Lesson 3: Shape descriptors

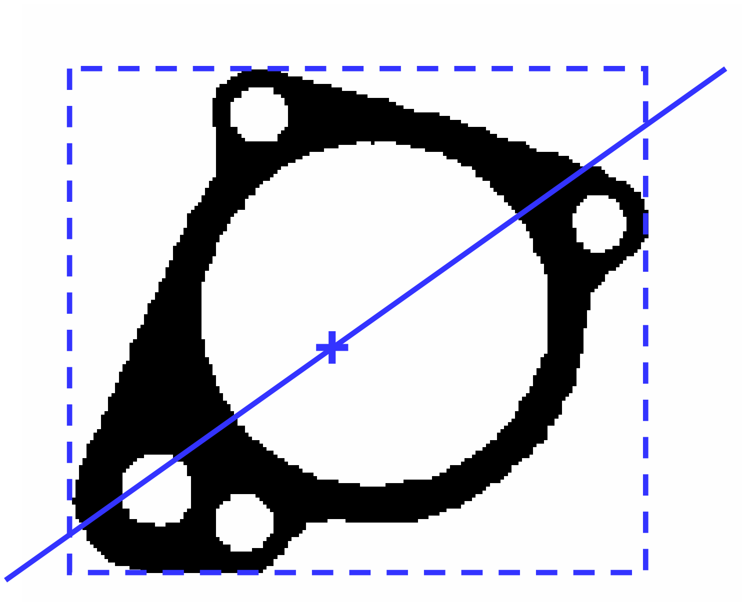
1. Position dependent
   - Bounding box
   - Centroid
   - Orientation

2. Position independent
   - Image moments
   - Perimeter
   - Elongation
   - Euler number

Introduction: descriptors

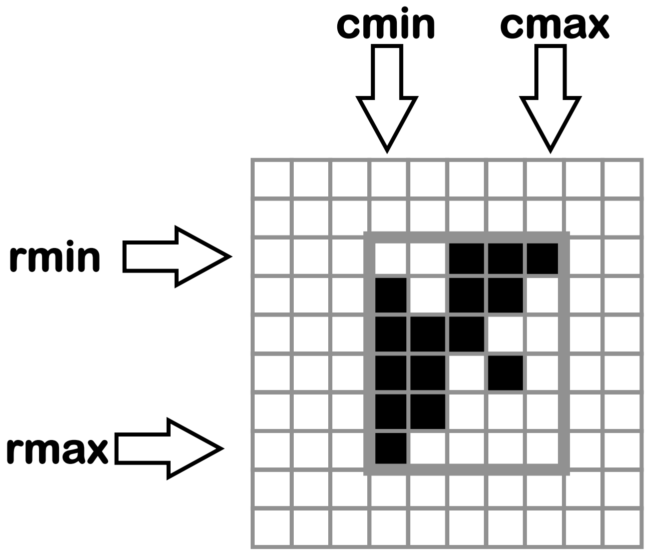
- Properties than can be used to **identify** and **localize** objects:
  - Position dependent
    - » Bounding box
    - » Centroid
    - » Orientation
  - Position independent
    - » Image moments
    - » Perimeter
    - » Elongation
    - » Holes (Euler number)
- Object identification based on descriptors is possible if:
  - Limited and known number
  - Stable positions
  - Isolated
  - Completely visible

- Objects characterized by their silhouette:

