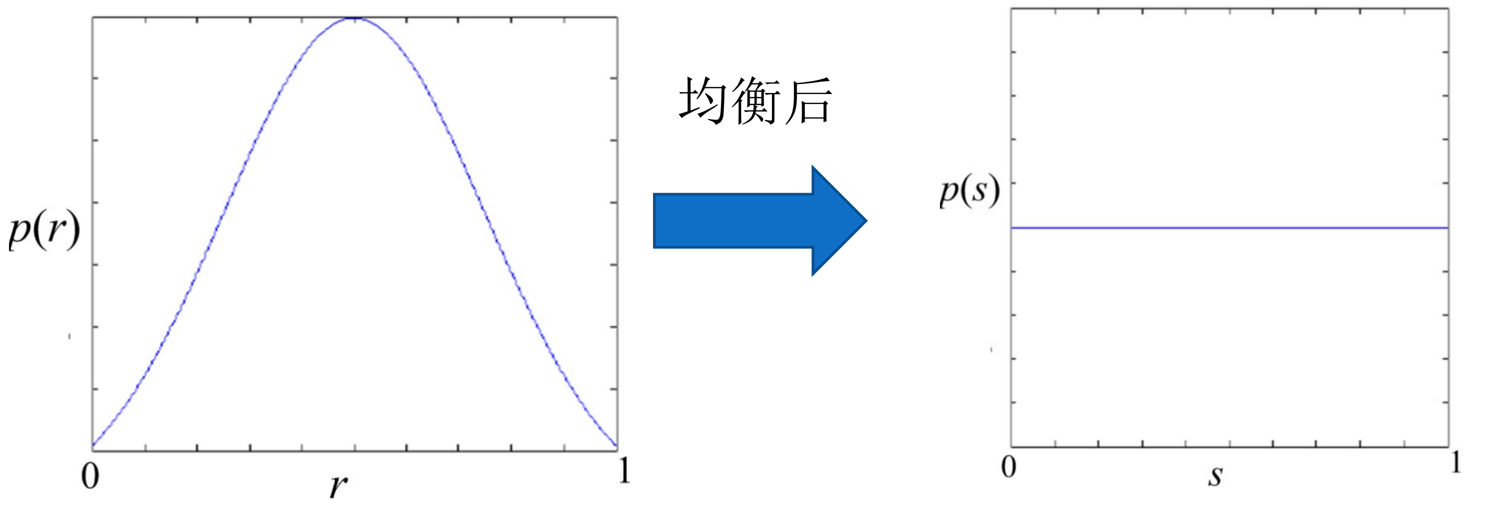
直方图均衡原理

- 灰度映射函数$T(\cdot)$，有 $s_k = T(r_k)$
- 要求：
  - 变换后的灰度仍保持从黑到白的单一变化顺序
  - 变换后灰度范围与原先一致。

- 满足约束条件
  - 在$0 \le r \le 1$ 中，$T(r)$是单调递增函数，且$0 \le T(r) \le 1$
  - 反变换 $r = T^{-1}(s)$， $T^{-1}(s)$ 也为单调递增函数，

直方图均衡原理

- 由于s_k取值不一定为整数，因此先考虑连续概率分布情况，再离散化。

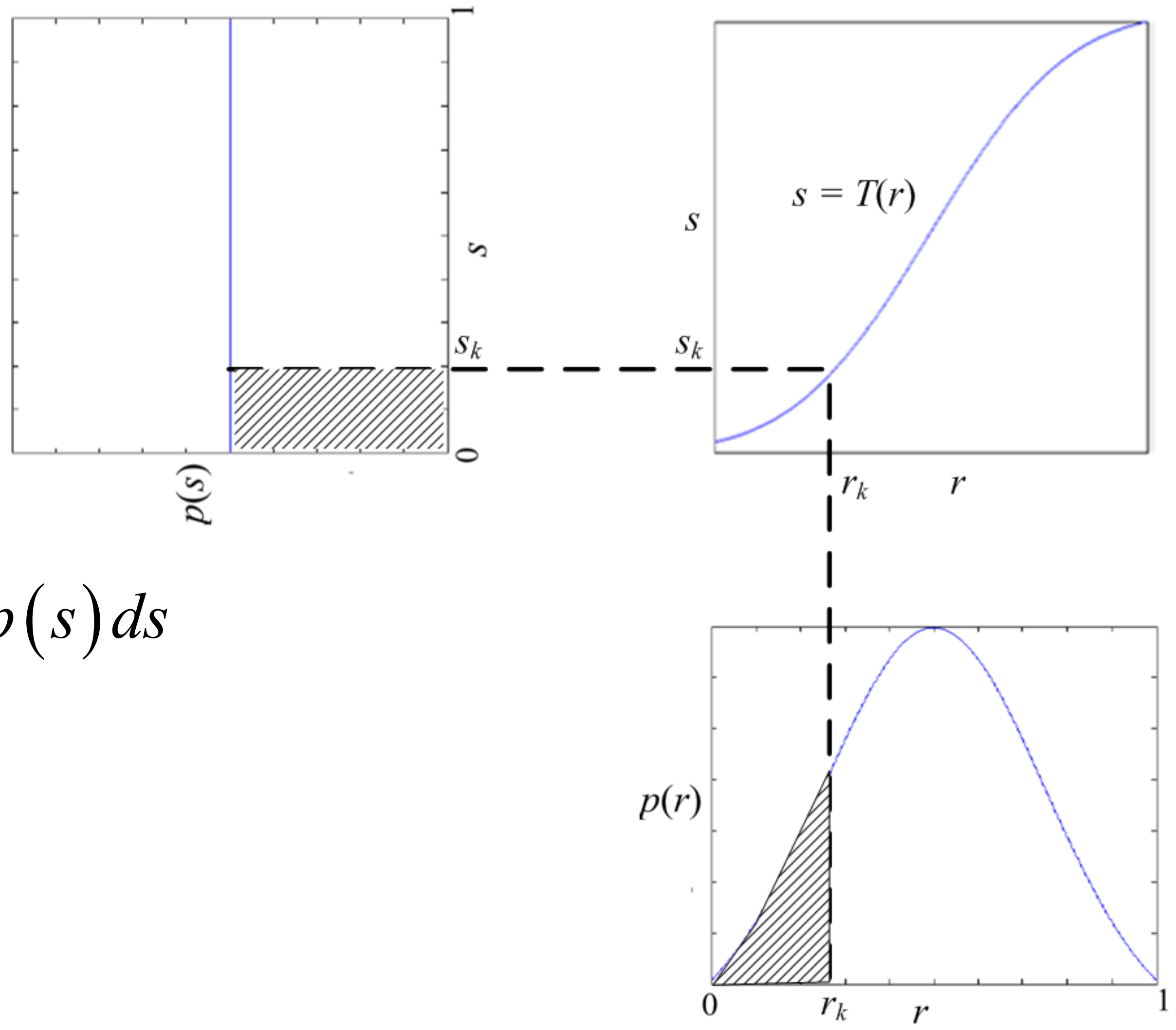


- 概率分布函数：

$$f(r_k)=\int_0^{r_k} p(r)dr \;\; ; \;\; f(s_k)=\int_0^{s_k} p(s)ds$$

直方图均衡原理

- 变换后图像在 $[0,s_k]$灰度级范围内像素面积 ＝ 原图像在$[0,r_k]$灰度级范围内像素面积。



$$f(r_k)=\int_0^{r_k} p(r)\,dr = f(s_k)=\int_0^{s_k} p(s)\,ds$$

$$s_k = T(r_k) = \int_0^{r_k} p(r)\,dr \quad p(s)=1$$

$$\int_0^{s_k} ds = \int_0^{r_k} p(r)\,dr$$

直方图均衡原理

- 连续模型下直方图均衡公式： $$T(r_k)=\int_0^{r_k} p(r)\,dr$$

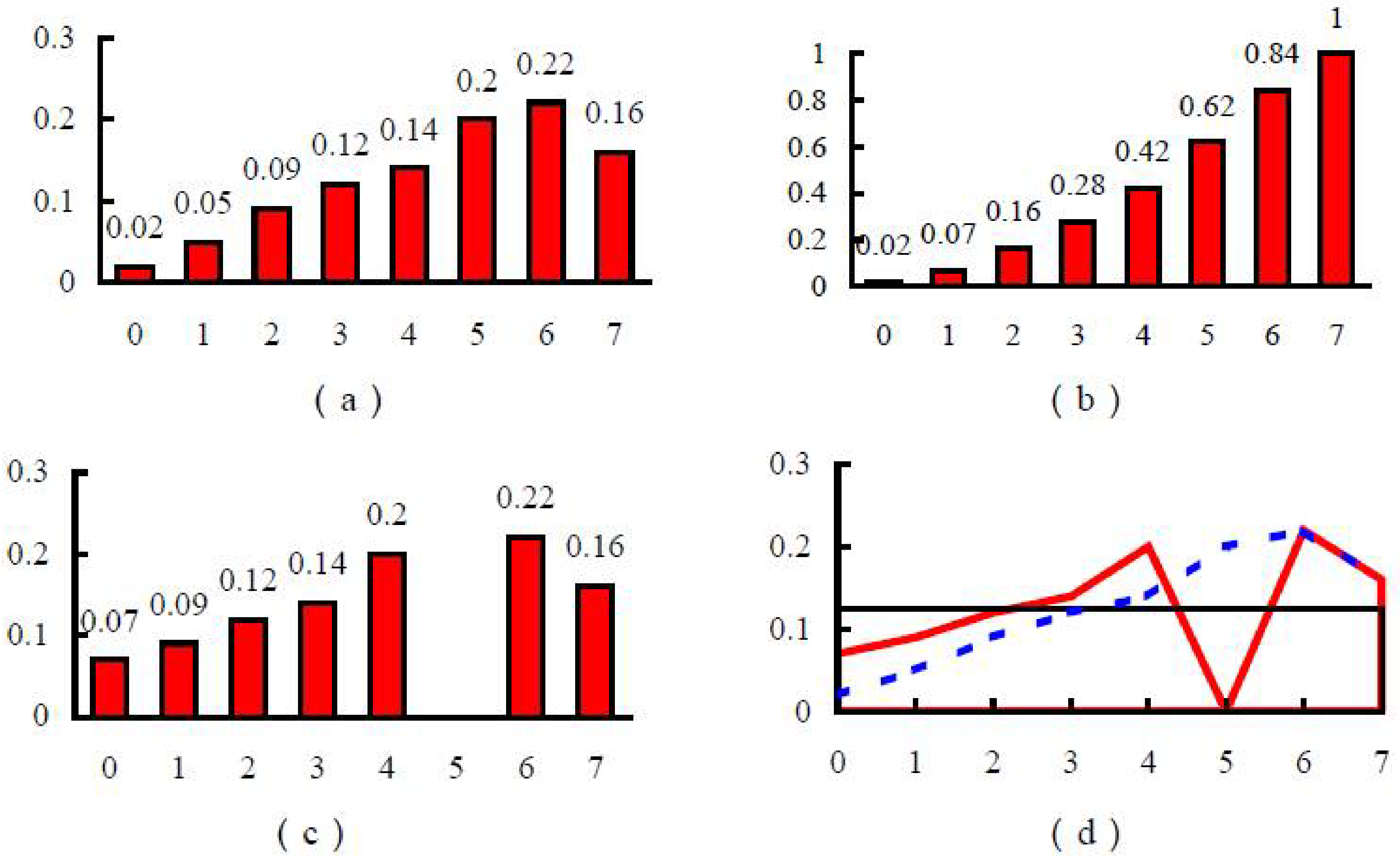
- 离散化：

$$\boxed{T(r_k)=\sum_{j=0}^{k} p(r_j) \quad =\sum_{j=0}^{k}\tilde{h}(r_j)=\sum_{j=0}^{k}\frac{n_j}{n}}$$

- 一般不能证明这一离散变换能产生离散均匀概率密度函数（均匀直方图）。但是这一离散变换的确有展开输入图像直方图的趋势。

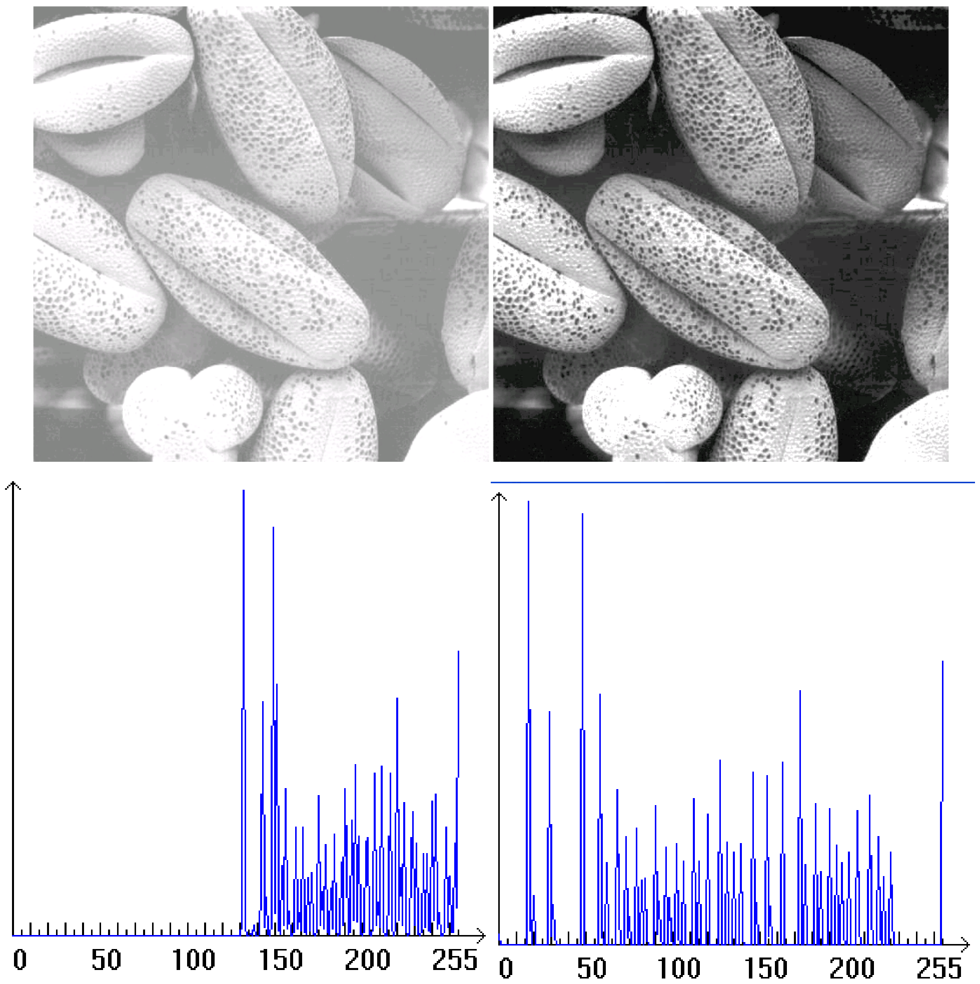
Hangzhou Dianzi University 杭州电子科技大学 School of Computer Science and Technology 计算机学院 周文晖

