基于几何基元的匹配方法（Geometric primitive based matching method）

步骤

1、提取轮廓

一个目标物体轮廓上所有点的点集（坐标位置） S={ (xi,yi) | i=1,2,3...m}

2、轮廓分割

将点集分割成线段、圆弧

3、基元拟合

用点集拟合出用几何参数表达的形状方程

4、基元匹配

利用几何方程参数进行匹配

一、提取轮廓（contour）

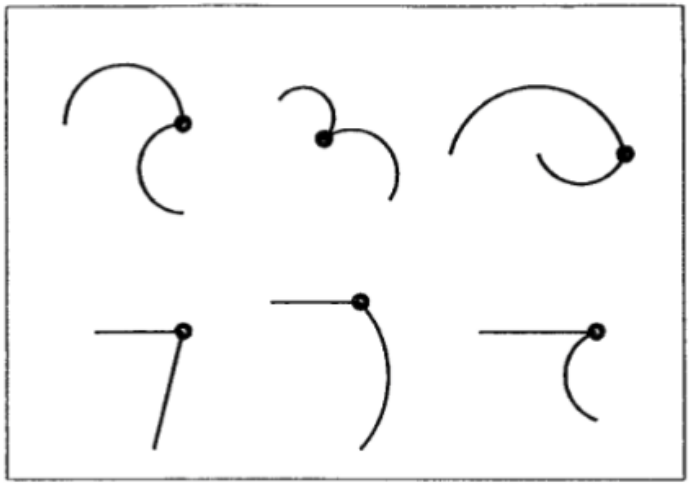
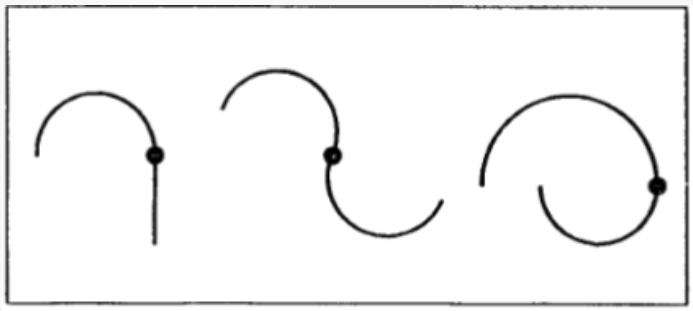
```
contours，hierarchy = cv2.find_contours(thresh, 0.5)
#contours是图像中所有轮廓的Python列表。每个单独的轮廓是对象的边界点的（x，y）坐标的Numpy阵列。
```

> Suzuki S , Be K . Topological structural analysis of digitized binary images by border following[J]. Computer Vision Graphics and Image Processing, 1985, 30(1):32-46.

二、轮廓分割

对于给定数量的基元（n），可以是弧或线段，目标是划分边界数据集（按逆时针顺序排列）：

轮廓是几何基元的集合，关键是找到分段点

分割点	图示
拐点：轮廓的切线是不连续的	
平滑连接：其切线是连续的，曲率不连续	

降噪

链码表示中，代表切线方向的“代码”被分配给曲线上的每个点，因此差分链码是两个相邻点之间的切线方向的变化。8链码对噪声非常敏感，使用噪声过滤技术来平滑数据。

每个轮廓上点的坐标取其领域h点集的坐标均值。

$$\begin{aligned} x_j &= \frac{1}{2h+1}\sum_{j-h\le l\le j+h} x_l \\ y_j &= \frac{1}{2h+1}\sum_{j-h\le l\le j+h} y_l \end{aligned} \tag{19}$$