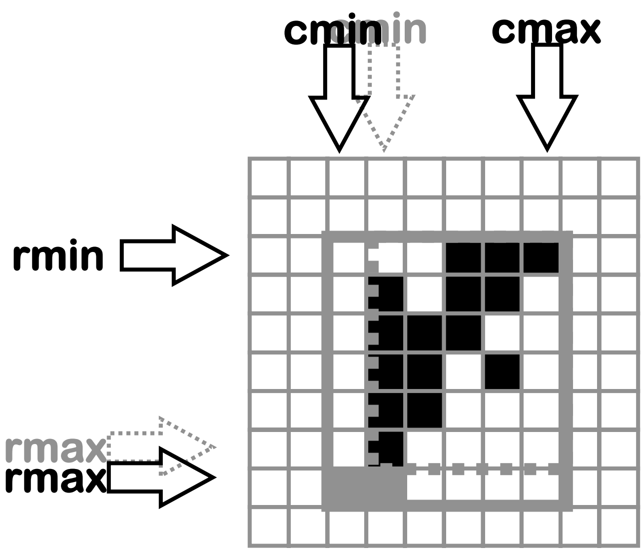


- Many descriptors can be computed during connectivity analysis

Bounding box



Sensible al ruido



- Simple to compute during connectivity:

```
crear_blob(s):
     rmin = rmax = FILA(s)
     cmin = COLUMNA(s)
     cmax = FINAL(s)

anadir_segmento_a_blob(b,s):
     si rmin > FILA(s)
        rmin = FILA(s)
     fsi
     ....

fusionar_blobs(b1, b2):
     rmin = min(rmin_1, rmin_2)
     rmax = max(rmax_1, rmax_2)
     cmin = min(cmin_1, cmin_2)
     cmax = max(cmax_1, cmax_2)
```

- Invariant to:
  – translation?
  – rotation?
  – scale?

Image moments

- **Definition:** moments of a continuous function:

$$\mathtt{m}_p = \int_{-\infty}^{\infty} x^p f(x) dx$$

$$p = 0, 1, 2, \ldots$$

- ***P(x)***, expected value (**mean**):

$$\mathtt{m}_1 = \int_{-\infty}^{\infty} x P(x) dx = \mu$$

$\Rightarrow$ $\mathtt{m}_0$?

- Central moments:

$$\mathtt{m}'_p = \int_{-\infty}^{\infty} (x-\mu)^p f(x) dx$$

$$p = 0, 1, 2, \ldots$$

- ***P(x)***, **variance**:

$$\mathtt{m}'_2 = \int_{-\infty}^{\infty} (x-\mu)^2 P(x) dx = \sigma^2$$

$\Rightarrow$ $\mathtt{m}'_3$? $\mathtt{m}'_4$?

- In R^2:

$$\mathtt{m}_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx dy$$

$$p, q = 0, 1, 2, \ldots$$

- **Uniqueness theorem:** the sequence { $\mathtt{m}_{p,q}$ } is univocally determined by ***f(x,y)***.

{ $\mathtt{m}_{p,q}$ } uniquely characterize a function

Image moments

- In the case of **digital binary images**, they become products of powers of pixel coordinates:

$$m_{p,q} \simeq \sum i^p j^q$$

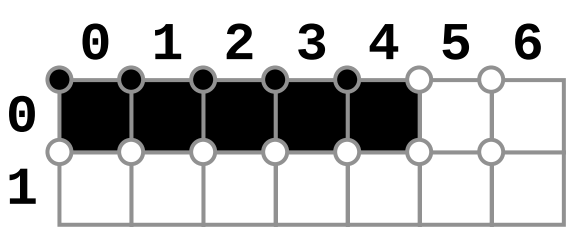
Of order 0: $\sum 1$ ← **Nu. of pixels**

1: $\sum i \quad \sum j$

2:

They uniquely characterize an object.

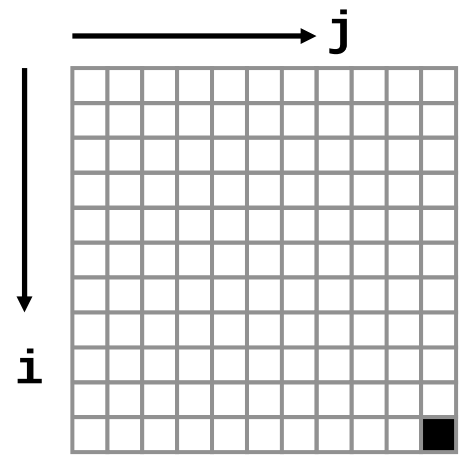
- Convention in digitization:



- Considering calibration: if S_x (S_y) its the distance that a point should move in the scene, in the ***x(y)*** direction, in order to move one pixel in the image:

$$\mathtt{Area} = S_x S_y \sum 1$$

- Because of precision, we use moments up to order 3.