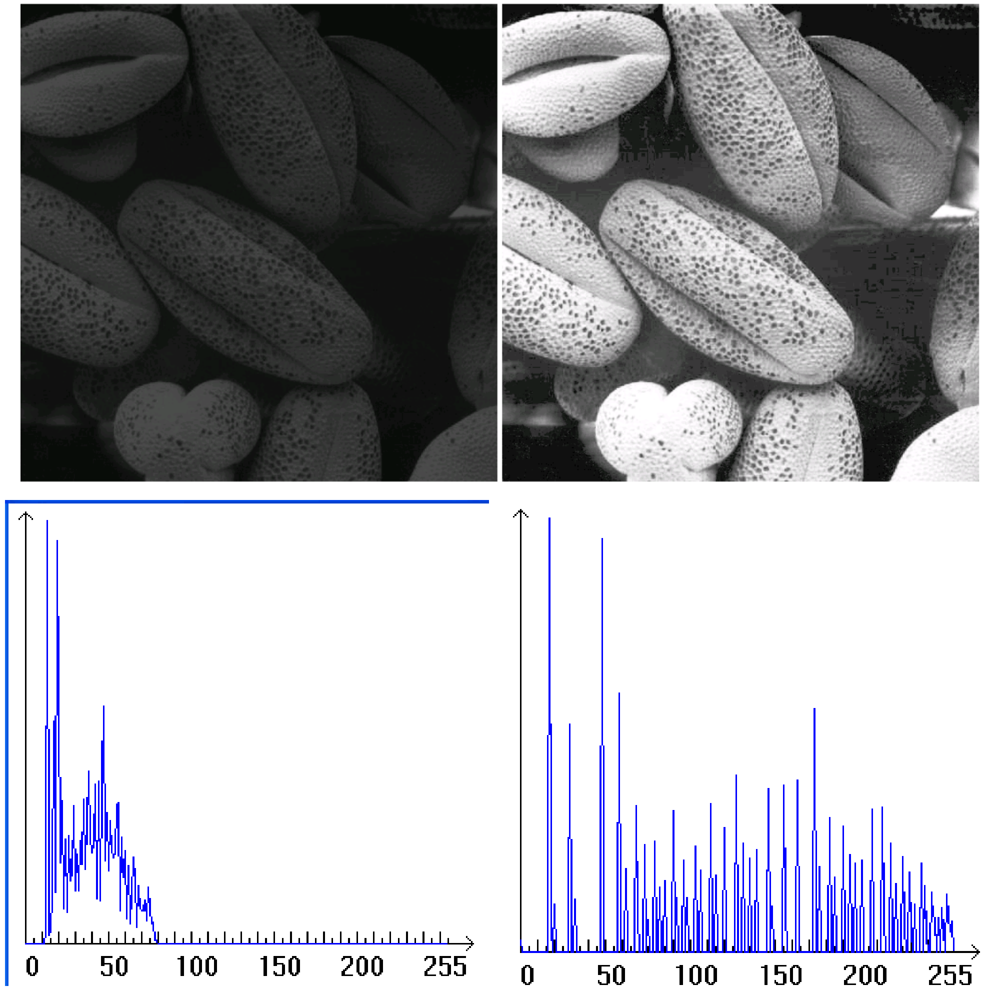
直方图均衡过程示例



映射到灰度级：	确定映射关系：
0->0.14	0->0
1->0.49	1->0
2->1.12	2->1
3->1.96	3->2
4->2.94	4->3
5->4.34	5->4
6->5.88	6->6
7->7.00	7->7

直方图均衡结果


0 50 100 150 200 255
0 50 100 150 200 255


0 50 100 150 200 255
0 50 100 150 200 255

3.3 直方图

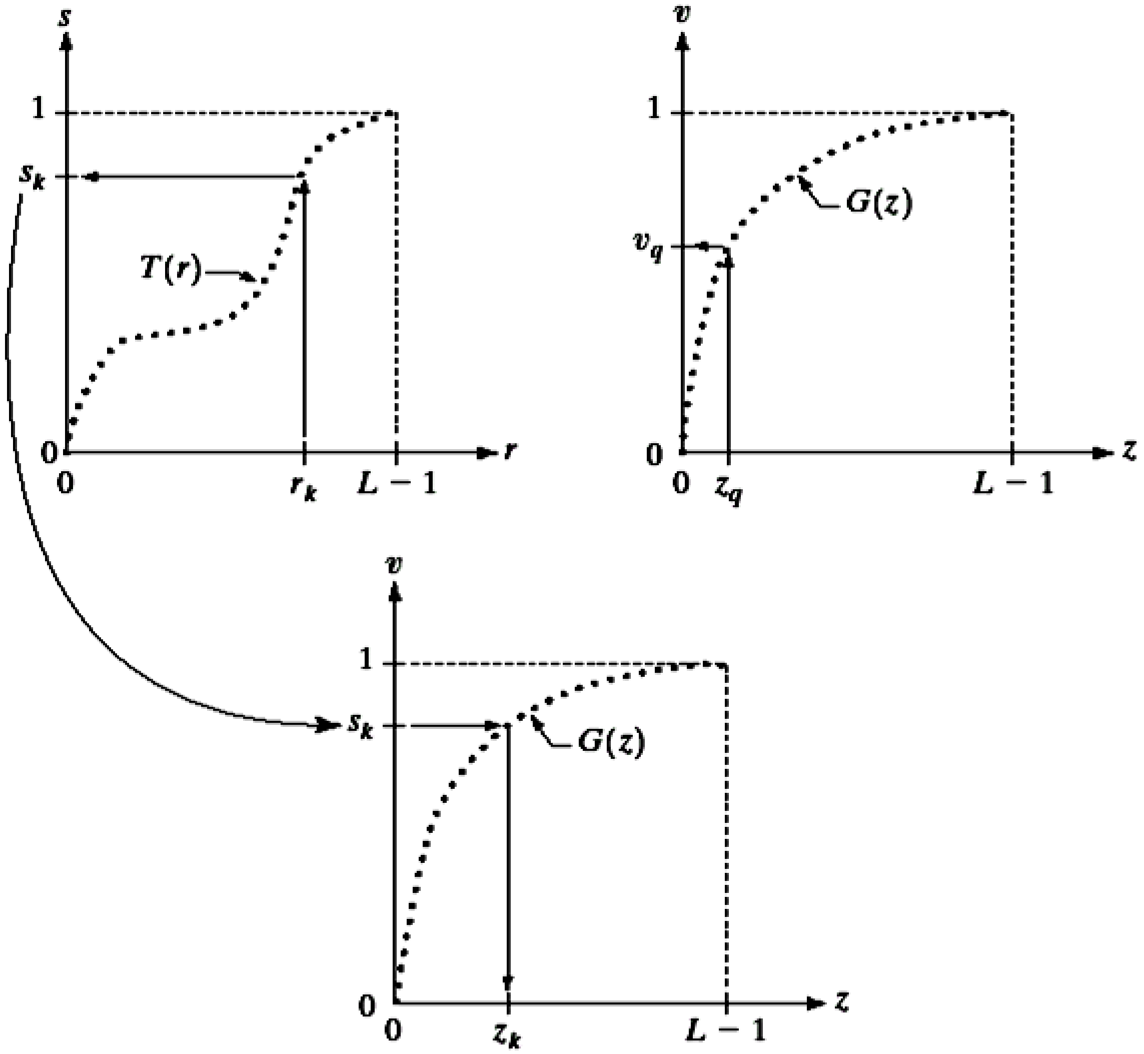
直方图规定化

直方图规定化

- 直方图均衡：自动增强整幅图像的对比度
- 直方图规定化：实现指定的直方图分布

- 思路：
  - 借助直方图均衡，

    即 **均衡后图像相等**



直方图规定化：连续模型

- 令 r 和 z 分别代表连续的输入、输出图像的灰度级。
- 从输入图像估计$P_r(r)$，$P_z(z)$为希望输出图像所具有的规定概率密度函数。

$$s_m = T(r_m) = \int_0^{r_m} p(r)dr \qquad v_n = G(z_n) = \int_0^{z_n} p(z)dz$$

输入图像直方图均衡　　　　指定直方图的均衡化

$$s = T(r) = G(z) = v$$

$$z_k = G^{-1}(s_k) = G^{-1}[T(r_k)]$$

直方图规定化：离散模型

- 步骤1：对原始输入图像进行直方图均衡

$$s_m = T(r_m) = \sum_{j=0}^{m} P_r(r_j) = \sum_{j=0}^{m} \frac{n_j}{n} \qquad m = 0,1,2,\ldots,L-1$$

- 步骤2：根据指定的直方图分布，进行直方图均衡

$$v_n = G(z_n) = \sum_{i=0}^{n} P_z(z_i) \qquad n = 0,1,2,\ldots,L-1$$

- 步骤3：求步骤2的反变换，将原始直方图对应映射到规定直方图

$$z_k = G^{-1}(s_k) = G^{-1}[T(r_k)] \qquad k = 0,1,2,\ldots,L-1$$

直方图规定化

$$z_k = G^{-1}(s_k) = G^{-1}[T(r_k)] \quad k = 0,1,2,\ldots,L-1$$

- $G^{-1}(\cdot)$ 难以获得，但 s_m 和v_n 可获得

$$s_m = \sum_{j=0}^{m} P_r(r_j) \qquad v_n = G(z_n) = \sum_{i=0}^{n} P_z(z_i)$$

- 若 $v_n \approx s_m$ ，将第 m 个灰度级投影到第 n 个灰度级。
- 单映射规则 (single mapping law / SML)：

$$\min\left|\sum_{j=0}^{m} P_r(r_j) - \sum_{i=0}^{n} P_z(z_i)\right| \quad \begin{matrix} i = 0,1,2,\ldots,L-1 \\ j = 0,1,2,\ldots,L-1 \end{matrix}$$

直方图规定化举例

- 给定图像具有64×64个像素，8个灰度级
- 其分布如下表，试按表中规定直方图进行变换

原始图像灰度级	0/7	1/7	2/7	3/7	4/7	5/7	6/7	7/7
原始图像各灰度级像素	790	1023	850	656	329	245	122	81
规定的直方图	0	0	0	0.15	0.20	0.30	0.20	0.15

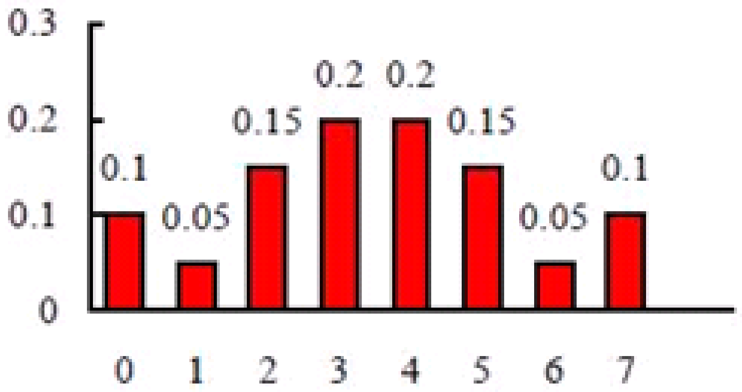
直方图规定化举例

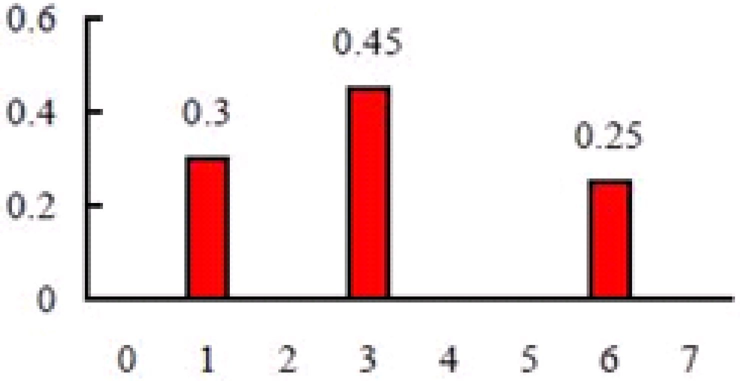
原始图像灰度级	0/7	1/7	2/7	3/7	4/7	5/7	6/7	7/7
原始图像各灰度级像素	790	1023	850	656	329	245	122	81
计算原始直方图	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02
原始累积直方图	**0.19**	**0.44**	**0.65**	**0.81**	**0.89**	**0.95**	**0.98**	**1.0**

规定直方图	0	0	0	0.15	0.20	0.30	0.20	0.15
规定累积直方图	**0**	**0**	**0**	**0.15**	**0.35**	**0.65**	**0.85**	**1.0**

SML映射	3	4	5	6	6	7	7	7
确定映射关系	**0→3**	**1→4**	**2→5**	**3,4→6**		**5,6,7→7**		
变换后直方图	0	0	0	0.19	0.25	0.21	0.24	0.11

