

MB2028 V1 成像图像清晰度评价算法

一.实现目标

让 MB2028 V1 与扫码目标之间保持不同的水平距离，产生不同的成像图像。需设计算法对图像进行清晰度评价，并找到样本内最佳成像距离。

二.当前问题

1.获取图像样本时，同一条件下（光照、条码位置、摄像头角度、水平距离），会产生亮度明显不同的成像图像。由于目前大多数图像清晰度评价算法对亮度变化敏感，因此对于同一条件下的图像样本，将会产生不同的评价函数值，影响了算法关键参数的调试。

2.仅凭肉眼难以分辨出最佳水平距离。因此无法评价算法对于查找最佳成像距离功能实现的优劣程度。

三.实现步骤

1.获取训练图集

获取了在三组训练图像，分别是：

- 只在台灯照射下的采集图像，编号为训练图集 A（过曝）
 - 无台灯并开启照明灯和对焦灯常亮模式下的采集图像，编号为训练图集 B
 - 不开台灯、照明灯、对焦灯下的采样图像，编号为训练图集 C（欠曝）
- 每组的距离间隔都为 1cm，采样距离范围为 5cm~20cm。并只保存各图中条码图形。

2.设计算法

（1）基于梯度变化统计的算法

● 算法一：Sobel 算子统计法

Sobel 算子是一个离散的一阶微分算子，用来计算图像灰度函数的近似梯度。

在空间域上 Sobel 算子很容易实现，执行速度快，对部分噪声具有平滑作用，还能够提供较为精确的边缘方向信息，缺点是边缘定位精度不够高。边缘是指一个物体与另一个物体的分界处，一般边缘内外处都会有灰度值上的差异，Sobel 算子就是通过像素点空间邻域内上下，左右相邻点的灰度加权运算，求取物体边缘。

经典 Sobel 的卷积因子为：

| | | |
|----|---|----|
| -1 | 0 | +1 |
| -2 | 0 | +2 |
| -1 | 0 | +1 |

| | | |
|----|----|----|
| +1 | +2 | +1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

对于待检测边缘的图像 I，分别在水平（X）方向和垂直方向（Y）方向求导，计算方法是分别图像 I 与卷积核 G_x 和 G_y 进行卷积，公式表述如下：

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I$$

之后对求得的水平和垂直方向的梯度图像上的每一点执行计算：

$$G = |G_x| + |G_y|$$

G 即为所求的梯度图像。随后对 G 进行统计：

$$f_s(I) = \sum_x \sum_y G(x,y)^2 \quad (G(x,y) > T)$$

上式中, $\sum_y G(x,y)$ 是在点 (x,y) 上的图像梯度值, T 为梯度阈值, $f_s(I)$ 为大梯度统计量。

由于各样本的像素点个数不同, 为了评价在相同像素点个数下各样本的清晰度, 因此需要进行如下计算:

$$f(I) = \frac{f_s(I)}{M \times N}$$

其中, M 为图像的高度 (纵轴上像素点的个数), N 为图像的宽度 (横轴上像素点的个数)。 $f(I)$ 即为图像清晰度评价函数。 $f(I)$ 的值越大, 说明图像梯度变化大, 清晰度高。

● 算法二: Tenengrad 函数

Tenengrad 函数式一种常用的图像清晰度评价函数, 是一种基于梯度的函数。在图像处理中, 一般认为对焦好的图像具有更尖锐的边缘, 故具有更大的梯度函数值。

Tenengrad 函数使用 Sobel 算子提取水平和垂直方向的梯度值。具体过程如下:

设 Sobel 卷积核为 G_x 和 G_y , 则图像 I 在点 (x,y) 处的梯度定义为:

$$S(x,y) = \sqrt{[G_x * I(x,y)]^2 + [G_y * I(x,y)]^2}$$

定义该图像的 Tenengrad 值为:

$$f(I) = Ten(I) = \frac{1}{M \times N} \sum_x \sum_y S(x,y)^2$$

Tenengrad 函数中已经去除了不同像素点总数对评价函数的影响。若 $f(I)$ 的值越大, 则图像越清晰。

(2) 基于方差统计的算法

● 算法三: SMD2 (灰度方差乘积) 函数

我们首先考虑使用对灰度方差进行分析的常规 SMD(standard mean difference)函数。当完全聚焦时, 图像最清晰, 图像中的高频分量也最多, 故可将灰度变化作为聚焦评价的依据, 灰度方差法的公式如下:

$$Var(x,y) = |I(x,y) - I(x,y-1)| + |I(x,y) - I(x+1,y)|$$

$$f(I) = \frac{1}{M \times N} \sum_x \sum_y Var(x,y)^2$$

灰度差分评价函数具有较好的计算性能, 但其缺点也很明显, 即在焦点附近灵敏度不高, 即该函数在极值点附近过于平坦, 从而导致聚焦精度难以提高。在论文《一种快速高灵敏度聚焦评价函数》中提出了一种新的评价函数, 称之为灰度方差乘积法, 即对每一个像素领域两个灰度差相乘后再逐个像素累加, 该函数定义如下:

$$Var2(x,y) = |I(x,y) - I(x,y-1)| \times |I(x,y) - I(x+1,y)|$$

$$f(I) = \frac{1}{M \times N} \sum_x \sum_y Var2(x,y)^2$$

● 算法四: 基于方差的模糊程度函数

当图像块的方差值越低时, 图像的模糊程度越高。因此可以通过计算图像中各像素灰度值与周围像素灰度值的偏离程度来评价图像模糊程度。整个图像的总偏离程度由每一个像

素点与周围像素点的偏离程度总和获得。

在计算某像素点灰度值与周围像素点灰度值偏离程度时，以该像素点为中心，取 $P \times P$ 的一个窗口，其中 P 为一个正奇数，窗口的取法为以该像素点为中心，向上下左右各延伸 $(P-1)/2$ 个像素，即从图像中取出一个 $P \times P$ 大小的图像块，用该图像块内各个像素灰度值与平均灰度值的方差 D ，反映该图像块中心像素点灰度值与周围像素灰度值的偏离程度。计算式如下：

$$u_i = \frac{\sum_{y=1}^P \sum_{x=1}^P I_i(x, y)}{P \times P}$$

其中 u_i 是第 i 个图像块的灰度值平均值。 $I_i(x, y)$ 表示第 i 个图像块第 y 行的第 x 个像素点灰度值。

$$D_i = \sum_{y=1}^P \sum_{x=1}^P [u_i - I_i(x, y)]^2$$

其中 D_i 表示第 i 个图像块的方差。

对图像各像素点灰度值与周围像素灰度值的总体偏离程度归一化为模糊度。计算如下

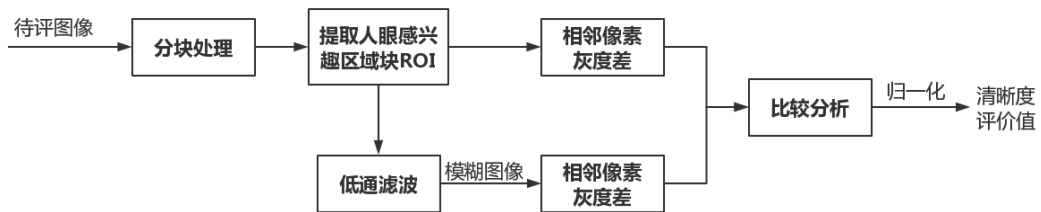
$$B(I) = \exp\left(-\frac{\sum_{i=1}^L D_i}{C \times L}\right)$$

其中 $B(I)$ 表示该图像的模糊程度评价函数。 L 为像素块总数。 C 是一个阈值，需取值为一个大于任意 D_i 的数。 $B(I)$ 越大，图像的模糊程度越高。

(3) 基于再次模糊的算法

● 算法五：Re-blur 函数

如果一幅图像已经比较模糊，那么再对它进行一次模糊处理，高频分量变化不大；若原图是比较清楚的，对它进行一次模糊处理，则高频分量变化会非常大。因此可以通过对待评测图像进行一次高斯模糊处理，得到该图像的退化图像，然后再比较原图像和退化图像相邻像素值的变化情况，根据变化的大小确定清晰度值的高低，计算结果越小则表明图像越清晰，反之越模糊。这种思路可称作基于二次模糊的清晰度算法，其算法简化流程图如下图：