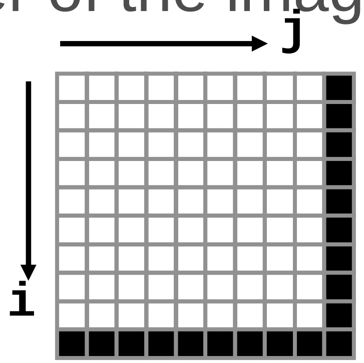
Use Floating point

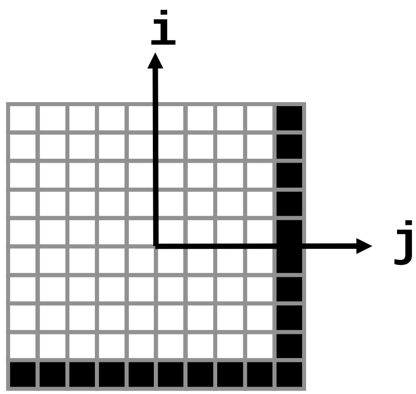
← $i^p j^q$ **large**

- Invariant to translation? To rotation? To scale change?

Image moments

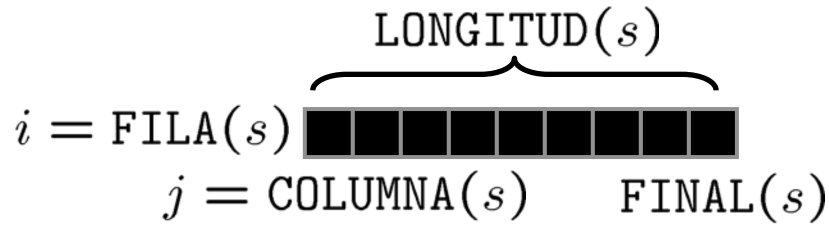
- Also because of precision, the origin of the reference system can be placed in the center of the image.





Smaller values, both positive and negative

- Can be computed segment-wise during connectivity:



$$m_{0,0,s} =$$
$$m_{1,0,s} =$$
$$m_{2,0,s} =$$
$$\ldots$$

- Efficient computation:

```
#define M max(R,C)
long x2[M], x3[M];

void precompute_powers()
{
  long i;

  for (i = 0; i < M; i++)
    x3[i] = i * (x2[i] = i*i);
}
```

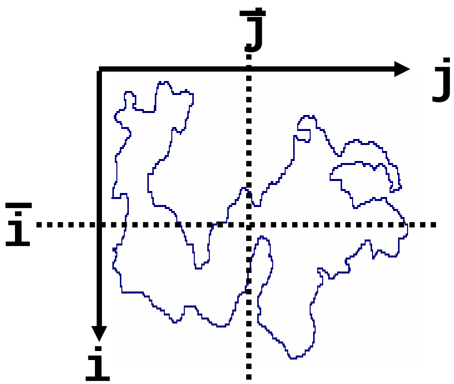
Centroid

- Planar objects, uniform density:

Averages of *i* and *j*:

$$\bar{i} = \frac{\sum i}{\sum 1} =$$

$$\bar{j} = \frac{\sum j}{\sum 1} =$$



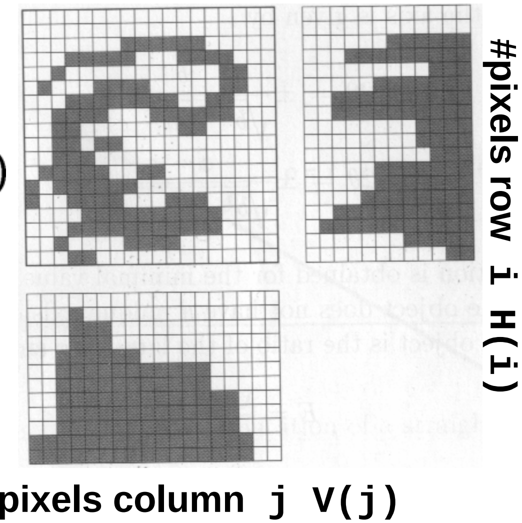
Use floating point

- May not be inside the object

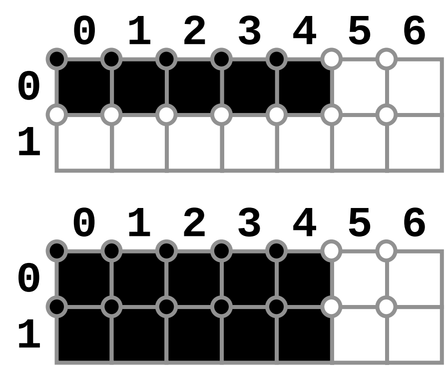
- Can be computed from the projections

$$\mathtt{H}(i) = \sum_{j=0}^{\mathtt{C}-1} \mathtt{B}(i,j)$$

$$\bar{i} = \frac{\sum i \cdot \mathtt{H}(i)}{\sum \mathtt{H}(i)}$$



- Effect of digitization:



$$\sum 1 = ?$$

$$\sum i = ?$$

$$\sum j = ?$$

$$\bar{i} = ?$$

$$\bar{j} = ?$$

Central moments

- **Central moments:** computed with respect to the centroid:

$$\mu_{p,q} = \sum (i-\bar{i})^p (j-\bar{j})^q$$

$$\mu_{0,0} = m_{0,0}$$
$$\mu_{1,0} =$$
$$\mu_{2,0} =$$
$$\mu_{1,1} =$$
$$\mu_{0,2} =$$
$$\cdots$$

- Invariant to translations:

$$i' = i + a$$

$$\overline{i'} = \frac{\sum i'}{\sum 1}$$
$$=$$
$$=$$
$$\overline{j'} = \cdots$$

$$\mu'_{p,q} = \sum (i'-\overline{i'})^p (j'-\overline{j'})^q$$
$$\cdots$$
$$= \sum (i-\bar{i})^p (j-\bar{j})^q$$
$$= \mu_{p,q}$$

Normalized moments

- **Normalized moments** (adjusted to scale):

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^{\gamma}}$$

$$\gamma = \frac{p+q}{2} + 1$$

- f.e., area:

$$\eta_{0,0} = \frac{\mu_{0,0}}{\mu_{0,0}^{1}} = 1$$

- Invariant to change of scale (cannot distinguish between different sizes):

$$i' = \alpha i$$

$$\overline{i'} = \frac{\sum i'}{\sum 1} = \frac{\alpha \sum i}{\sum 1} = \alpha \overline{i}$$

$$\overline{j'} = \ldots$$

$$\mu'_{p,q} = \frac{\sum (i' - \overline{i'})^p (j' - \overline{j'})^q}{\mu_{0,0}^{\gamma}}$$

$$\ldots$$

Invariant moments

- **Invariant moments to translations and rotations:**

$$
\begin{aligned}
\phi_0 &= \mu_{0,0} \\
\phi_1 &= \mu_{2,0} + \mu_{0,2} \\
\phi_2 &= (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2 \\
\phi_3 &= (\mu_{3,0} - 3\mu_{1,2})^2 + (3\mu_{2,1} - \mu_{0,3})^2 \\
\phi_4 &= (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{2,1} + \mu_{0,3})^2 \\
\phi_5 &= (\mu_{3,0} - 3\mu_{1,2})(\mu_{3,0} + \mu_{1,2}) \left[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{2,1} + \mu_{0,3})^2\right] \\
&+ (3\mu_{2,1} - \mu_{0,3})(\mu_{2,1} + \mu_{0,3}) \left[3(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2\right] \\
\phi_6 &= (\mu_{2,0} - \mu_{0,2}) \left[(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2\right] \\
&+ 4\mu_{1,1}(\mu_{3,0} + \mu_{1,2})(\mu_{2,1} + \mu_{0,3})
\end{aligned}
$$

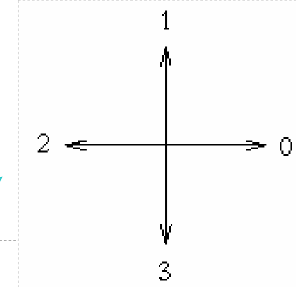
- To distinguish an object from its mirror image:

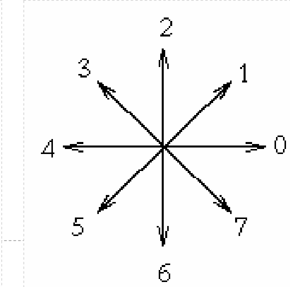
$$
\begin{aligned}
\phi_7 &= (3\mu_{2,1} - \mu_{0,3})(\mu_{3,0} + \mu_{1,2}) \left[(\mu_{3,0} + \mu_{1,2})^2 - 3(\mu_{2,1} + \mu_{0,3})^2\right] \\
&+ (\mu_{3,0} - 3\mu_{1,2})(\mu_{2,1} + \mu_{0,3}) \left[3(\mu_{3,0} + \mu_{1,2})^2 - (\mu_{2,1} + \mu_{0,3})^2\right]
\end{aligned}
$$

- Want them invariant to scale? use η in lieu of μ.
- Readability? logarithm of absolute value

$$\psi_i = \log \mid \phi_i \mid, \quad i = 0, 1, \ldots, 7$$

Perimeter





- Related with the **frontier** of a blob.