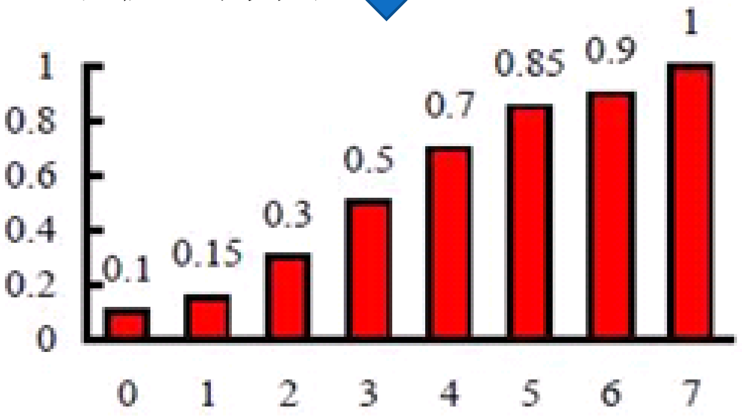
直方图规定化举例



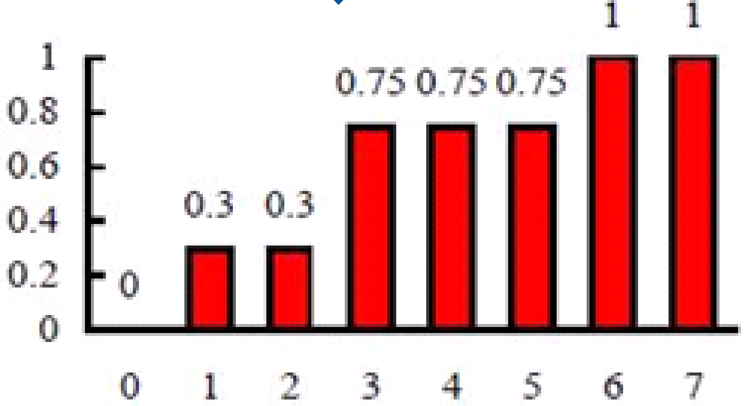


累积直方图

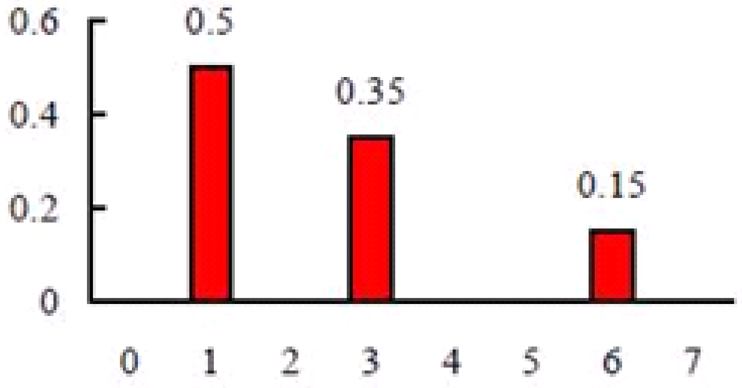
累积直方图



(c)



(d)



(e)

映射关系：0,1,2,3 → 1
4,5 → 3
6,7 → 6

3.4 空域滤波

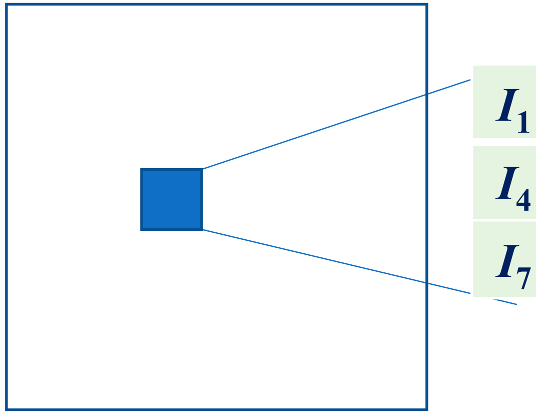
模板滤波

空域滤波概念

- 概念：利用像素本身以及其邻域像素的灰度关系进行图像增强的方法。
- 滤波取自信号处理中的概念
- 空域滤波是在图像空间通过**邻域操作**完成的。
- 邻域操作通常借助**模板运算**来实现

模板滤波

令F表示图像



I_1	I_2	I_3
I_4	I_5	I_6
I_7	I_8	I_9

$*$

令H表示模板

h_1	h_2	h_3
h_4	h_5	h_6
h_7	h_8	h_9

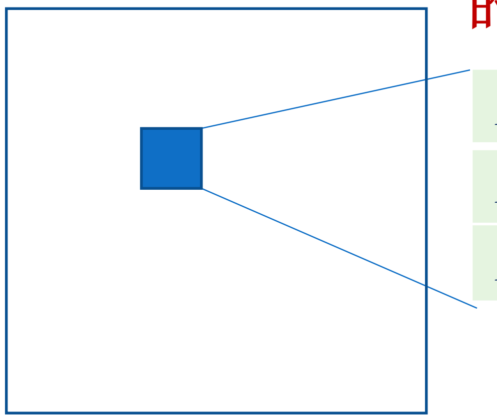
3×3模板

- 模板滤波：图像与模板的卷积

$$F' = F * H$$

卷积计算

以 I_5 为中心的邻域区域



I_1	I_2	I_3
I_4	$\boxed{I_5}$	I_6
I_7	I_8	I_9

$*$

h_1	h_2	h_3
h_4	h_5	h_6
h_7	h_8	h_9

卷积计算

$$I_5' = h_9 \cdot I_1 + h_8 \cdot I_2 + h_7 \cdot I_3 + h_6 \cdot I_4 + h_5 \cdot I_5 + h_4 \cdot I_6 + h_3 \cdot I_7 + h_2 \cdot I_8 + h_1 \cdot I_9$$

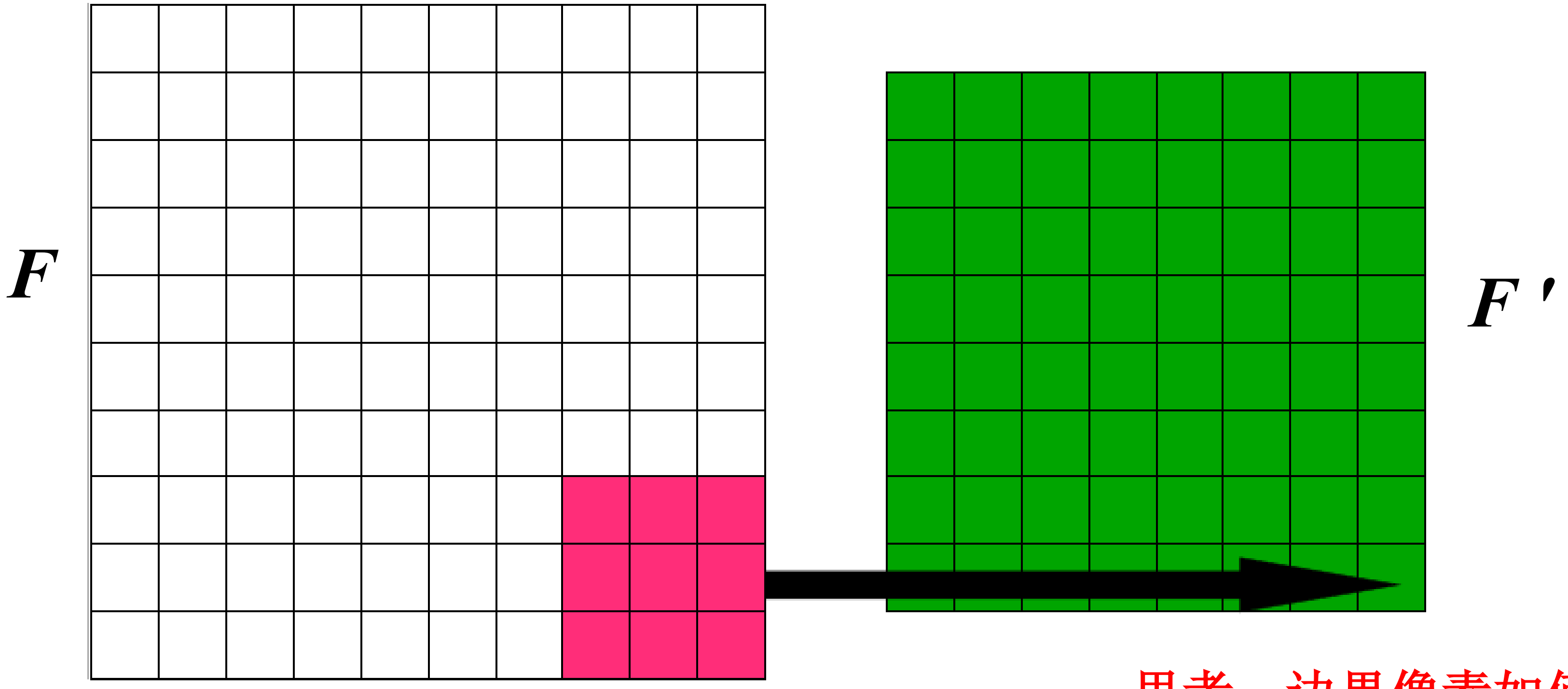
- 由于模板通常都是中心对称的，即可忽略模板以中心反转的过程，有

$$I_5' = h_1 \cdot I_1 + h_2 \cdot I_2 + h_3 \cdot I_3 + h_4 \cdot I_4 + h_5 \cdot I_5 + h_6 \cdot I_6 + h_7 \cdot I_7 + h_8 \cdot I_8 + h_9 \cdot I_9$$

Hangzhou Dianzi University 杭州电子科技大学 *School of Computer Science and Technology* 计算机学院 周文晖

模板滤波

- 模板滤波过程：遍历图像中所有像素，计算每个像素的邻域与模板的卷积值。



思考：边界像素如何处理?

模板滤波的程序实现

```
for ( j = 0; j < height; j++ ) {
    for ( i = 0; i < width; i++ ) {
        sum = 0;
        for ( n = -win_radius; n <= win_radius; n++ ) {
            for ( m = -win_radius; m <= win_radius; m++ ) {
                sum += Img[j-n][i-m] * Template[n+win_radius][m+win_radius];
            }
        }
    }
}
```

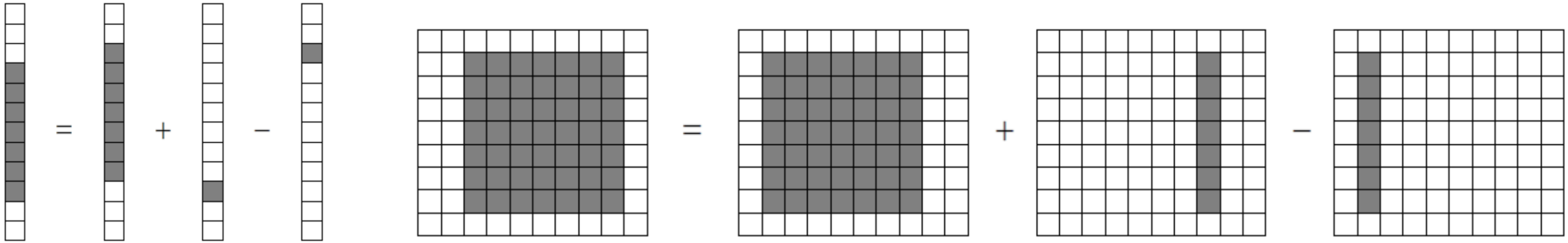
Box Filtering

- McDonnel 提出了盒滤波（Box Filtering）技术，采用递归的二维均值滤波快速计算方法

[1] McDonnel.M.J. Box-Filtering Techniques. Computer Graphics and Image Processing, 1981(17):65-70.

[2] O. Faugeras, , B. Hotz H. Mathieu, T. Vieville, Z Zhang, P. Fua, E. Theron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy, Real-time correlation-based stereo: algorithm, implementations and applications, INRIA Technical Report #2013, August 1993

[3] Karsten, M. Dennis, D. Hesser, J. Reinhard, M. Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation. Proc. of IEEE Workshop on Stereo and Multi-Baseline Vision, 2001.



积分图与Box Filtering

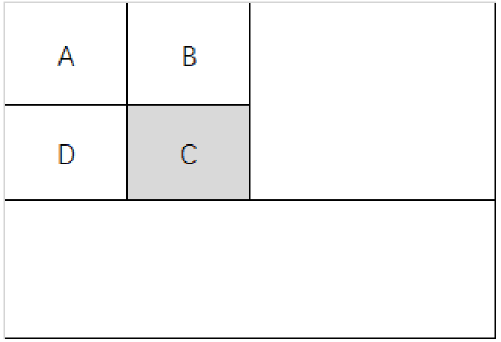
- **积分图：**积分图中任意位置的值是由原图中该位置和其左上角所有位置值的和。

$$\mathbf{I}(x,y) = \mathbf{i}(x,y) + \mathbf{I}(x-1,y) + \mathbf{I}(x,y-1) - \mathbf{I}(x-1,y-1)$$

i为原图，**I**为积分图

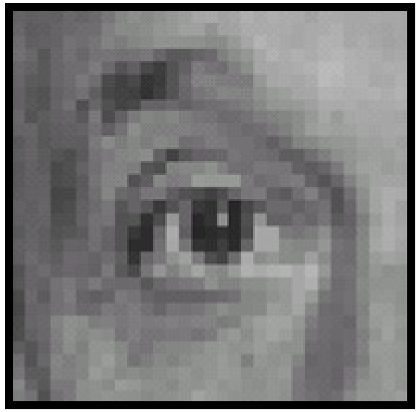
原图						积分图				
1	2	3	2	4		1	3	6	8	12
0	5	1	7	2		1	8	12	21	27
3	1	5	9	8		4	12	21	39	53
5	2	6	2	1		9	19	34	54	69
1	0	8	5	4		10	20	43	68	87

- 当得到积分图之后，阴影区域的和即可计算得到。该阴影可以是任意形状矩形



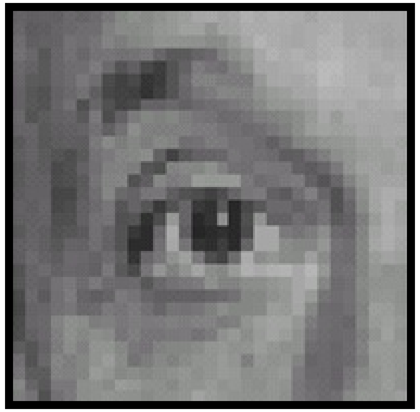
$$\sum_{\substack{A(x)<x'\leq C(x)\\ A(y)<y'\leq C(y)}} i(x',y') = I(C_{right\_bottom}) + I(A_{right\_bottom}) - I(B_{right\_bottom}) - I(D_{right\_bottom})$$