其中h是点pi处领域宽度，邻域平均的应用具有使边界率变化变弱。

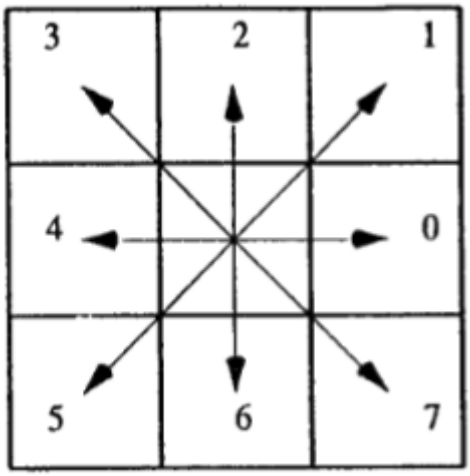
链码

> 从边界（曲线）起点S开始，按顺时针方向观察每一线段走向，并用相应的指向符表示，结果就形成表示该边界（曲线）的数码序列，称为原链码。

$$\zeta_{j,k} = \tan^{-1}\left[\frac{y_j - y_{j-k}}{x_j - x_{j-k}}\right], \quad j = 1, 2, \ldots, m \tag{1}$$

m个点的序列描述闭合轮廓。S的链码由以下序列组成，k为支撑长度，下图为k=1的情况。

k越大噪声影响越小



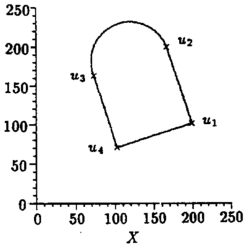
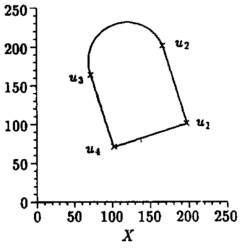
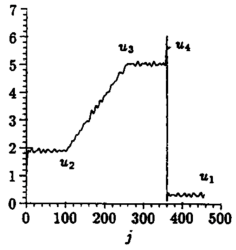
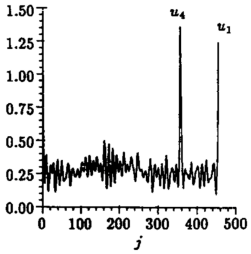
差分链码

$$\delta_{j,k} = \tan^{-1}\left[\frac{y_{j+k} - y_j}{x_{j+k} - x_j}\right] - \tan^{-1}\left[\frac{y_j - y_{j-k}}{x_j - x_{j-k}}\right]$$
$$j = 1, 2, \ldots, m$$

其中δj，k是在点pj处具有支撑长度k的切线角的变化。

差分链码可以反应切线方向的变化率。

具体流程

原始轮廓	均值滤波	链码	差分链码
			
右下角为U1，逆时针依次为U2、U3、U4		平滑连接成为倾斜段的两个端点	δj，k序列中的两个尖峰表示拐点

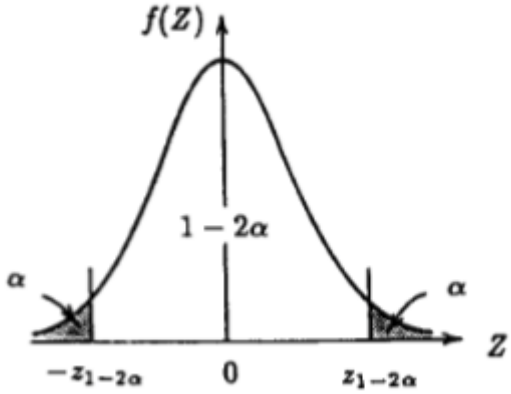
拐点的提取

$$\bar{\delta}_{j,k} = |\delta_{j,k} - \mu_\delta|, \quad j = 1, 2, \ldots, m$$

差分链码减去均值u的绝对值、代表偏离平均值的绝对值，用绝对值建立高斯分布，在某个置信范围内选取阈值。将偏离均值最大的一部分点作为分段点。

$$T_b = z_{1-2\alpha}\sigma_b \tag{2}$$

阈值由标准差求得



非极大值抑制判断标准：大于阈值且为邻域最大值

$$\begin{aligned}&\tilde{\delta}_{j,k} \geq T_{\tilde{b}}\\ &\text{and:}\\ &\tilde{\delta}_{j,k} > \tilde{\delta}_{l,k}, \quad \text{for } j-k \leq l \leq j+k, \quad \text{and } l \neq j\end{aligned} \tag{3}$$