Obtaining the frontier:

1. Look for **P_0**, the pixel in the lowest column, within all pixels in the lowest row. Variable **`dir`** will store the previous motion along the frontier of the pixel prior to the current one:

   `dir = 3` (4-connectivity)

   `dir = 7` (8-connectivity)

2. Search in the neighborhood of the current pixel, counterclockwise, beginning with:

   `(dir + 3) mod 4` (4-connectivity)

   `(dir + 7) mod 8,` if **`dir`** is even (8-connectivity)

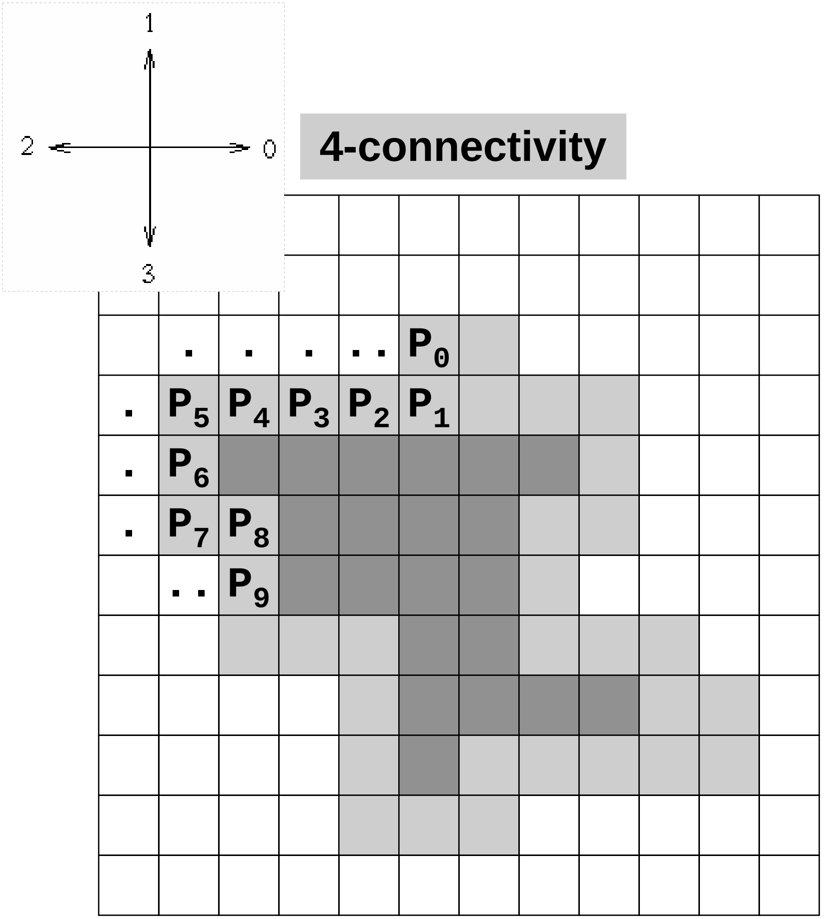
   `(dir + 6) mod 8,` if **`dir`** is odd (8-connectivity)

