OpenCV距离变换函数：distanceTransform()介绍

原创 机器视觉001 2017-11-08 21:41:41 15522 收藏 20 版权

分类专栏： OpenCV 文章标签： OpenCV distanceTransform

OpenCV距离变换函数：distanceTransform()介绍

OpenCV 距离变换函数：distanceTransform()介绍

原理介绍：

距离变换于 1966 年被学者首次提出，目前已被广泛应用于图像分析、计算机视觉、模式识别等领域，人们利用它来实现目标细化、骨架提取、形状插值及匹配、粘连物体的分离等。距离变换是针对二值图像的一种变换，是计算并标识空间点（对目标点）距离的过程，它最终把二值图像变换为灰度图像(其中每个栅格的灰度值等于它到最近目标点的距离)。在二维空间中，一幅二值图像可以认为仅仅包含目标和背景两种像素，目标的像素值为 1，背景的像素值为 0；距离变换的结果不是另一幅二值图像，而是一幅灰度级图像，即距离图像，图像中每个像素的灰度值为该像素与距其最近的背景像素间的距离。

现有的距离变换算法主要采用两类距离测度：**非欧式距离**和**欧式距离**。前者常用的有城市街区、棋盘、倒角等距离，算法采用串行扫描实现距离变换，在扫描过程中传递最短距离信息。这些算法简单快速，易于实现，但得到的仅仅是欧式距离变换（EDT）的一种近似值，在很多应用中不能满足精度要求。欧式距离变换（EDT）精度高，与实际距离相符，应用更广泛。为此，很多学者也在研究高效快速的真实 EDT 算法。

函数原型：

```
void cv::distanceTransform (
InputArray  src,
OutputArray  dst,
OutputArray  labels,
int  distanceType,
int  maskSize,
int  labelType = DIST_LABEL_CCOMP
)
```


参数介绍：

src：8 位单通道二值图像。
dst：输出距离图像。8 位整型或 32 位浮点型单通道图像。
labels：输出 2D 的标签（离散 Voronoi 图），类型为 CV_32SC1。
distanceType：距离类型，取值如下：

Enumerator	
DIST_USER	User defined distance.
DIST_L1	distance = |x1-x2| + |y1-y2|
DIST_L2	the simple euclidean distance
DIST_C	distance = max(|x1-x2|,|y1-y2|)
DIST_L12	L1-L2 metric: distance = 2(sqrt(1+x*x/2) - 1))
DIST_FAIR	distance = c^2(|x|/c-log(1+|x|/c)), c = 1.3998
DIST_WELSCH	distance = c^2/2(1-exp(-(x/c)^2)), c = 2.9846
DIST_HUBER	distance = |x|<c ? x^2/2 : c(|x|-c/2), c=1.345

maskSize：距离变换掩模大小，取值如下表：

Enumerator	
DIST_MASK_3	mask=3
DIST_MASK_5	mask=5
DIST_MASK_PRECISE	

暂时不支持 DIST_MASK_PRECISE。DIST_L1 或 DIST_C 距离类型模式下，3×3 掩模和 5×5 掩模的结果一样。

labelType：标签类型，取值如下表：

Enumerator	
DIST_LABEL_CCOMP	each connected component of zeros in src (as well as all the non-zero pixels closest to the connected component) will be assigned the same label
DIST_LABEL_PIXEL	each zero pixel (and all the non-zero pixels closest to it) gets its own label.

具体代码：

```
Mat lv_MatImageDist = Mat(lv_MatImageGray.size(), CV_32F);
cv::cvtColor(lv_MatImageRead, lv_MatImageGray, cv::COLOR_RGB2GRAY, 0);
cv::blur(lv_MatImageGray, lv_MatImageGray, Size(15, 15));
cv::threshold(lv_MatImageGray, lv_MatImageGray, lv_nTrackBarPos, 255.00, THRESH_BINARY);
// 距离变换
cv::distanceTransform(lv_MatImageGray, lv_MatImageDist, noArray(), DIST_L1, DIST_MASK_3, CV_32F);
// 归一化距离图像
cv::normalize(lv_MatImageDist, lv_MatImageDist, 255.00, 0.00, CV_MINMAX);
lv_MatImageDist.convertTo(lv_MatImageNow, CV_8U);
lv_pOpenCVDlg->HV_ImageShow(lv_MatImageNow, lv_strNameWindow);
```