符合条件的点视为拐点。

平滑连接的分段点

为了稳定后续操作，首先通过将比例因子乘以切线$，k来标准X和Y方向上的比例。

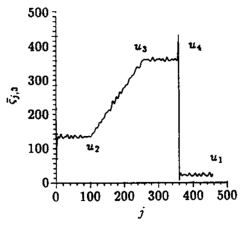
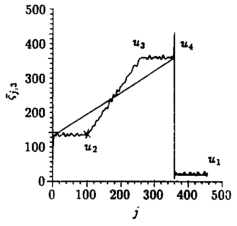
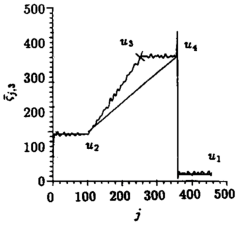
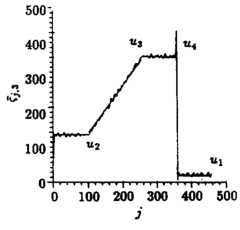
$$\bar{\zeta}_{j,k} = \frac{m}{2\pi}\zeta_{j,k}, \quad j = 1, 2, \ldots, m \tag{4}$$

分段线性逼近，得到平滑连接点。数据和近似函数之间的最大误差是检测分裂点的标准。

$$E_{\infty}\left(u_1, u_{i+1}\right) = \max_{j} \epsilon_{i,j} \tag{5}$$

逼近误差，其中ei，j是区间i的第j个点与其近似函数之间的欧几里德距离，并且（U1，Ui+1）是区间的两个端点。

达到分段数量或者误差低于某一个值时停止分裂。

三、基元拟合

圆弧：近似段不平行于X轴

线段：近似段与X轴平行（斜率为零）。

$$\begin{aligned}&e_j^2 = \left(f\left(x_j, y_j\right)\right)^2\\ &\text{where:}\\ &f(x,y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0\end{aligned} \tag{6}$$

直线的误差

$$\epsilon_{i,j}^2 = (x_{i,j}\cos\theta_i + y_{i,j}\sin\theta_i - d_i)^2, \quad \text{for } j = 1, 2, \ldots, m_i, \quad \text{and } i \in \mathcal{I}_l$$

圆弧的误差

$$e_{i,j}^2 = \left(\left((x_{i,j} - a_i)^2 + (y_{i,j} - b_i)^2\right)^{1/2} - r_i\right)^2, \quad \text{for } j = 1, 2, \ldots, m_i, \quad \text{and } i \in \mathcal{I}_c$$

调整分段点

BA-1

Step 0. Set the step-size $D = 1$.

Step 1. For $i = 1$ to n, do:

1.1. Compute

$$\epsilon_0' = (E_2(u_{i-1}, u_i - 1) + E_2(u_i, u_{i+1} - 1)),$$

$$\epsilon_1' = (E_2(u_{i-1}, u_i - 1 - D) + E_2(u_i - D, u_{i+1} - 1)), \text{ and}$$

$$\epsilon_2' = (E_2(u_{i-1}, u_i - 1 + D) + E_2(u_i + D_i, u_{i+1} - 1)).$$

1.2. If $\min\{\epsilon_1', \epsilon_2'\} < \epsilon_0'$, then:
if $\epsilon_2' > \epsilon_1'$, set $u_i = u_i - D$, and go to Step 1.1;
else set $u_i = u_i + D$, and go to Step 1.1.

Step 2. If any break points have been adjusted, repeat Step 1;
else stop.

选择最小的步长，D = 1.在步骤1中，断点向左或向右逐点移动，直到每对相邻间隔的误差范数最小化。步骤2检查在前一次迭代中是否有任何改进。如果误差范数已经减小，则再执行一次迭代;否则，算法终止。由于这是严格减少的过程，因此算法无法循环。

改进算法先用大步长

BA-2

Step 0. For $D = D_{\max}$ down to 1, do
BA-1.