模板滤波举例



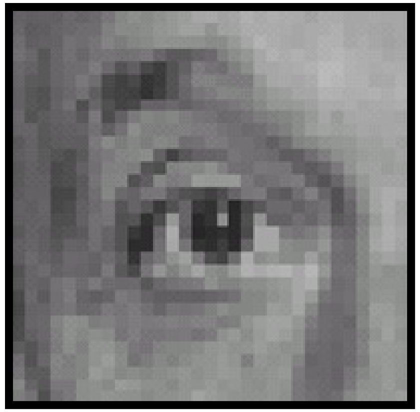
原图像

$$* \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} =$$



相同图像

模板滤波举例：平移

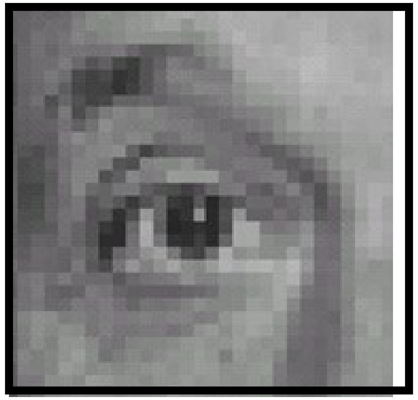


原图像

*

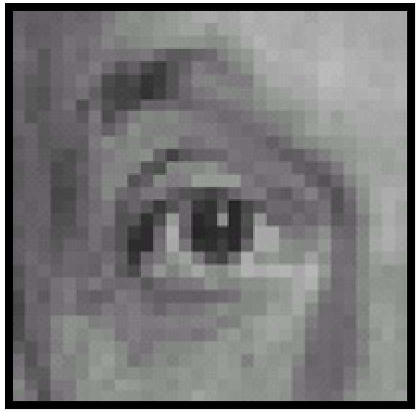
0	0	0
1	0	0
0	0	0

=



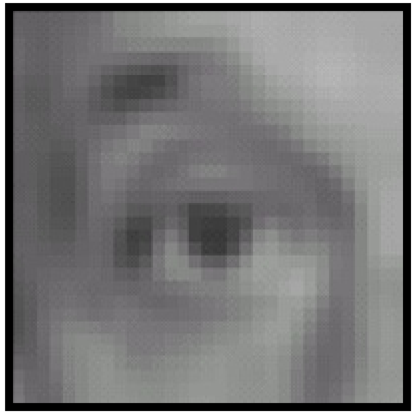
图像左移一个像素

模板滤波举例：邻域平均



原图像

$$* \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} =$$



图像模糊
均值滤波

模板滤波举例：邻域平均

原图

(a)

叠加均匀分布随机噪声

(b)

3×3平滑模板

(c)

5×5平滑模板

(d)

(e)

7×7平滑模板

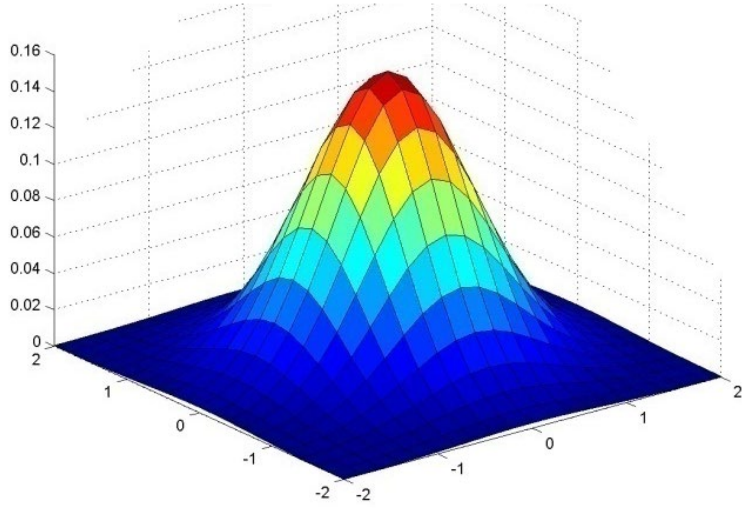
(f)

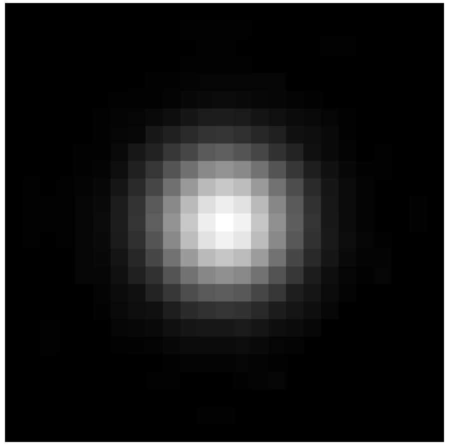
9×9平滑模板

(g)

11×11平滑模板

模板滤波举例：加权平均(高斯滤波)





$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

高斯模板计算

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

$$x = 0; y = 0; \sigma = 1 \rightarrow G_0 = \frac{1}{2\pi} \approx 0.159$$

0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	$\boxed{0.159}$	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5×5, σ = 1

$\frac{1}{273}$ ×

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

5×5, σ = 1 高斯模板

类似有：

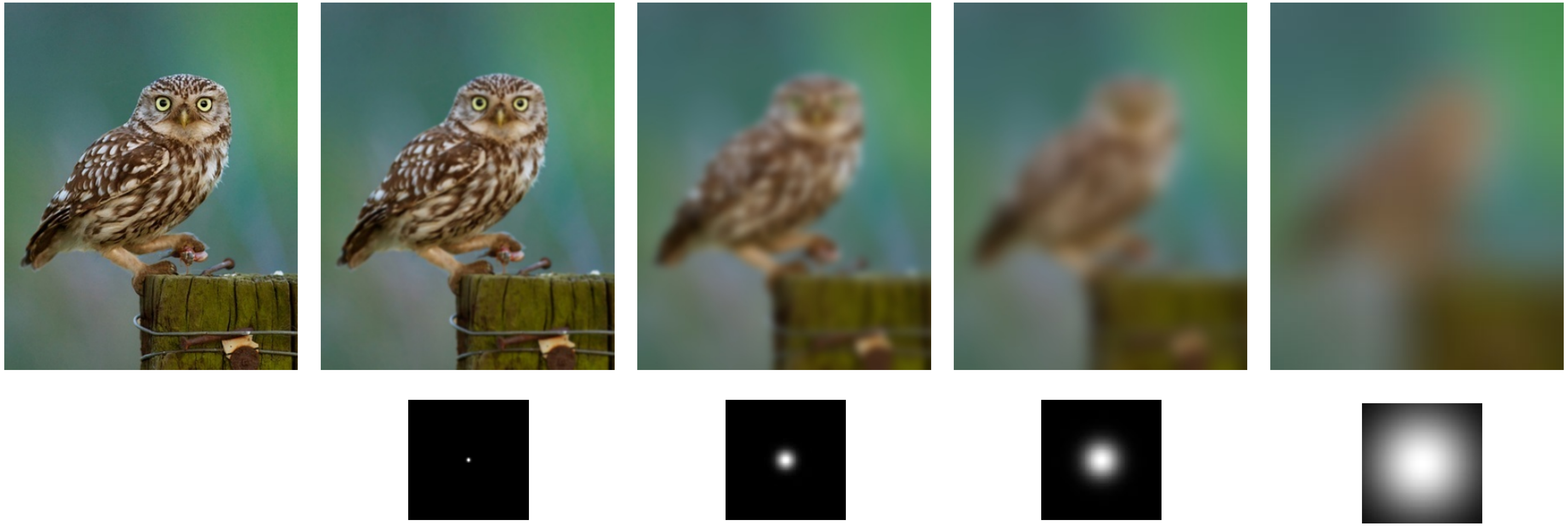
$\frac{1}{16}$ ×

1	2	1
2	4	2
1	2	1

3×3, σ = 0.5 高斯模板

优势：可以用移位代替乘积。

模板滤波举例：高斯滤波



σ = 1 pixel　σ = 5 pixels　σ = 10 pixels　σ = 30 pixels

滤除图像中的高频分量

高斯滤波特点

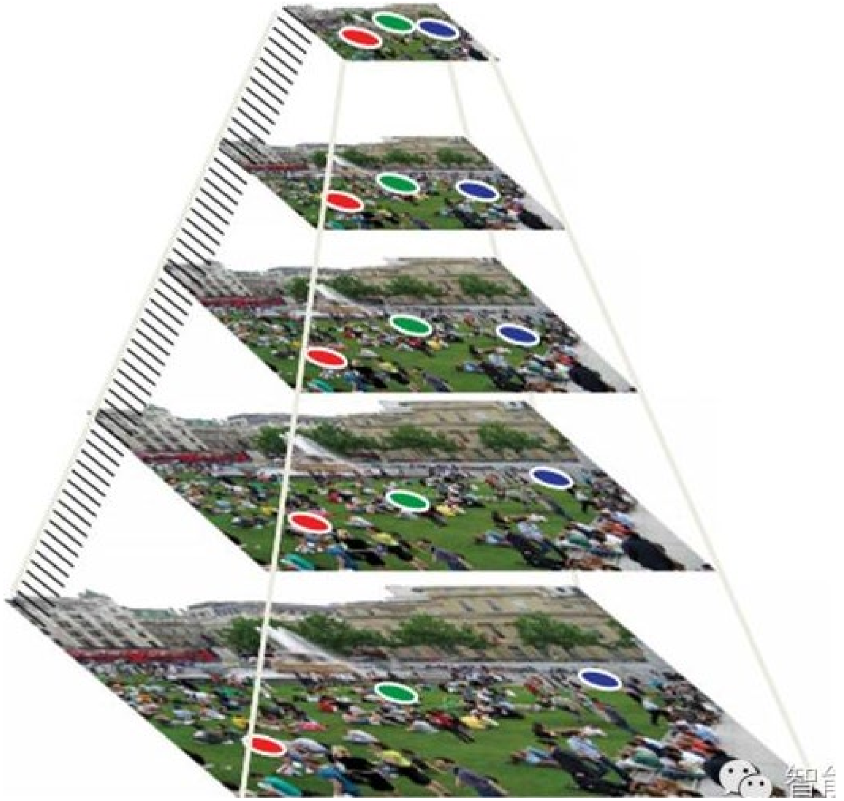
- 二维高斯模板可分解为两个一维的高斯模板相乘
  - 水平高斯模板和垂直高斯模板

$$
\begin{aligned}
G_\sigma(x,y) &= \frac{1}{2\pi\sigma^2}\exp^{-\frac{x^2+y^2}{2\sigma^2}} \\
&= \left(\frac{1}{\sqrt{2\pi}\sigma}\exp^{-\frac{x^2}{2\sigma^2}}\right)\left(\frac{1}{\sqrt{2\pi}\sigma}\exp^{-\frac{y^2}{2\sigma^2}}\right)
\end{aligned}
$$

$$
\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}
$$

高斯金字塔（Gaussian Pyramid）

- 金字塔的底层为原始图像，每向上一层则是通过高斯滤波和1/2采样得到



模板滤波举例：锐化（图像增强）

−

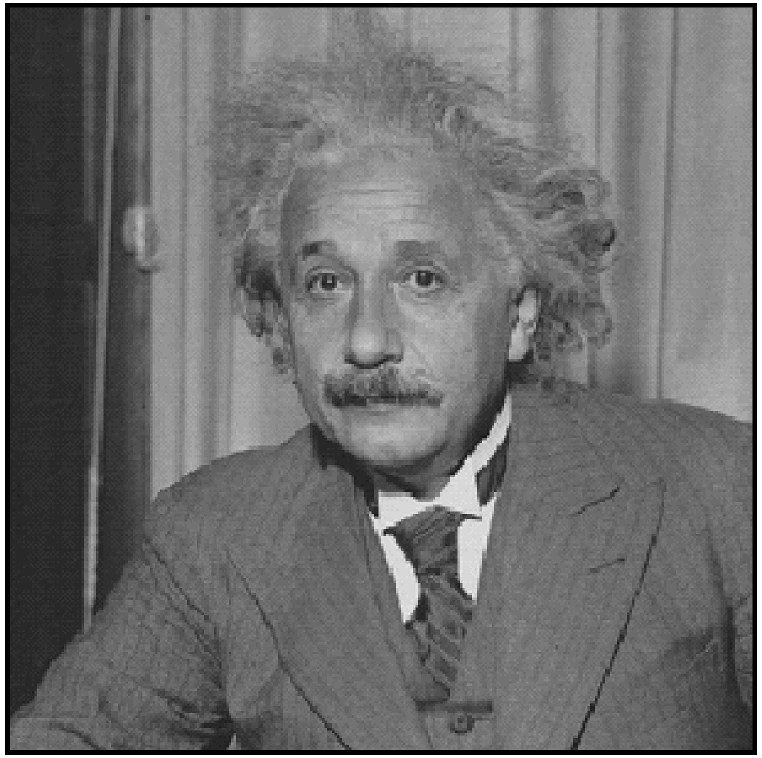
=

Let’s add it back:

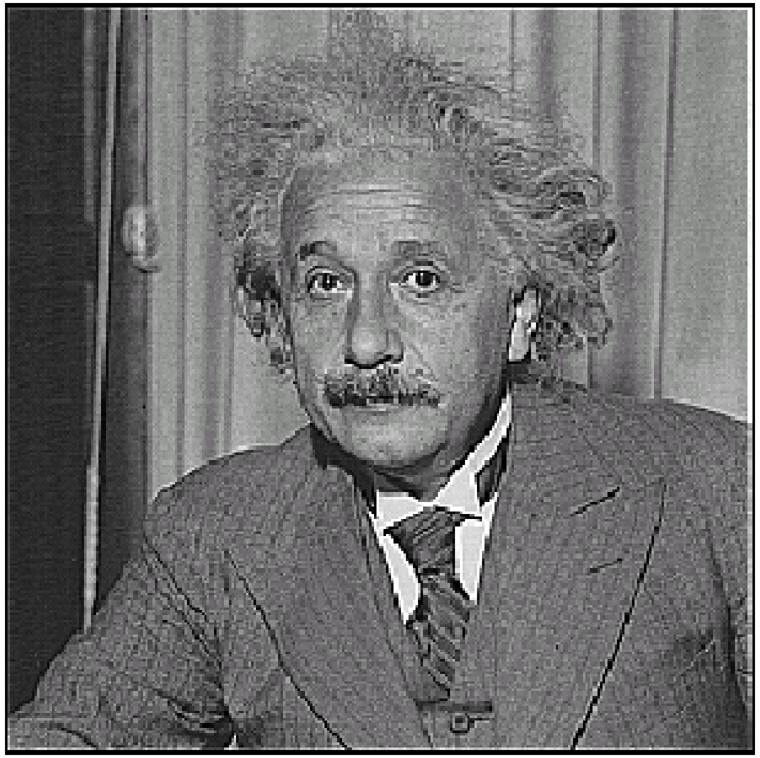
$+ \alpha$

=

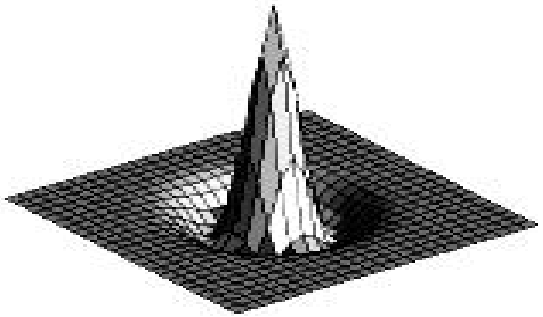
模板滤波举例：锐化（图像增强）



before



after



模板滤波举例：锐化（图像增强）

• 拉普拉斯算子

0	1	0
1	-4	1
0	1	0

（a）拉普拉斯算子

1	1	1
1	-8	1
1	1	1

（b）扩展的拉普拉斯算子

0	-1	0
-1	4	-1
0	-1	0

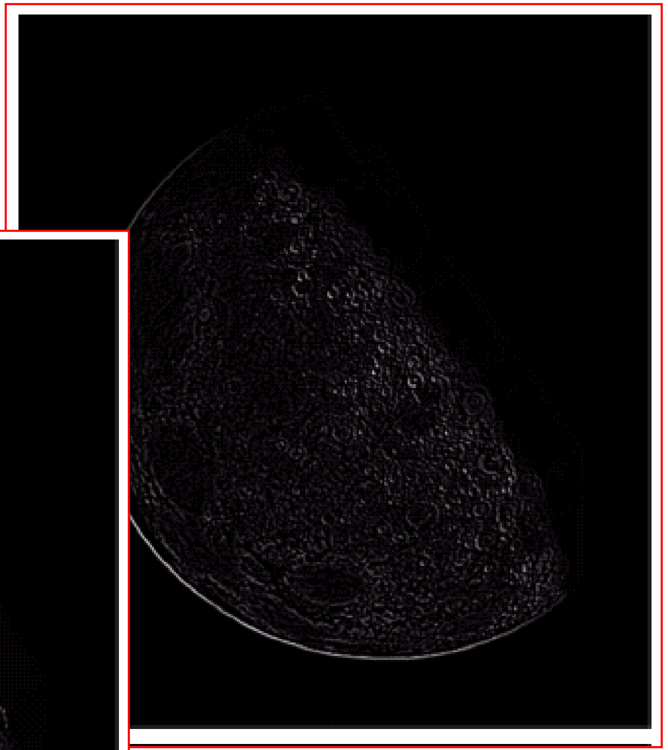
（c）中心系数为正的拉普拉斯算子

-1	-1	-1
-1	8	-1
-1	-1	-1

（d）中心系数为正的扩展的拉普拉斯算子

拉普拉斯图像增强原理

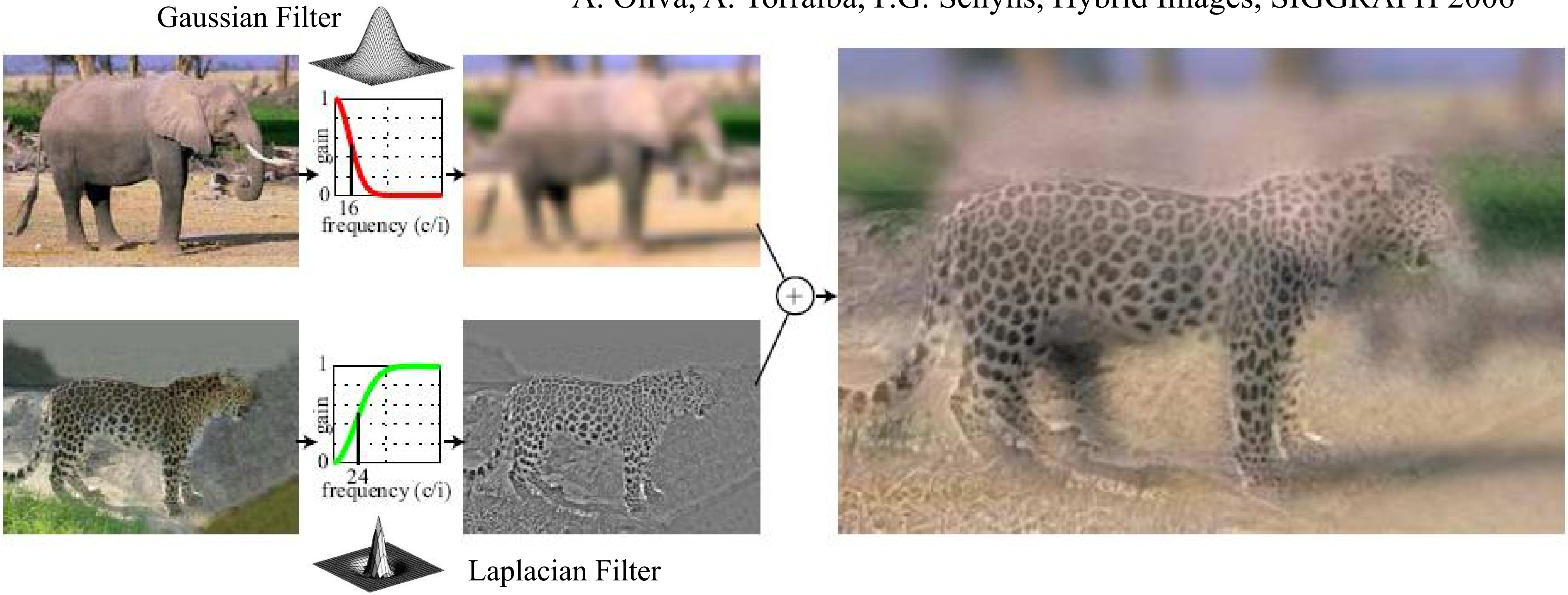
原图



立斯算子滤波结果

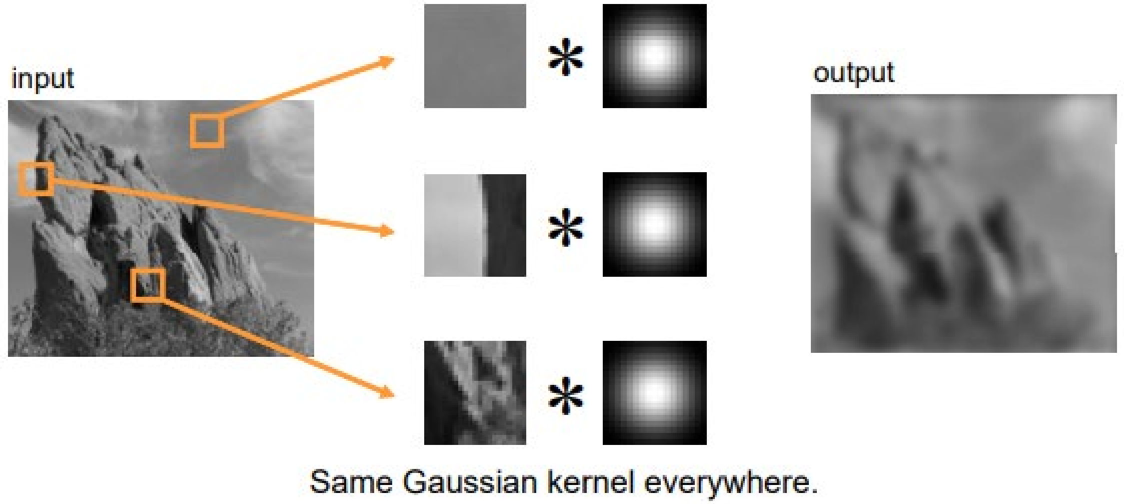
拉普拉斯算子应用——混合图像

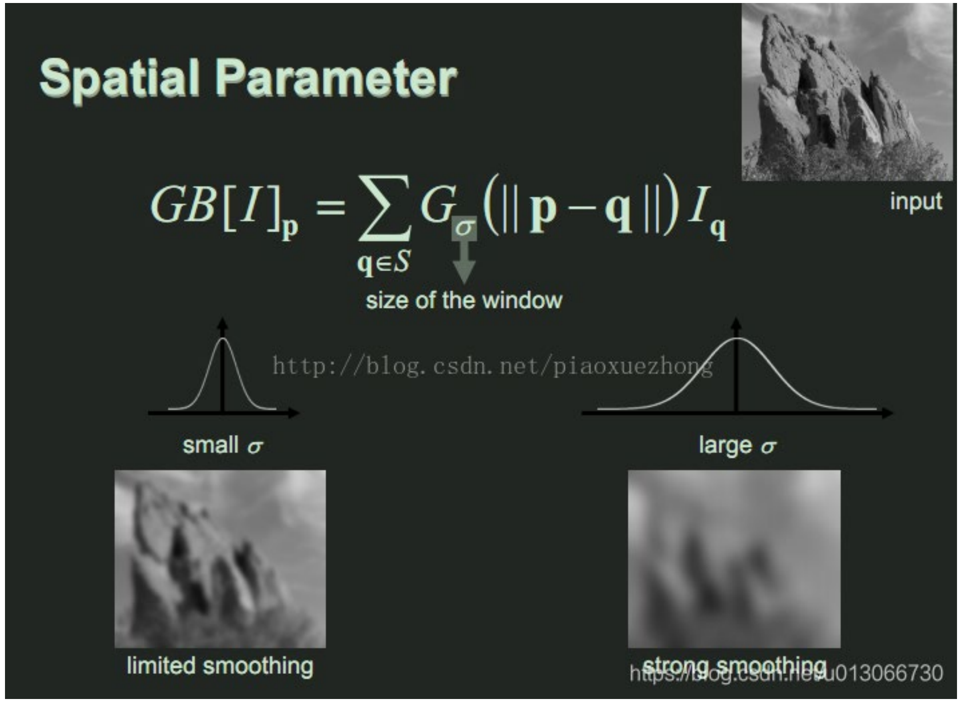
- A. Oliva, A. Torralba, P.G. Schyns, Hybrid Images, SIGGRAPH 2006



高斯滤波的缺陷

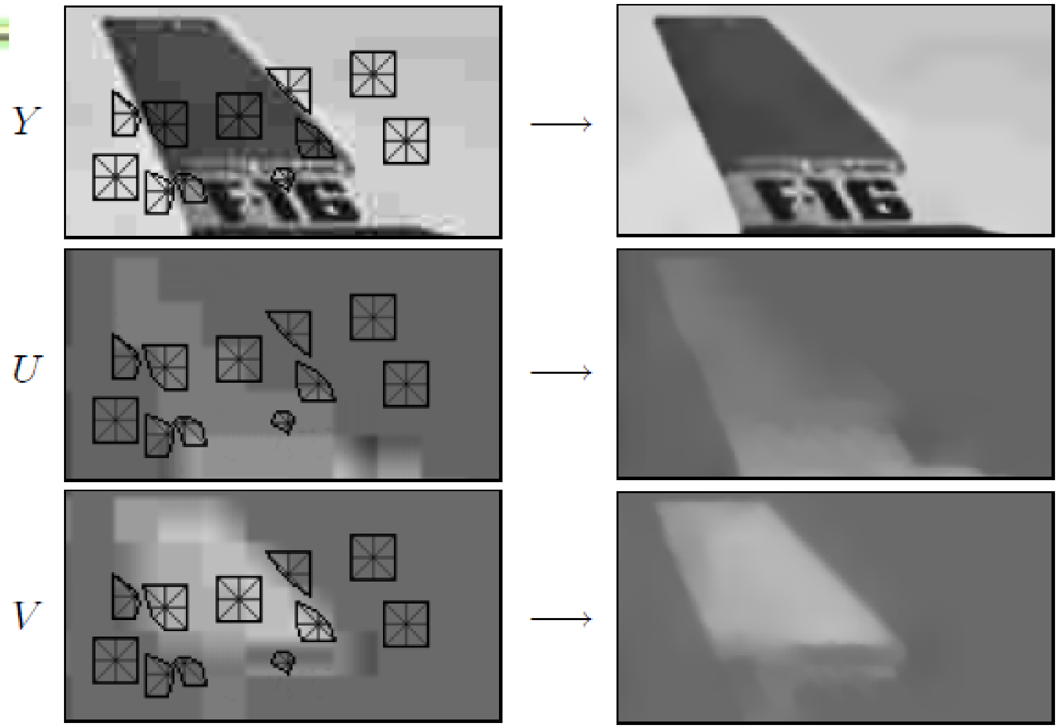
- 高斯滤波是以距离为权重，设计滤波模板作为滤波系数，只考虑了像素间的空间位置上的关系。
- 平坦区域正常滤波，图像细节没有变化，而在突变的边缘上，因为只使用了距离来确定滤波权重，导致边缘被模糊。

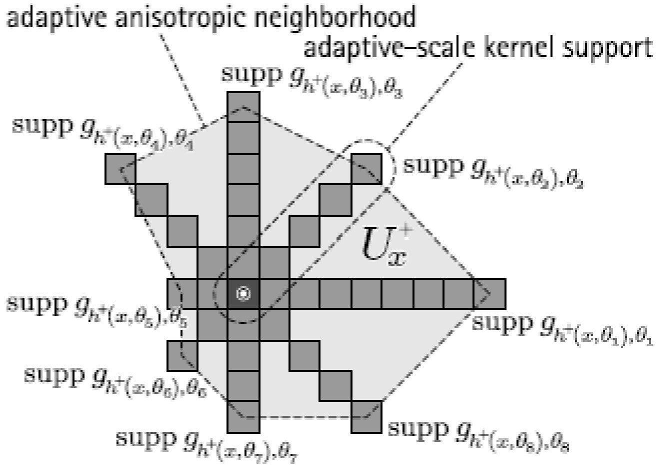


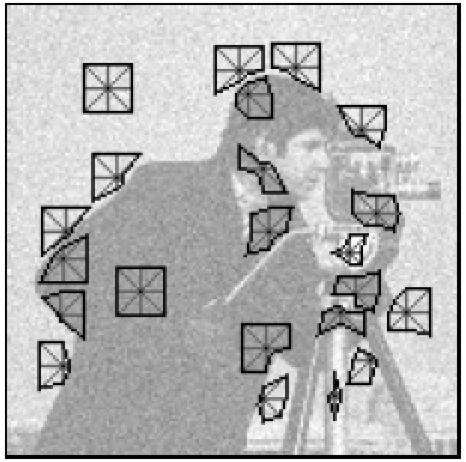


自适应窗口滤波

- A. Foi, V. Katkovnik and K. Egiazarian, Pointwise Shape-Adaptive DCT for High-Quality Denoising and Deblocking of Grayscale and Color Images. IEEE Transactions on Image Processing, 2007. 16(5): p. 1395-1411.