步骤一：分块处理

与算法四相同，先将图像分块为 L 个图像块，并计算每个图像块的均值和方差。

$$u_i = \frac{\sum_{y=1}^P \sum_{x=1}^P I_i(x, y)}{P \times P}$$

$$D_i = \sum_{y=1}^P \sum_{x=1}^P [u_i - I_i(x, y)]^2$$

设定一个阈值 T ，令 $f_K = \{f_k | D_i > T, K < L\}$ 。其中 f_K 为所选取的图像块集合， K 为所选取图像块个数。

步骤二：低通滤波

将图像 f_k 经过垂直和水平滤波获得相应模糊图像块 b 。

$$b_V = h_V * f_k$$

$$b_H = h_h * f_k$$

其中： b_V 、 b_H 分别为图像块 f_k 经过垂直和水平滤波后获得的模糊图像块， h_V 、 h_h 分别为滤波器垂直和水平模型，可采用上面提到的 Sobel 卷积核。

步骤三：对单个像素点计算相邻像素灰度值变化

$$Df_V(i, j) = |f_k(i, j) - f_k(i - 1, j)|$$

$$Df_H(i, j) = |f_k(i, j) - f_k(i, j - 1)|$$

$$Db_V(i, j) = |b_V(i, j) - b_V(i - 1, j)|$$

$$Db_H(i, j) = |b_H(i, j) - b_H(i, j - 1)|$$

步骤四：对图像块进行计算：

$$sf_V = \sum_{j=1}^m \sum_{i=1}^n Df_V(i, j)$$

$$sf_H = \sum_{j=1}^m \sum_{i=1}^n Df_H(i, j)$$

$$sV_V = \sum_{j=1}^m \sum_{i=1}^n Db_V(i, j)$$

$$sV_H = \sum_{j=1}^m \sum_{i=1}^n Db_H(i, j)$$

步骤五：对图像差异值进行归一化

$$bf_V = \frac{sf_V - sV_V}{sf_V}$$

$$bf_H = \frac{sf_H - sV_H}{sf_H}$$

步骤六：计算最终评价值

对每一个图像块都按照步骤二到步骤五进行处理，然后对计算所得的所有 bf_V 、 bf_H 分别取平均值 $\overline{bf_V}$ 、 $\overline{bf_H}$ 。则最终评价值为：