相似实验结果

1.提取轮廓

2.多边形近似

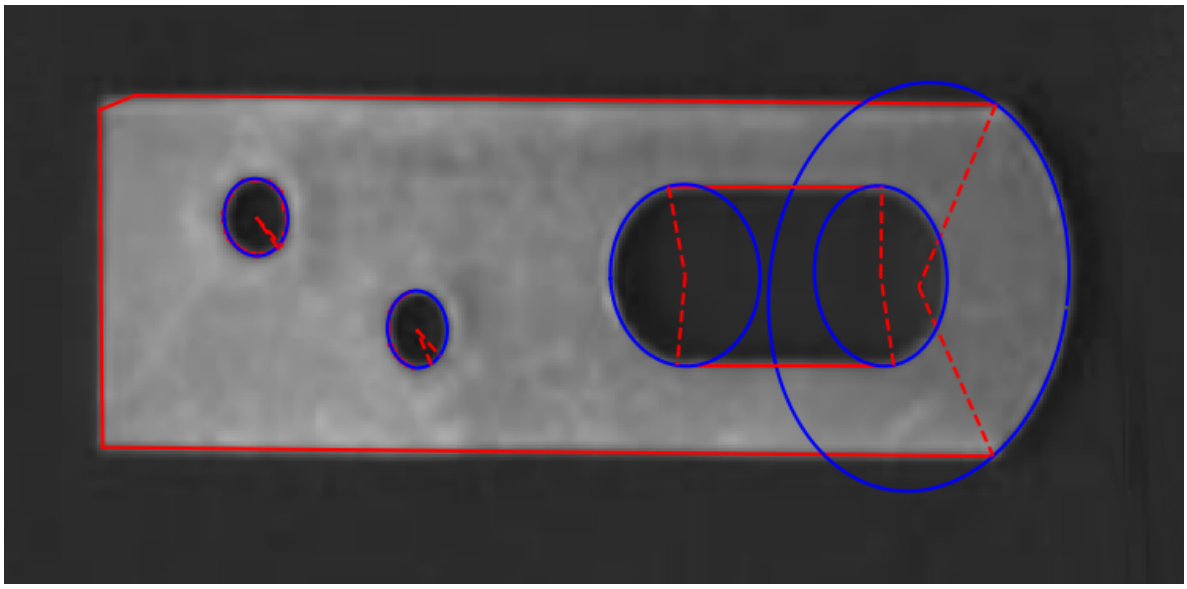
```
epsilon = elength* cv2.arcLength(cnt, True)  # 多边形拟合偏差epsilon =0.05*周长
```

3.轮廓分割

阈值T=mean_disten点集平均距离

当两点之间距离大于mean_disten时视为线段，点集数量大于6的时候拟合为椭圆

4.基元拟合用最小二乘法



结果为几何基元

$$CP = \{GF_i | i = 1, 2, \ldots, n\} \tag{7}$$

$$GF_i = \begin{cases} L_i, & i \in I_1 \\ C_i, & i \in I_c \end{cases} \tag{8}$$