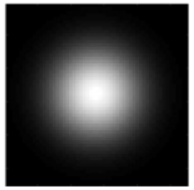


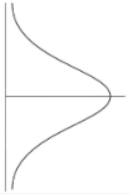


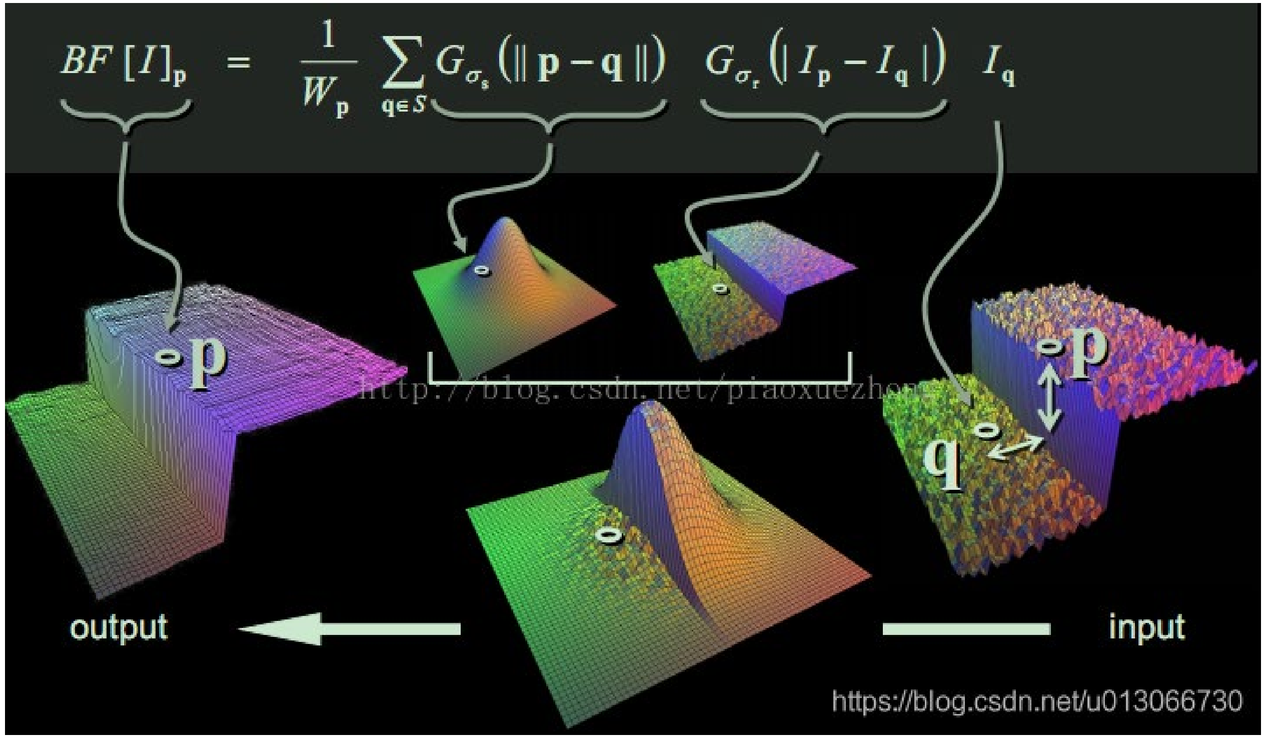
具有边缘保持特性的滤波—双边滤波

- **Bilateral Filters 双边滤波：** Qingxiong Yang; Kar-Han Tan; Narendra Ahuja. **Real-time O(1) bilateral filtering. CVPR2009**

$$BF[I]_p = \underbrace{\frac{1}{W_p}}_{\text{Normalization Factor}} \sum_{q \in S} \underbrace{G_{\sigma_s}(\|p-q\|)}_{\text{Space Weight}} \underbrace{G_{\sigma_r}(|I_p - I_q|)}_{\text{Range Weight}} I_q$$





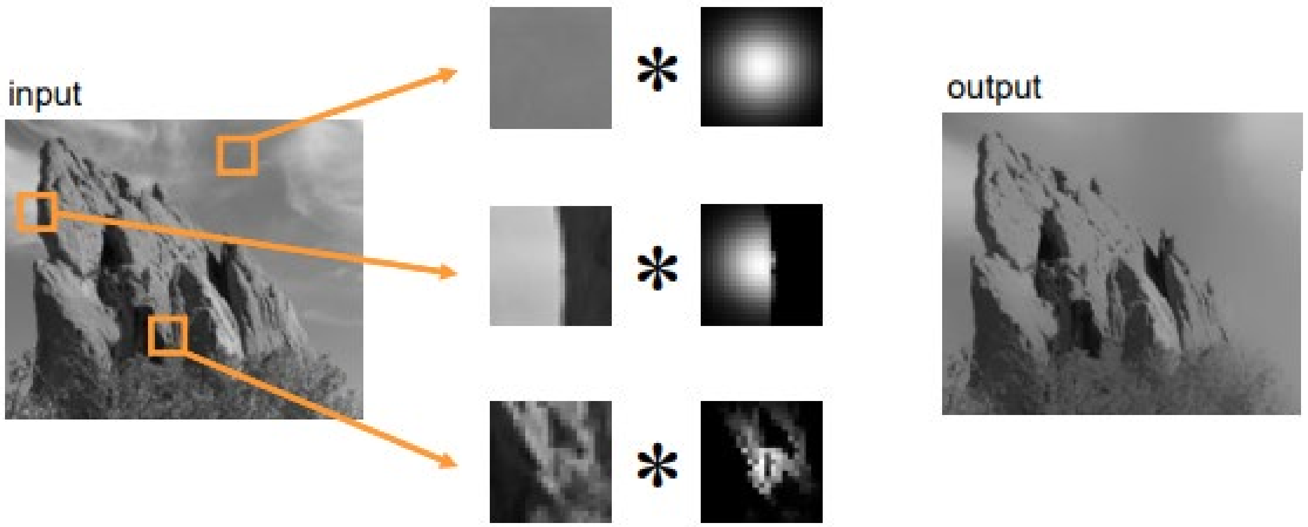


空间距离：当前点距离滤波模板中心点的欧式距离。

灰度距离：当前点距离滤波模板中心点的灰度差值的绝对值

具有边缘保持特性的滤波—双边滤波

- 双边滤波的核函数是空间域核与像素范围域核的综合结果：
- 1）在图像平坦区域，像素值变化小，则像素差值接近0，对应的像素范围域权重接近1，此时空间域权重起主要作用，相当于进行高斯模糊；
- 2）在图像边缘区域，像素值变化大，则像素差值大，对应的像素范围域权重变大，保护边缘信息。



The kernel shape depends on the image content.

Joint Bilateral Upsampling

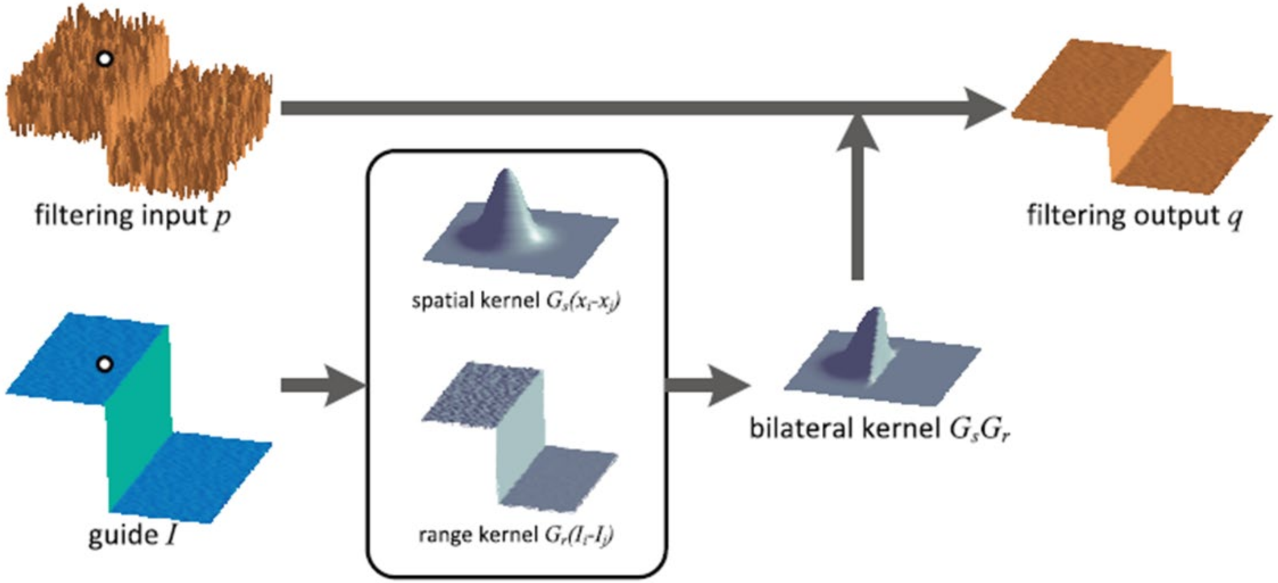
- Johannes Kopf, Michael F. Cohen, Dani Lischinski, Matt Uyttendaele. Joint Bilateral Upsampling. ACM Transactions on Graphics. 26(3): 96, 2007.

为了改善BF权值的稳定性，引入了联合双边滤波器（Joint Bilateral Filter, JBF)。**两者之间的差别**就是JBF用了一个导向图作为值域权重的计算依据。

BF：$$J_p = \frac{1}{k_p} \sum_{q \in \Omega} I_q \, f(\|p-q\|) \, g(\|I_p - I_q\|)$$

在值域的权重计算过程引入一幅导向图像$\tilde{I}$，则称为联合双边滤波，导向图像**必须**与待处理的图像相似。

JBF：$$J_p = \frac{1}{k_p} \sum_{q \in \Omega} I_q \, f(\|p-q\|) \, g(\|\tilde{I}_p - \tilde{I}_q\|)$$



导向滤波

• K. He, J. Sun and X. Tang, Guided Image Filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013. 35(6): p. 1397-1409.

导向滤波有两个输入：原始图 $\boldsymbol{p}$ 和引导图 $\boldsymbol{I}$，导向滤波公式： $q_i = \sum_j W_{ij}(I)p_j,$

假设：引导图 $\boldsymbol{I}$ 与输出图 $\boldsymbol{q}$ 之间满足线性关系：

$$q_i = a_k I_i + b_k, \forall i \in \omega_k$$

线性关系两边求导得到 $\nabla q_i = a \nabla I_i$

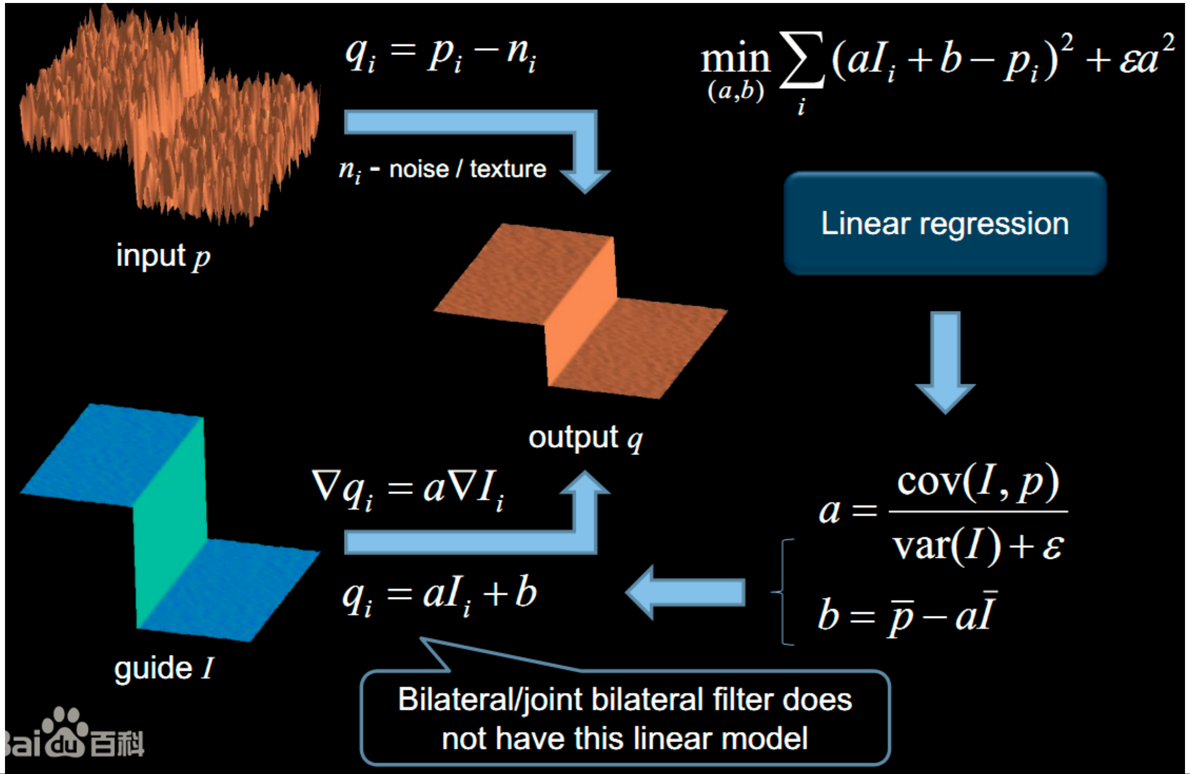
这表明引导图的梯度变化与输出图的梯度变化呈线性关系

已知引导图 $\boldsymbol{I}$，求系数a、b的值：

$$\underbrace{\min_{(a,b)} \sum_i (\underbrace{aI_i + b}_{\text{输出}q_i} - p_i)^2}_{\text{噪声的平方和}} + \underbrace{\varepsilon a^2}_{\text{正则化项}}$$

