

Binarization Error Process

设 X 为二值化处理后得到的二值图像，仅由0和1组成，其中(0,1)分别表示文本像素和背景像素。为了估计文本笔画的最小长度，我们重新定义了 X 。

定义序列 $vec_R(X) = [x_1 \ x_2 \ \dots \ x_N], x_i \in \{0,1\}$ ，该序列由 X 的所有像素逐行拼接而成。我们在 $vec_R(X)$ 中依次计算由相邻0像素组成的每个片段的像素数，并且让它们从左到右生成一个长度不递减的序列(用 $N_R(X)$ 表示)。示例如下：

$$X = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad vec_R(X) = [1001010010100001], N_R(X) = [1,1,2,2,4] \quad (1)$$

定义 X 中的最小文本长度(MLT)公式如下：

$$MLT(X) = mean(N_R(X)(1:p)), p = \lfloor r^\wedge \cdot length(N_R(X)) \rfloor \quad (2)$$

其中 $r^\wedge \in (0,1)$ 为比例系数， $\lfloor x \rfloor$ 表示高斯函数。

对于 $vec_R(X)$ 中,由相邻0像素组成的每一个片段 $[\dots 1 \ x_k \ x_{k+1} \ \dots \ x_{k+m-1} \ 1 \ \dots]$

对于 $vec_R(X)$ 中由相邻0像素组成的每一段，其中 $x_i = 0$ ($k \leq i \leq k + m - 1$)，

$x_{k-1} = x_{k+m} = 1$,由以下规则修正：

$$x'_i = \begin{cases} 1, & \text{if } m \leq MLT(X) \\ x_i, & \text{if } m > MLT(X) \end{cases}, \quad \forall i \in (k, k + m - 1) \quad (3)$$

由 $vec_R(X) = [\dots 1 \ x_k \ x_{k+1} \ \dots \ x_{k+m-1} \ 1 \ \dots]$,根据上述规则，可以得到修正序
列： $vec'_R(X) = [\dots 1 \ x'_k \ x'_{k+1} \ \dots \ x'_{k+m-1} \ 1 \ \dots]$ ，然后我们把它转换成一个新的二
值图像 $X'_R = vec_R^{-1}(vec'_R(X))$ 。举个例子，对于(1)式当中的 X ，当 $r^\wedge = 0.2$ 和由
(2)式得到的 $MLT(X) = 1$ 时，我们可以得到：

$$vec_R(X) = [1001010010100001] \Rightarrow vec'_R(X) = [1001110011100001] \quad (4)$$

$$X = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow X'_R = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

为了估计 X 中文本笔画的最小宽度，我们将 X 转换为序列 $\text{vec}_c(X)$ ，该序列通过按列连接 X 的所有像素生成。根据上面类似的操作，我们得到另一个二进制图像 X'_c ：

$$X = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow X'_c = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad (6)$$

最终二值化后的图像矩阵 \hat{X} 如下定义：

$$\hat{X}(i,j) = \begin{cases} 1, & \text{if } X'_R(i,j) = 1 \text{ and } X'_C(i,j) = 1 \\ 0, & \text{otherwise} \end{cases}, \quad \forall (i,j) \in \Omega \quad (7)$$

由上规则，我们可得到最终的图像矩阵 \hat{X} ：

$$\hat{X} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

后处理方法流程图例子如下图 1 所示：

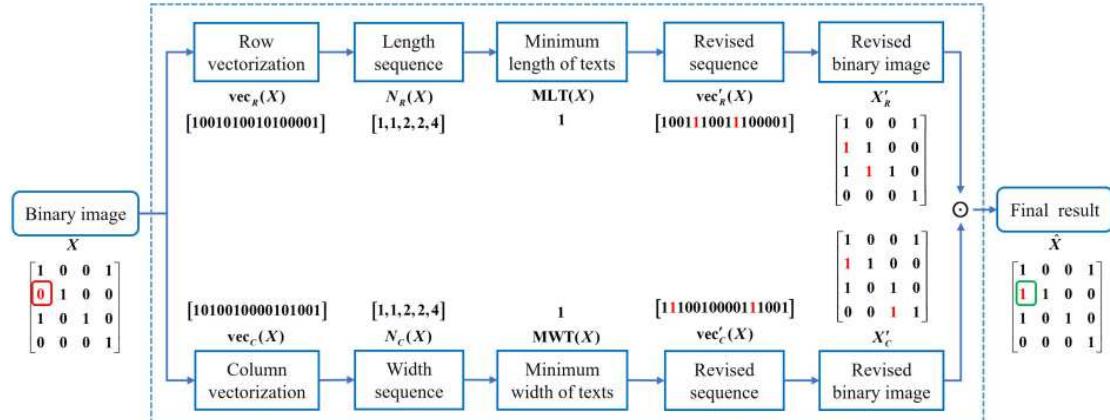


图 1 后处理方法流程图