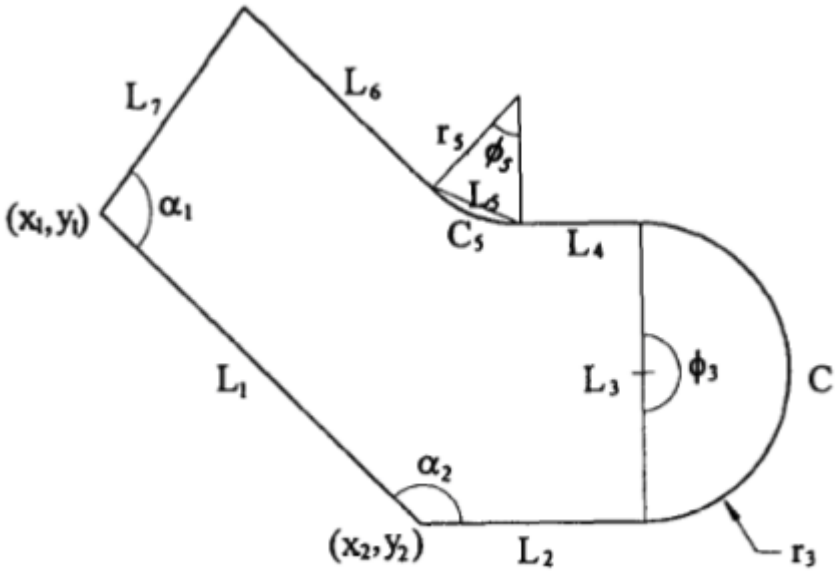
四、基元匹配

The shape model

描述CP

$$L_i = \{k_i, \alpha_i\}, i \in I_1 \ (k_i = l_i / l_1) \tag{9}$$

$$C_i = \{\lambda_i, \alpha_i, \pm\phi_i\}, i \in I_c \ (\lambda_i = r_i / l_1) \tag{10}$$



圆弧弦长

$$k_i = 2 \times \lambda_i \times \sin(\phi_i / 2), i \in I_c \tag{11}$$

> 法线式：**x·cosα+ysinα-p=0【适用于不平行于坐标轴的直线】**
>
> 过原点向直线做一条的垂线段，该垂线段所在直线的倾斜角为α，p是该线段的长度

$$L_i : X\cos\omega_i + Y\sin\omega_i = \delta_i, i = 1, 2, \ldots, n \tag{12}$$

$$C_i : (X - u_i)^2 + (Y - v_i)^2 = \lambda_i^2, i \in I_{\mathrm{c}} \tag{13}$$

假设L1为单位线段

$$\begin{pmatrix} X^s \\ Y^s \end{pmatrix} = l_1 \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{pmatrix} \begin{pmatrix} X \\ Y \end{pmatrix} + \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \tag{14}$$

$$\begin{aligned} &L_i : (X^s - x_1)\cos(\omega_i + \theta_1) + (Y^s - y_1)\sin(\omega_i + \theta_1) \\ &= \delta_i l_1, i \in I_c \end{aligned} \tag{15}$$

$$C_i : (X^s - \overline{u}_i l_1 - x_1)^2 + (Y_s - \overline{v}_i l_1 - y_1)^2 = (\lambda_i l_1)^2, i \in I_c \tag{16}$$

形状仿射变换变换后线方程 与（比例）（角度）（位移）参数有关

$$E_{i,j} = \begin{cases} \left|(x_{i,j} - x_1)\cos(\theta_1 + \omega_i) + (y_{i,j} - y_1)\sin(\theta_1 + \omega_i) - \delta_i l_1\right|, & j = 1, 2, \ldots, m_i \quad i \in I_i \\ \left|(x_{i,j} - \overline{u}_i l_1 - x_1)\cos(\psi_{i,j}) + (y_{i,j} - \overline{v}_i l_1 - y_1)\sin(\psi_{i,j}) - \lambda_i l_1\right|, & j = 1, 2, \ldots, m_i \quad i \in I_c \end{cases} \tag{17}$$

$$\begin{aligned} &\min_{x_1 \cdot y_1, t_1, E_m \geq 0, \theta_1} E_{\max} \\ &\text{subject to } E_{i,j} \leq E_{\max}, j = 1, 2, \ldots, m_i; i = 1, 2, \ldots, n \end{aligned} \tag{18}$$

最小化误差。得到匹配形状的位置、比例、角度

> [1] Ventura J A , Wan W . Accurate matching of two-dimensional shapes using the minimal tolerance zone error[J]. Image and Vision Computing, 1997, 15(12):889-899.
>
> [2] Chen J M , Ventura J A , Wu C H . Segmentation of planar curves into circular arcs and line segments[J]. Image and Vision Computing, 1996, 14(1):71-83.