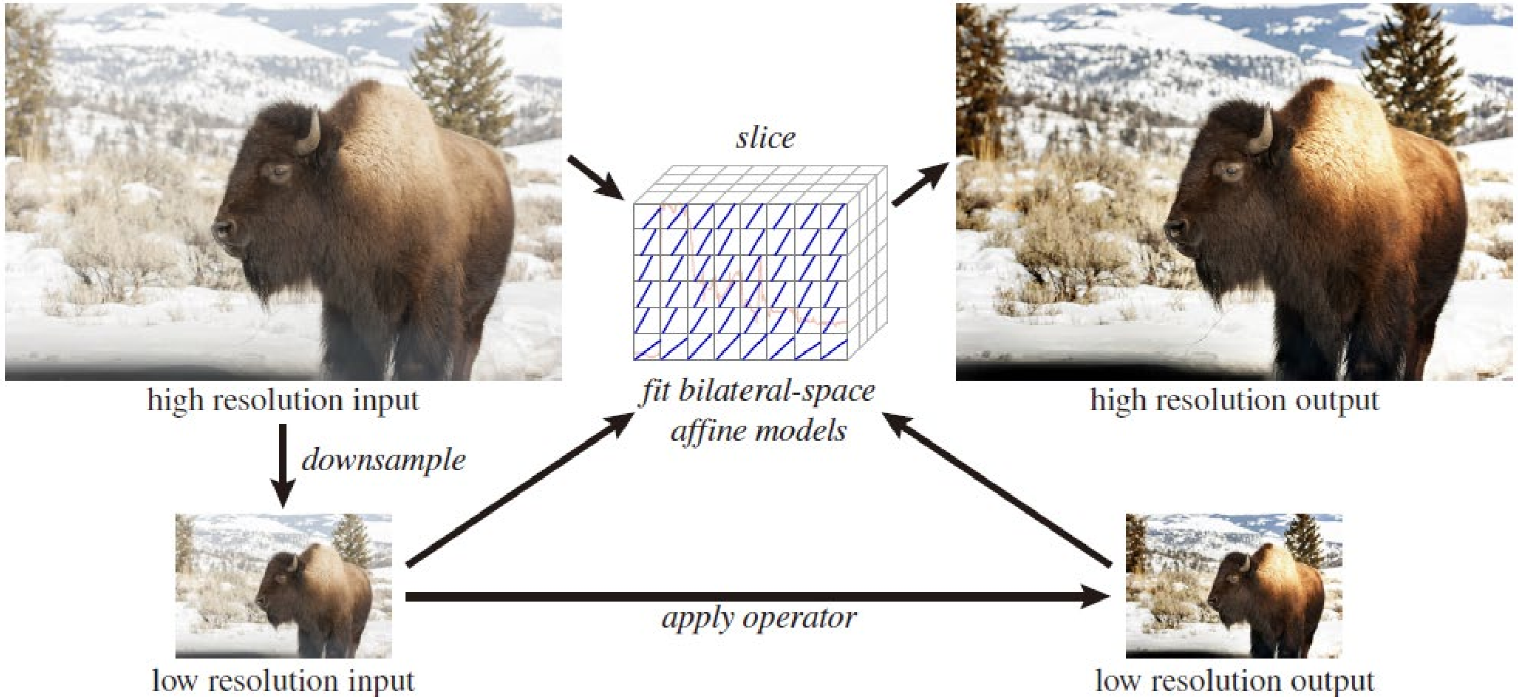
由最小二乘法可得参数a,b的表达式

$$\begin{cases} a = \dfrac{\operatorname{cov}(I,p)}{\operatorname{var}(I)+\varepsilon} \\ b = \bar{p} - a\bar{I} \end{cases}$$



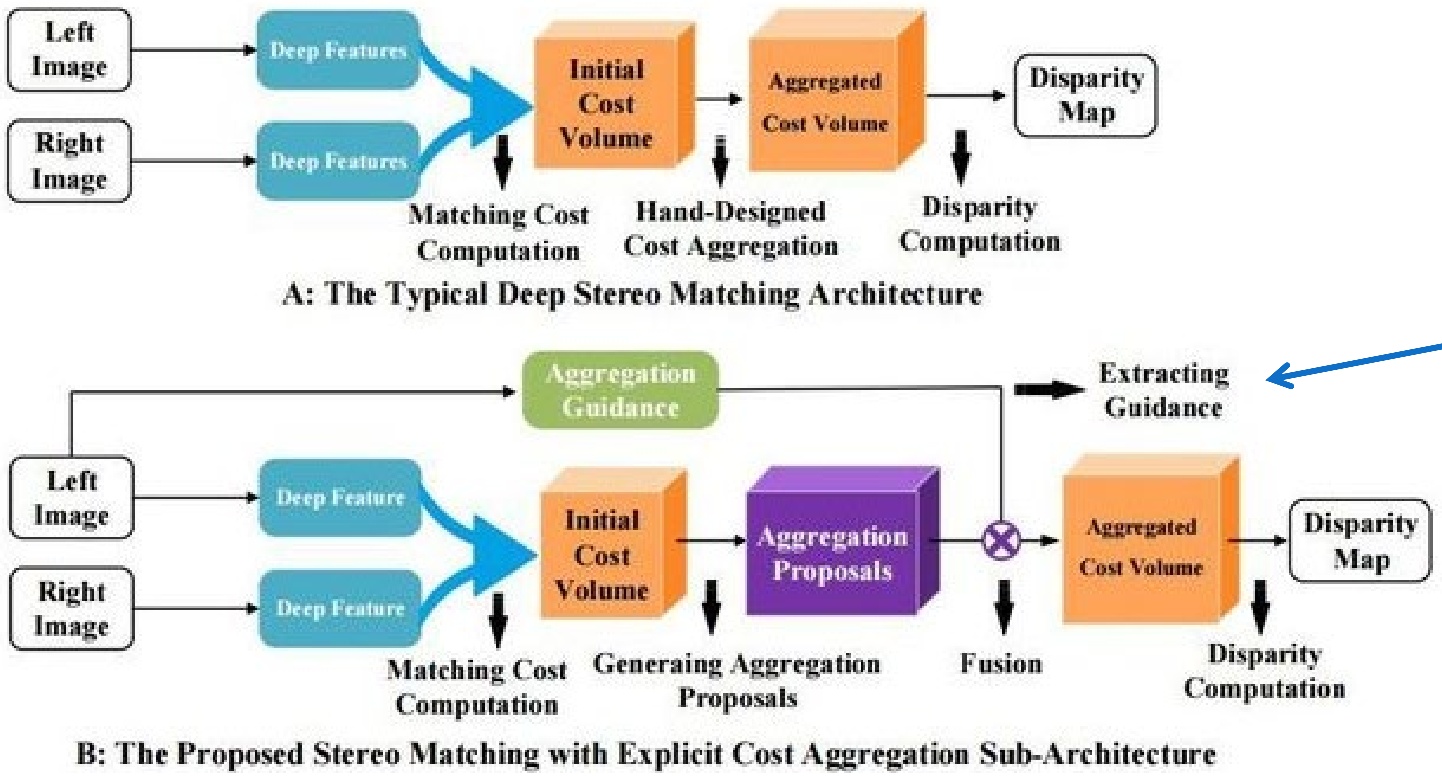
Bilateral Guided Upsampling

- J. Chen, A. Adams, N. Wadhwa, and S.W. Hasinoff. Bilateral Guided Upsampling. in ACM Transactions on Graphics (SIGGRAPH Asia 2016). 2016.



边缘保持/增强的深度学习

L. Yu, Y. Wang, Y. Wu, and Y. Jia. Deep Stereo Matching with Explicit Cost Aggregation Sub-Architecture. in Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18). 2018.



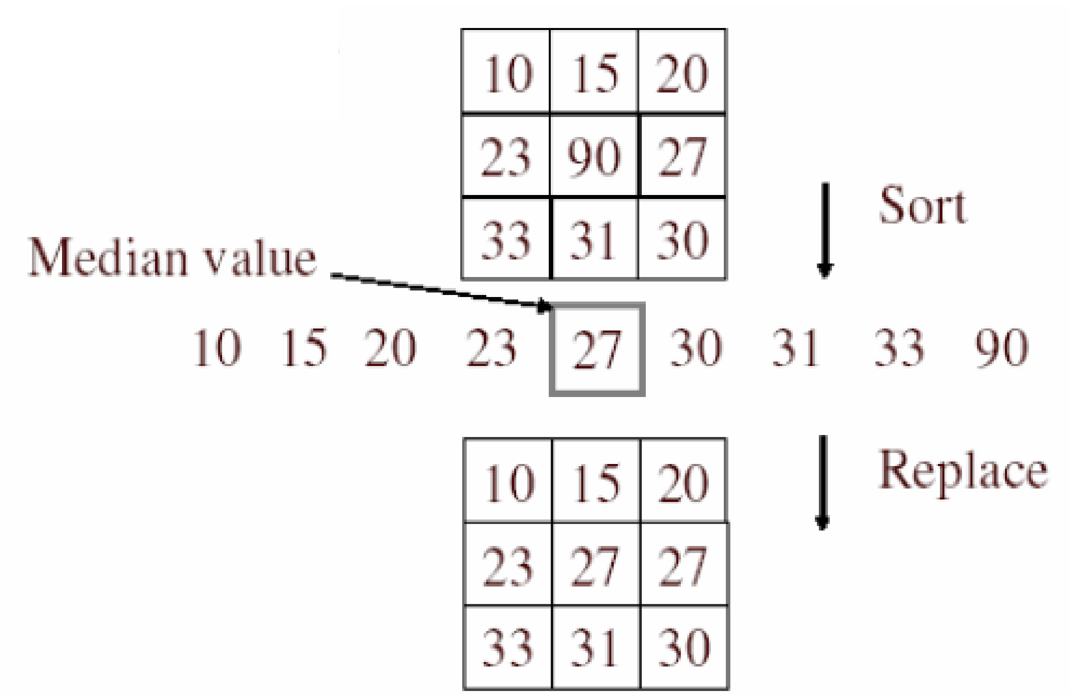
思想：采用轻卷积网络提取低阶结构信息，并作为导向信息，为立体匹配带来全局视角引导。

3.4 空域滤波

非线性滤波

中值滤波

- 选择局部窗口中亮度的中间值代替窗口中心像素

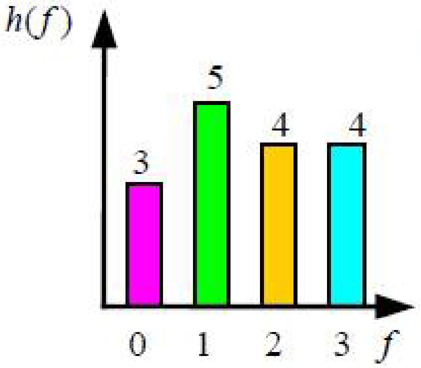


中值滤波步骤

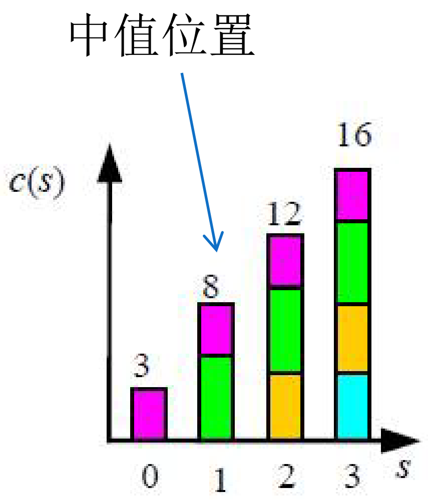
- 步骤：
  - 将模板中心与图像中某像素位置重合
  - 读取模板下各对应像素的灰度值
  - 将这些灰度值从小到大排成一列
  - 找出这些灰度值里排在中间的一个
  - 将这个中间值赋给对应模板中心位置像素
  - 遍历图像中所有像素

基于直方图的快速中值滤波

0	1	2	3
1	2	3	1
2	3	1	0
3	1	0	2



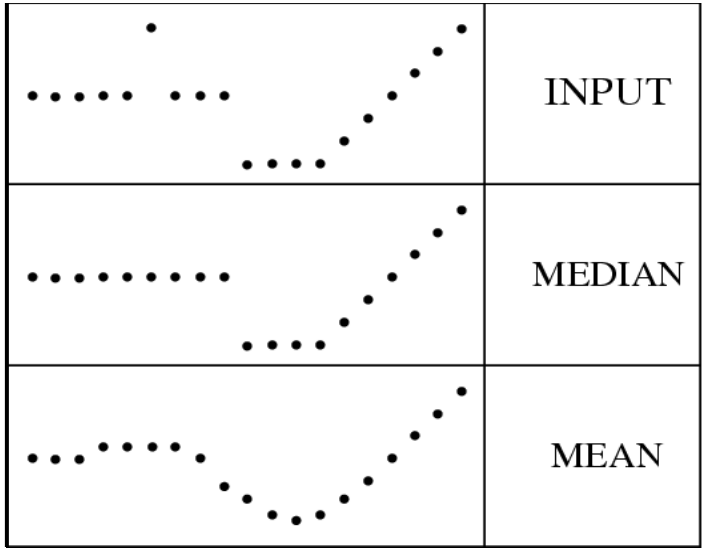
直方图



累积直方图

中值滤波 vs. 线性滤波

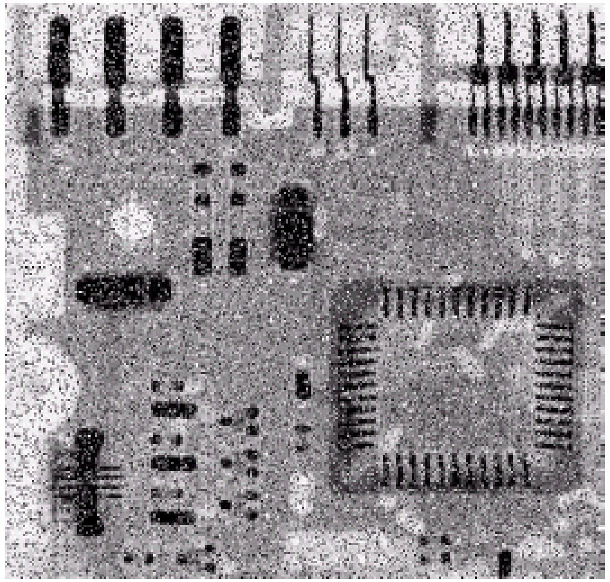
- 宽度为5的滤波模板



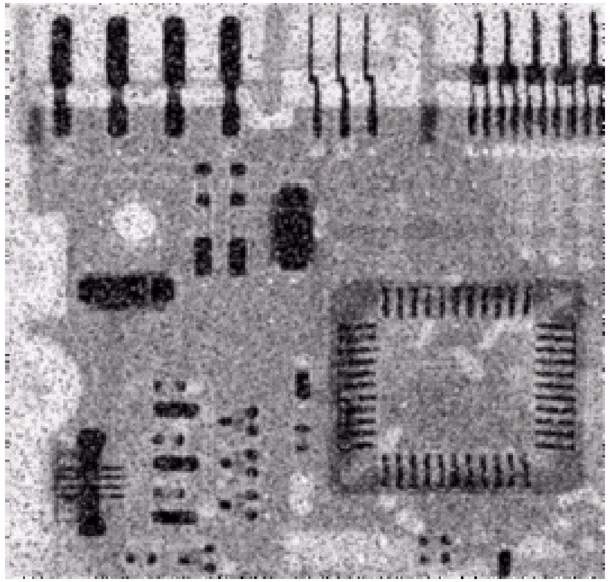
- 结论:
  - 1、中值滤波可有效消除突变
    线性滤波总是响应所有的变化
  - 2、中值滤波具有部分不连续保持特性
    线性滤波会产生平滑过渡的效果

