

去除彩色噪声 (CNR)

如果单独对每个通道进行去噪，没有利用上通道之间的相关性。

1. Kodak patent

整体流程如下，先转为 LAB 空间，在这三个空间上分别进行边缘检测，融合这些边缘，利用这些边缘的类型考虑不同的邻居，之后在 AB 空间上进行模糊。

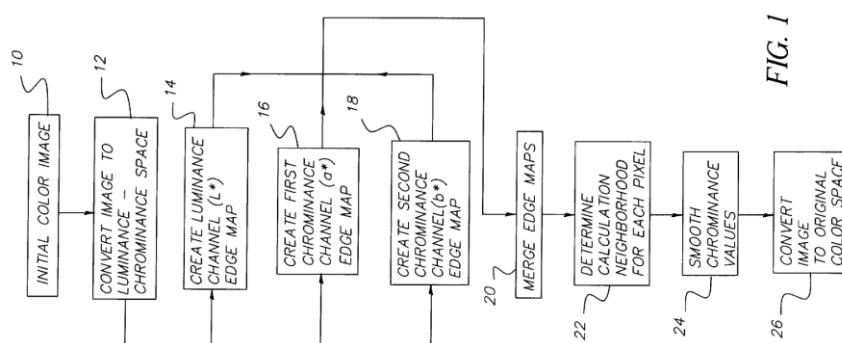


FIG. 1

1. 在 LAB 空间上分别进行边缘检测：这篇就是利用一个 5*5 的滤波进行检测，检测四种，横竖斜，如下：

$$\begin{aligned}
 h &= \frac{1}{13} \begin{pmatrix} 1 & 2 & 0 & -2 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 2 & 0 & -2 & -1 \end{pmatrix} & v &= \frac{1}{13} \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ 1 & -2 & -2 & -2 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \\
 s &= \frac{1}{13} \begin{pmatrix} -1 & -1 & -1 & -1 & 0 \\ 1 & -2 & -2 & 0 & 1 \\ -1 & -2 & 0 & 2 & 1 \\ -1 & 0 & 2 & 2 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix} & b &= \frac{1}{13} \begin{pmatrix} 0 & -1 & -1 & -1 & -1 \\ 1 & 0 & -2 & -2 & 1 \\ 1 & 2 & 0 & -2 & -1 \\ 1 & 2 & 2 & 0 & -1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}
 \end{aligned}$$

2. 融合边缘：将上面三个空间所得边缘融合，注意它并没有完全进行边缘提取（就是类似 canny 的结果，0 是边缘，1 不是边缘），而是每个点处理后都有一个值 g。如果一个点和周围点的值相似，那么他们是属于同一片区域。

$$g = g(L^*) + g(a^*) + g(b^*)$$

3. 根据边缘考虑不同的邻居：从中心点开始，向四周八个方向进行扩散，如果检测到当前这个点和中心点的值 g 大于阈值，那么就停止对应方向的搜索。下图是一个例子：

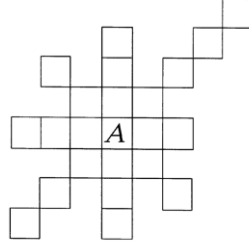


FIG. 2

2. Apple Patent

很简单，没想到这都可以提出专利。就是转成 YCrCb，然后调整 Cr 和 Cb。调整方式就是通过加权平均，权重的计算也很简单：就是判断该点和窗口中心的 Cr 和 Cb 是否接近，如果大于阈值就认为是边缘，那么就不参与平均，权重为 0；如果小于阈值就参与计算，权重为 1。

$$C'_{b_0} = \frac{\sum_{n=-N_1}^{N_2} (C_{b_n} T_n)}{\sum_{n=-N_1}^{N_2} (T_n)}, \quad C'_{r_0} = \frac{\sum_{n=-N_1}^{N_2} (C_{r_n} T_n)}{\sum_{n=-N_1}^{N_2} (T_n)},$$

wherein: If $|C_{b_n} - C_{b_0}| + |C_{r_n} - C_{r_0}| > \Gamma$,
Then $T_n = 0$, Else $T_n = 1$.

wherein: If $|C_{b_n} - C_{b_0}| + |C_{r_n} - C_{r_0}| > \Gamma$,
Then $T_n = 0$, Else $T_n = 1$.

专利的后面还写了一大堆，主要就是讲的行处理和列处理可以分开进行，先进行方向的上述去噪，再将列方向也使用上面的公式计算（列方向用的是行方向处理后的结果）。

3. STM Patent

比较花里胡哨，会有许多的公式。自顶向下来讲，还是在 Cr 和 Cb 空间上进行调整：

$$Cr_k = originalCr_k + f(Cf) \cdot \left(denoisedCr_L - originalCr_k \right)$$

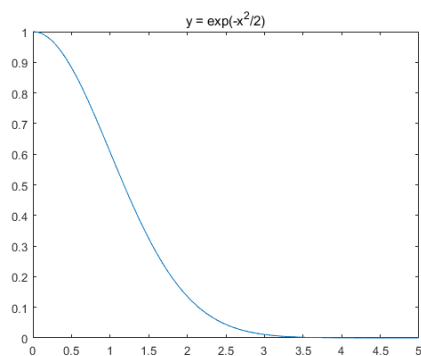
$$Cb_k = originalCb_k + f(Cf) \cdot \left(denoisedCb_L - originalCb_k \right)$$

公式上面有两个符号不晓得：f(Cf)和 denoisedCr，自顶向下嘛，慢慢来。

1. 首先 f(Cr)中 f 这个函数计算为：

$$f(x) = e^{-\frac{1}{2} \left(\frac{x}{sigma} \right)^2} \quad x \in [0, \text{max value}]$$

它的形式如下，也就是说 x 越大对应的值越小：



而其中的自变量 Cf 则是根据如下方式计算而来的：

$$Cf = \begin{cases} DY & \text{if } DY = \min(DY, DCr, DCb) \\ \max(DY, DCr, DCb) & \text{otherwise} \end{cases}$$

$$DCr = \max_l(Cr) - \min_l(Cr)$$

$$DCb = \max_l(Cb) - \min_l(Cb)$$

$$DY = \max_l(Y) - \min_l(Y)$$

2. denoisedCr 是根据如下方式计算的，根据周围像素和当前像素的色差距离计算不同的权重，然后加权平均所得：

$denoisedCr = \sum_{k=1}^{M \times M} \frac{WCr_k}{sumweightCr} \cdot Cr[k]$ $denoisedCb = \sum_{k=1}^{M \times M} \frac{WCb_k}{sumweightCb} \cdot Cb[k]$	$WCr_k = e^{-\frac{1}{2} \left(\frac{ Y(k) - Y(c) }{sigmaY} \right)^2} \cdot e^{-\frac{1}{2} \left(\frac{ Cr(k) - Cr(c) }{sigmaCr} \right)^2}$ $sumweightCr = \sum_{k=1}^{M \times M} WCr_k$ $WCb_k = e^{-\frac{1}{2} \left(\frac{ Y(k) - Y(c) }{sigmaY} \right)^2} \cdot e^{-\frac{1}{2} \left(\frac{ Cb(k) - Cb(c) }{sigmaCb} \right)^2}$ $sumweightCb = \sum_{k=1}^{M \times M} WCb_k$
---	---

3. 最后回顾公式：

$$Cr_k = originalCr_k + f(Cf) \cdot \left(denoisedCr - originalCr_k \right)$$