$$\text{Blur}(I) = \sqrt{\overline{bf_V}^2 + \overline{bf_H}^2}$$

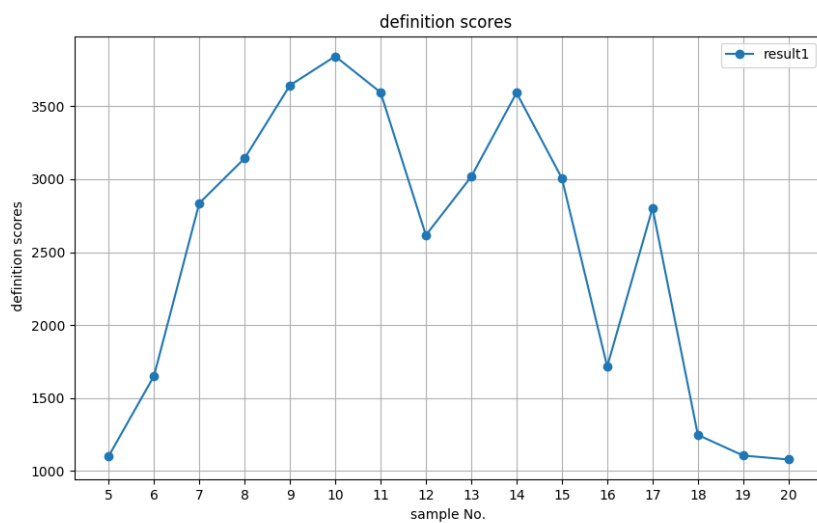
当 Blur 值越小时，则表明图像越清晰。

3. 算法调试结果

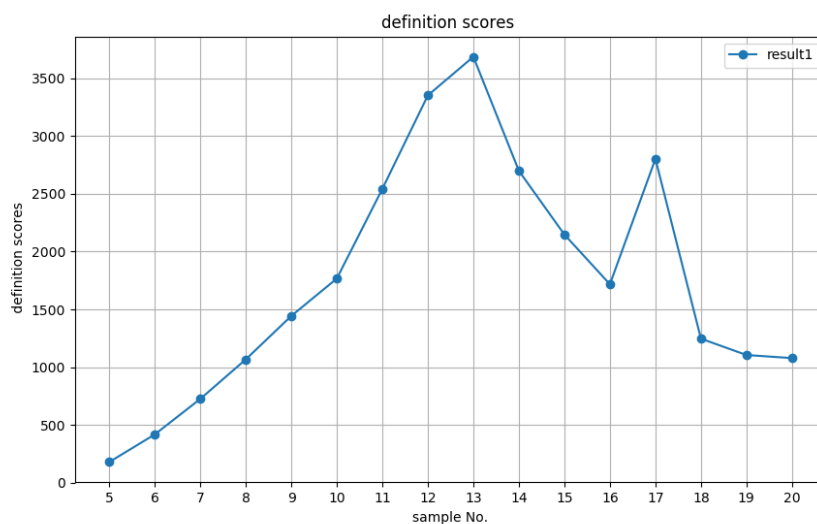
● 算法一：Sobel 算子统计法

尝试下在三个图集下的效果。

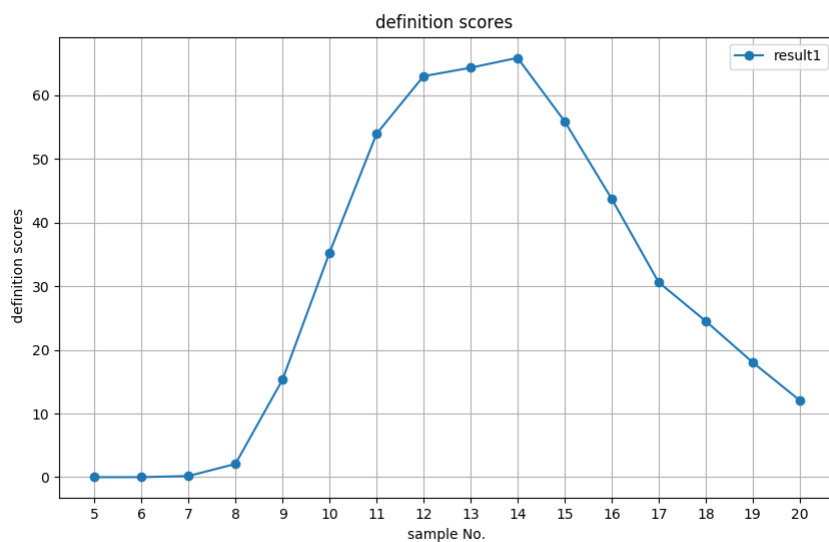
图集 A:



图集 B:



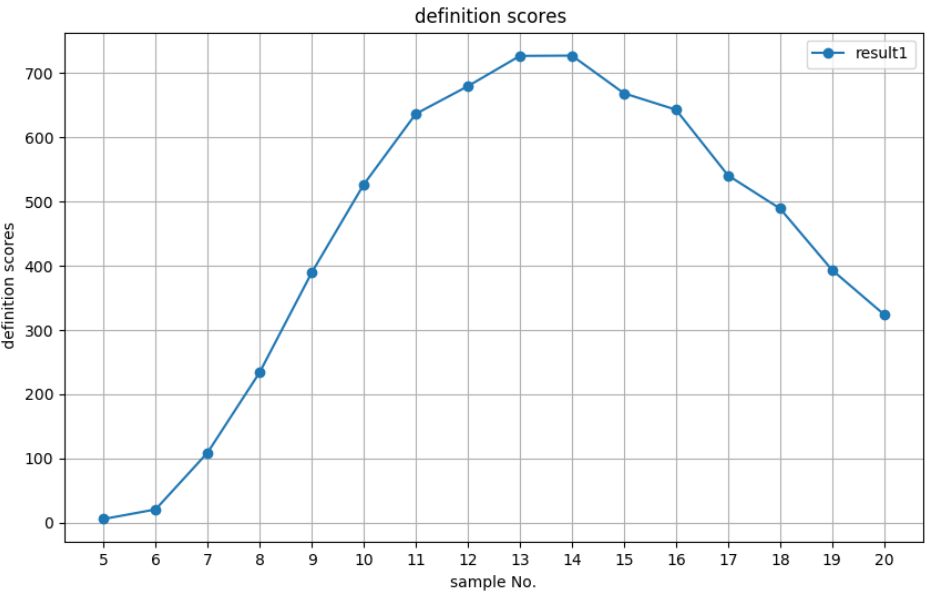
图集 C:



可以观察到，对于图集 C，即无灯光影响下的采样图像，该评价函数可以呈现与人眼评价结果相同的效果。即水平距离为 12、13、14 厘米时，图像的清晰度最高。

● 算法二：Tenengrad 函数

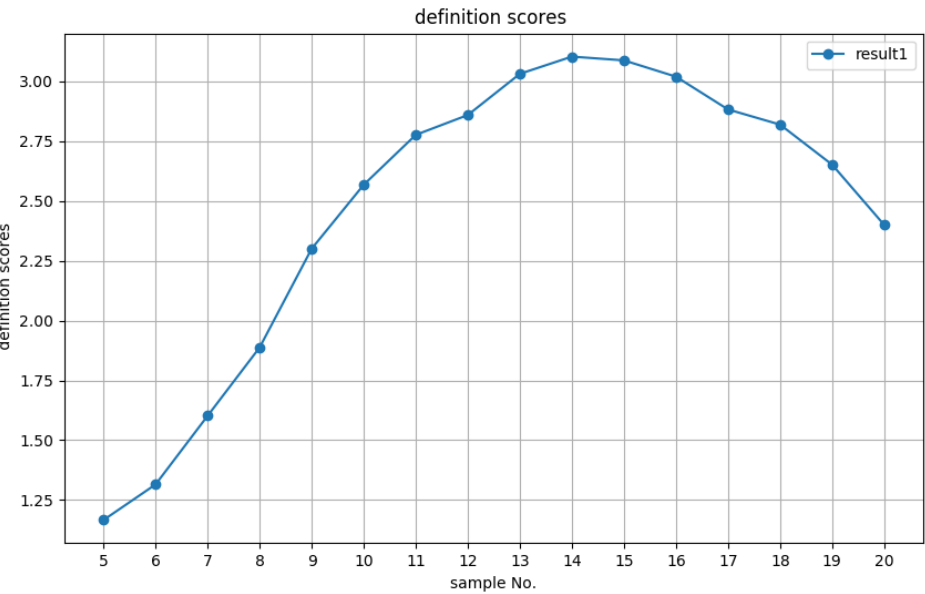
图集 C:



可观察到，评价函数值越高，图像越清晰。与人眼评价结果相同。其中 12、13、14cm 下清晰度最高。

● 算法三：SMD2 （灰度方差乘积）函数

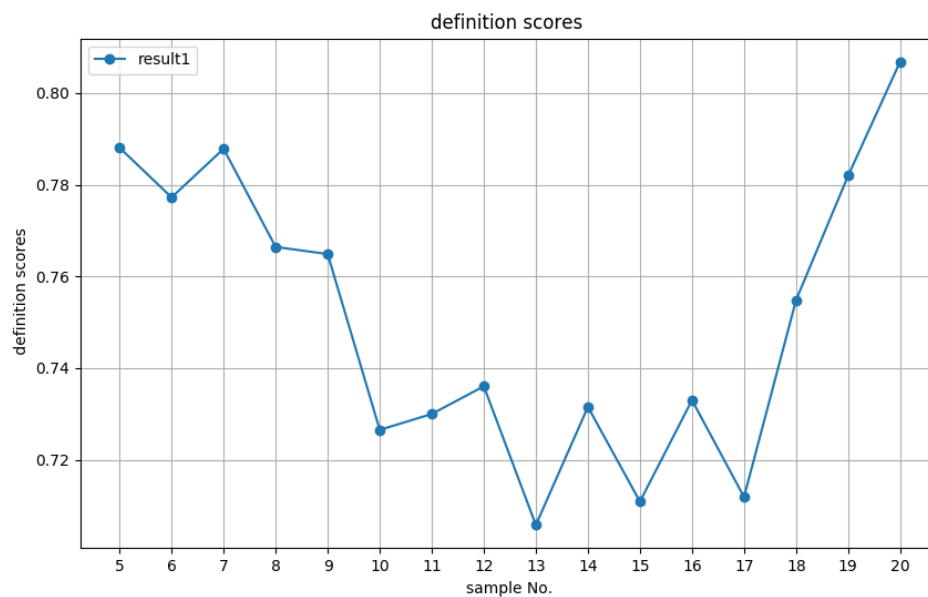
图集 C:



可观察到，评价函数值越高，图像越清晰。与人眼评价结果相同。其中 13、14、15cm 下清晰度最高。

● 算法四：基于方差的模糊程度函数

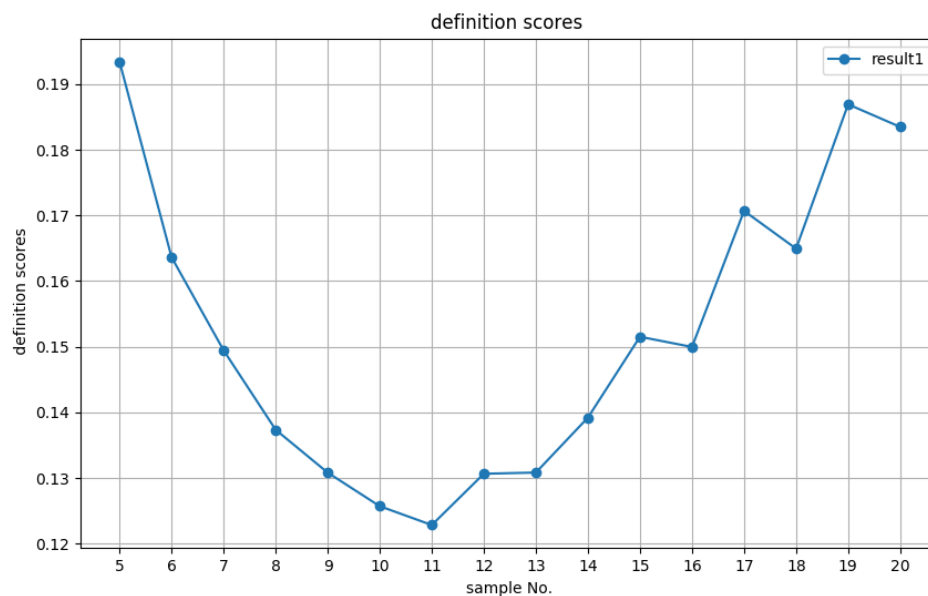
图集 C:



可观察到，评价函数值越低，图像越清晰。13cm 下清晰度最高。

● 算法五：Re-blur 函数

图集 C:



可观察到，评价函数值越低，图像越清晰。9、10、11cm 下清晰度最高。

四.运行时间

| 算法 | 平均运行时间 | 平均运行时间（C语言） |
|----|--------|-------------|
|----|--------|-------------|

| | | |
|---------------------|---------|---------|
| 算法一：Sobel 算子统计法 | 10.75ms | 3.9ms |
| 算法二：Tenengrad 函数 | 19.2ms | 17.25ms |
| 算法三：SMD2 （灰度方差乘积）函数 | 43.2ms | 1.75ms |
| 算法四：基于方差的模糊程度函数 | 22.85ms | 6.55ms |
| 算法五：Re-blur 函数 | 93.7ms | 14.0ms |