中值滤波 vs. 均值滤波

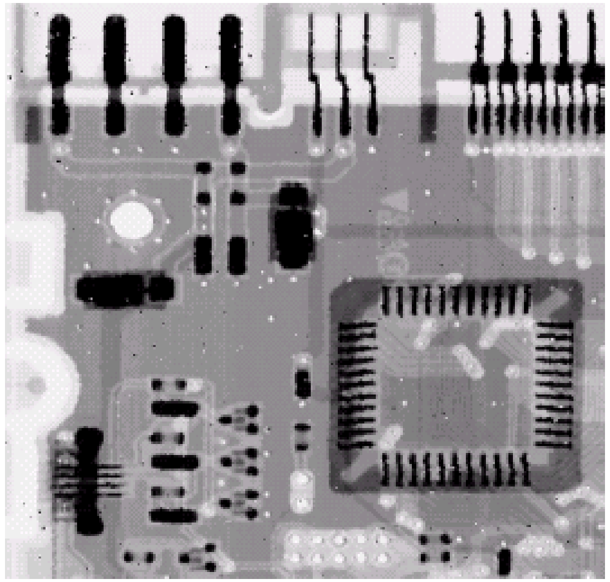
- 中值滤波器比均值滤波器更适合去除加性椒盐噪声



含噪图像



均值滤波结果



中值滤波结果

中值滤波 vs. 高斯滤波

3x3　5x5　7x7

Gaussian

Median

非线性锐化滤波

- **梯度**（基于一阶微分）

$$\nabla f = \left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y}\right]^{\mathrm{T}} = \left[G_X \quad G_Y\right]^{\mathrm{T}}$$

−1		1
−1		1
−1		1

$\frac{\partial f}{\partial x}$

1	1	1
−1	−1	−1

$\frac{\partial f}{\partial y}$

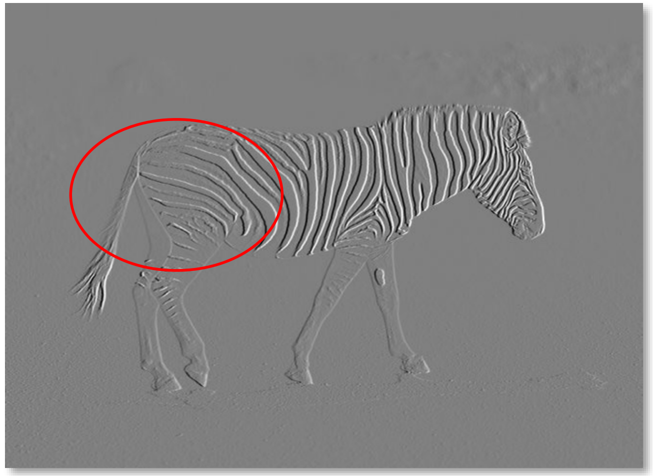
$$\left|\nabla f_{(2)}\right| = \mathrm{mag}(\nabla f) = \left[G_X^2 + G_Y^2\right]^{1/2}$$

$$\left|\nabla f_{(1)}\right| = |G_X| + |G_Y|$$

$$\left|\nabla f_{(\infty)}\right| = \max\left\{|G_X|, |G_Y|\right\}$$

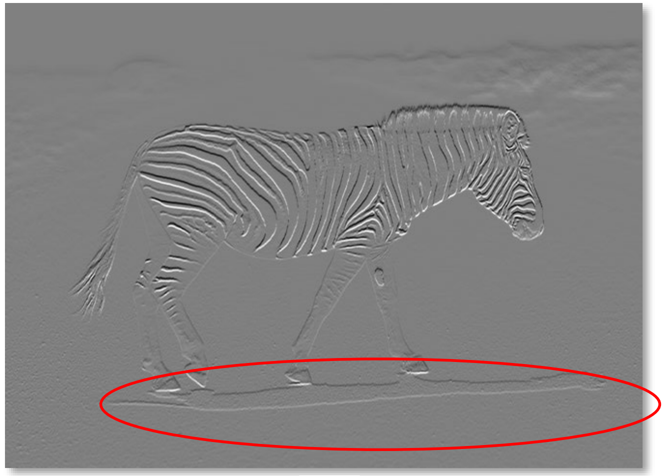
非线性锐化滤波

f



$\frac{\partial f}{\partial x}$

$$\|\nabla f\| = \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2}$$



$\frac{\partial f}{\partial y}$

习题

第四章 图像预处理

习题4.1 P75

- 设给定平移量(2, 5)，并用2和5作为放缩因子沿X和Y轴进行尺度变换，分别计算对图像点(2,5)先平移变换后尺度变换和先尺度变换后平移变换所得的结果。

- 平移矩阵

（假定左平移）

$$\mathbf{T}=\begin{bmatrix}1 & 0 & u_0\\ 0 & 1 & v_0\\ 0 & 0 & 1\end{bmatrix}=\begin{bmatrix}1 & 0 & -2\\ 0 & 1 & -5\\ 0 & 0 & 1\end{bmatrix}$$

尺度变化

$$\mathbf{S}=\begin{bmatrix}s_x & 0 & 0\\ 0 & s_y & 0\\ 0 & 0 & 1\end{bmatrix}=\begin{bmatrix}2 & 0 & 0\\ 0 & 5 & 0\\ 0 & 0 & 1\end{bmatrix}$$

- 图像点

$$\begin{bmatrix}u\\ v\\ 1\end{bmatrix}=\begin{bmatrix}2\\ 5\\ 1\end{bmatrix}$$

习题4.1 P75

- 先平移变换后尺度变换

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \mathbf{ST}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- 先尺度变换后平移变换

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \mathbf{TS}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -5 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 4 \\ 25 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 20 \\ 1 \end{bmatrix}$$

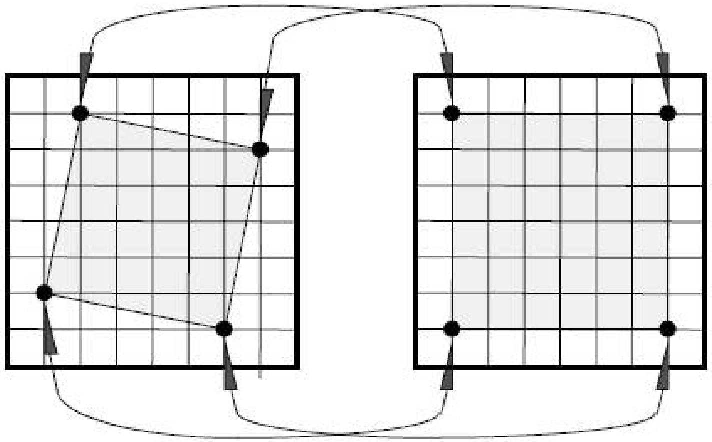
习题4.2 P75

- 给出实现对一个像素先平移、再旋转、最后尺度变换的变换矩阵。

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \mathbf{S} \cdot \mathbf{R} \cdot \mathbf{T} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$$

习题4.3 P75

- 设用三角形代替下图中的四边形，建立与下式相对应的校正几何形变的空间变换式。



$$x' = k_1 x + k_2 y + k_3 xy + k_4$$

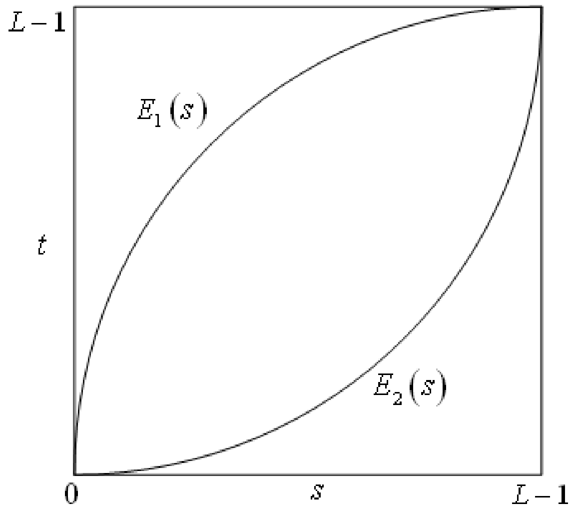
$$y' = k_5 x + k_6 y + k_7 xy + k_8$$

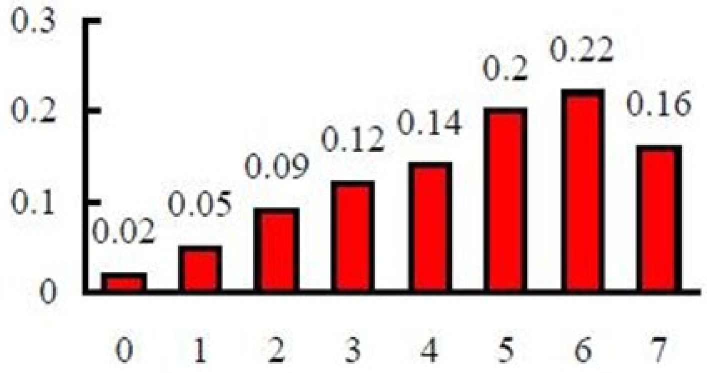
- 解: 以顶点为对应点，一个对应点可列出2个公式，因此三角形的三个顶点可列出6个公式，最多求解6个几何形变参数，因此空间变换式为：

$$\begin{cases} x' = k_1 x + k_2 y + k_3 \\ y' = k_4 x + k_5 y + k_6 \end{cases}$$

习题4.5 P75

- $E_1(s)$和$E_2(s)$为两条灰度变换曲线
- (1) 讨论这两条曲线的特点、功能
  及适用场合
- (2) 设L=8， $E_1(s)=\text{int}[(7s)^{1/2}+0.5]$，
  $E_2(s)=\text{int}[s^2/7+0.5]$，分别对下
  面直方图所对应的图像进行灰度变换，给出变
  换后图像的直方图（可画图或列表）





习题4.5 P75

- (1) 讨论这两条曲线的特点、功能及适用场合

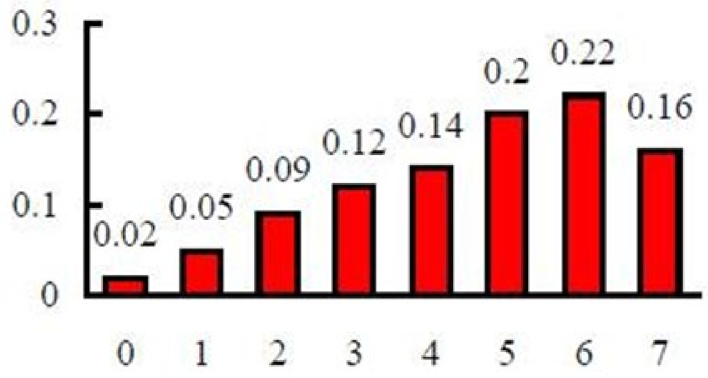
- $E_1(s)$可较大地提高原图中灰度较小像素的灰度值，并拉升灰度较小像素间的对比度。但会压缩灰度较大像素间的对比度。提升整幅图像的亮度。
- 适用于原图像（背景）偏暗的场合

- $E_2(s)$可较大地提高原图中灰度较大像素的灰度值，并拉升灰度较大像素间的对比度。但会压缩灰度较小像素间的对比度。降低整幅图像的亮度。
- 适用于原图像（背景）偏亮的场合

习题4.5 P75

- 问题(2): L=8， $E_1(s)$=int[$(7s)^{1/2}$+0.5]，

  $E_2(s)$=int[$s^2/7$+0.5]

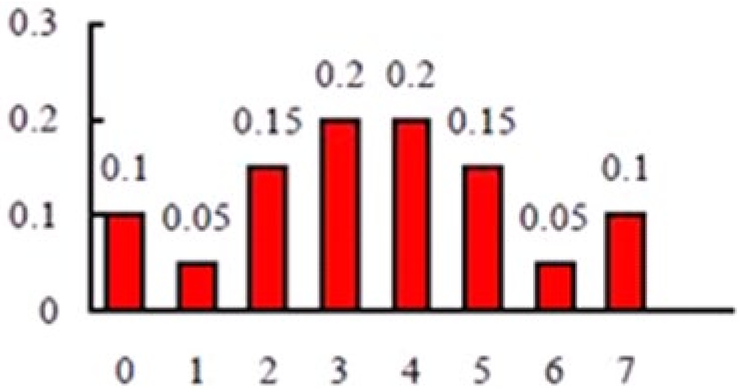


s	0	1	2	3	4	5	6	7
$(7s)^{1/2}$	0	2.6458	3.7417	4.5826	5.2915	5.9161	6.4807	7
$E_1(s)$	0	3	4	5	5	6	6	7
$p(n_k)$	0.02	0	0	0.05	0.09	0.26	0.42	0.16
$s^2/7$	0	0.1429	0.5714	1.2857	2.2857	3.5714	5.1429	7
$E_2(s)$	0	0	1	1	2	4	5	7
$p(n_k)$	0.07	0.21	0.14	0	0.2	0.22	0	0.16

习题4.8 P75

- 设原始图像直方图如下图所示，

  列表进行直方图均衡化计算



灰度级	0	1	2	3	4	5	6	7
原始直方图	0.1	0.05	0.15	0.2	0.2	0.15	0.05	0.1
累积直方图g_f	0.1	0.15	0.3	0.5	0.7	0.85	0.9	1.0
int($(L$-1)*g_f+0.5) 归一化(0, L-1)间	1	1	2	4	5	6	6	7
确定映射关系	0,1->1		2->2	3->4	4->5	5,6->6		7->7
新直方图	0	0.15	0.15	0	0.2	0.2	0.2	0.1

习题4.11 P76

- 将M幅图像相加求平均可以获得消除噪声的效果，用一个n×n的模板进行平滑滤波也可获得消除噪声的效果, 试比较两种方法的消噪效果

- M幅图像相加求平均 → 时间轴上的平均 容易产生运动模糊(重影)

- n×n模板的平滑滤波 → 空间上的平均 容易产生空间模糊(边缘模糊)

习题4.12 P76

- 讨论用于空间滤波的平滑滤波和锐化滤波的相同点、不同点以及联系

- 相同点：都能减弱或消除频域空间中的某些分量，而不影响或较少影响其它分量，从而达到增强效果。
- 不同点：平滑滤波减弱或消除高频分量，增强低频分量，平滑图像中的细节信息。锐化滤波减弱或消除低频分量，增强高频分量，锐化图像中细节信息。
- 联系：两者效果相反，互为补充；从原始图像中减去平滑滤波结果可得到锐化滤波效果；而原始图像中减去锐化滤波结果可得到平滑滤波效果

The End