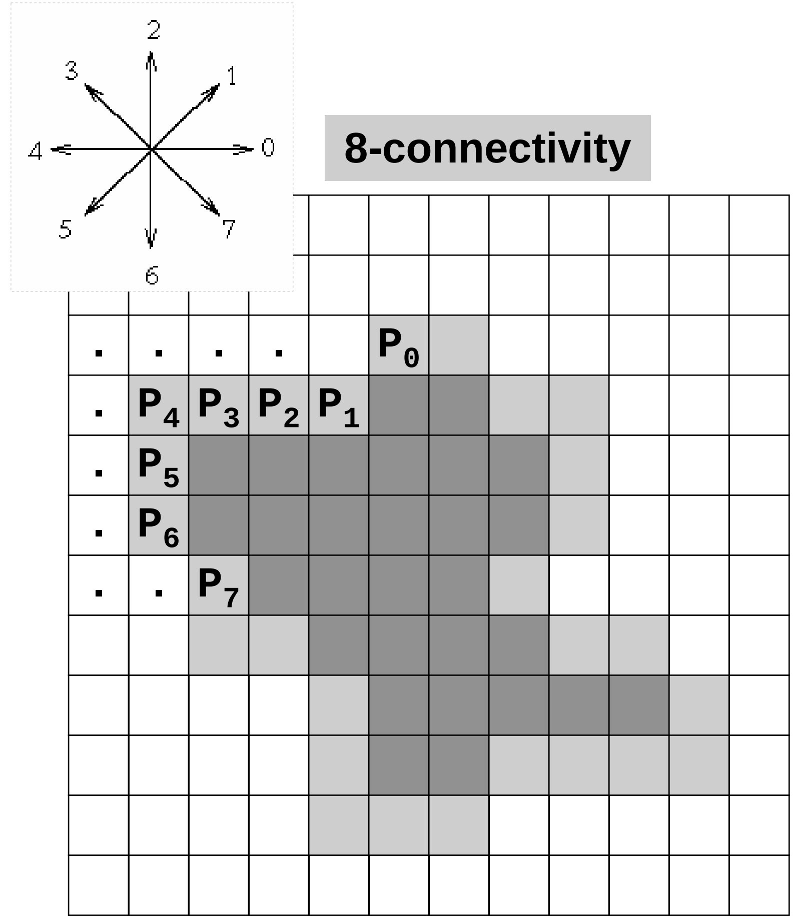
   The first found pixel is the next one in the frontier, **P_i**. Update **`dir`**

3. If **$(P_i = P_1) \wedge (P_{i-1} = P_0)$** , end. Otw, step 2.

4. Pixels **$P_0 ... P_{n-2}$** constitute the frontier.

Perimeter

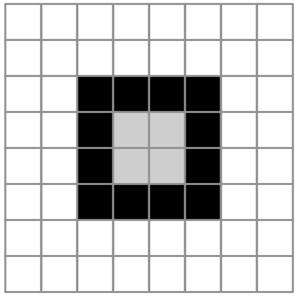




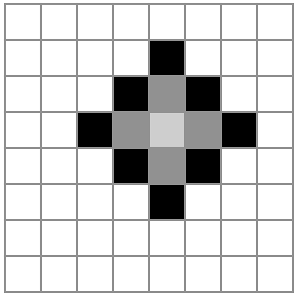
- Valid for regions of more than one pixel.
- 4-connectivity: sides adjacent to the frontier
- 8-connectivity: pixels adjacent to the frontier

Perimeter

- **Computing** perimeter:
  - Frontier pixels?
  - Sides adjacent to the frontier?
- Digitization results in a great variation in the value of the perimeter.

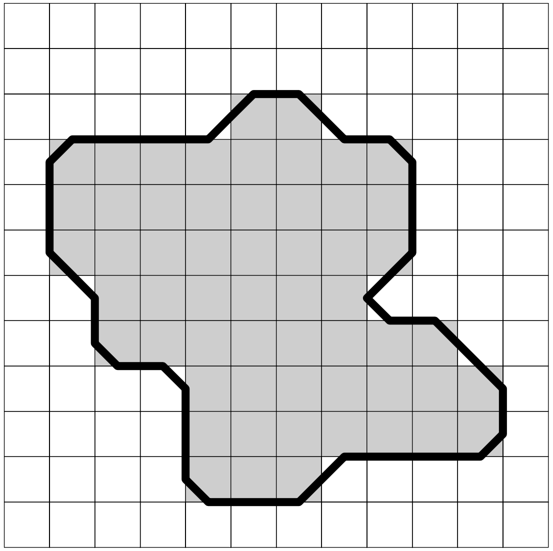


8-p = 12
4-p = 12
l = 16



8-p = 8
4-p = 12
l = 20

- This effect can be mitigated "cutting" corners:



N

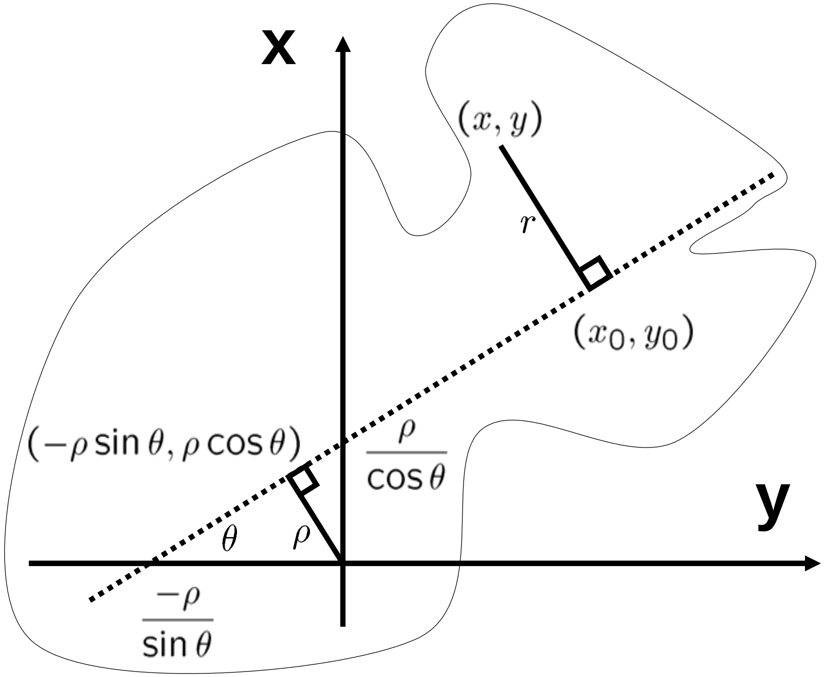
N_h horizontal frontiers
N_v vertical frontiers
N_c corners

$$P = S_x N_h + S_y N_v - \frac{S_x + S_y - \sqrt{S_x^2 + S_y^2}}{2} N_c$$

What about holes?

Orientation

- **Axis of minimum inertia:** of minimum order 2 moment.
- Which shapes have infinite?
- Which shapes have several?



- **Obtention**: straight line that minimizes the squared distances to the points of the object (total regression)

- In polar coordinates:

$$\rho = x\cos\theta + y\sin\theta$$

- Distance:

$$r^2 = (x\cos\theta + y\sin\theta - \rho)^2$$

- Minimize:

$$\begin{aligned}\chi^2 &= \sum r_{ij}^2 \\ &= \sum (j\cos\theta + i\sin\theta - \rho)^2\end{aligned}$$

- Differentiate with respect to ρ:

$$\begin{aligned}\frac{\partial \chi^2}{\partial \rho} &= \sum 2(j\cos\theta + i\sin\theta - \rho)(-1) \\ &= \cos\theta \sum j + \sin\theta \sum i - \rho \sum 1 \\ &= 0\end{aligned}$$

- The centroid belongs to the axis:

$$\rho = \bar{j}\cos\theta + \bar{i}\sin\theta$$

Orientation

- Minimize:

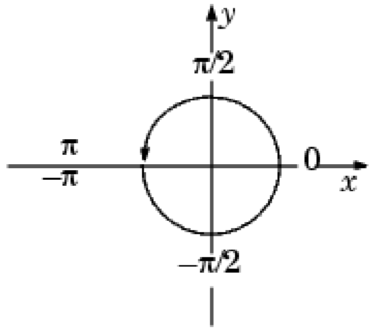
$$\chi^2 = \sum r_{ij}^2$$
$$= \sum (j \cos\theta + i \sin\theta - \bar{j}\cos\theta - \bar{i}\sin\theta)^2$$

- ...
- Solution:

$$\theta = \frac{1}{2} \cdot \tan^{-1}\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right)$$

- Ambiguity in direction:

rango de $\tan^{-1}$



$$\theta \in [-\pi/2, \pi/2]$$

~~atan() returns the arc tangent of x in the range -p/2 to p/2.~~

Watch it with the implementation of this solution (see **atan2**).

Elongation, compactness

- **Elongation:** measured from the inequality:

$$E = \frac{P^2}{A} \geq 4\pi$$

P: perimeter
A: area

- **Compactness:**

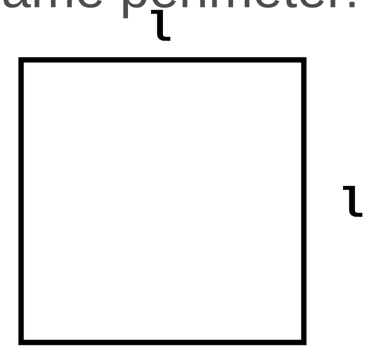
$$C = \frac{4\pi}{E} = \frac{4\pi A}{P^2} \leq 1$$

- The circle is the convex figure of minimum elongation (more compact):

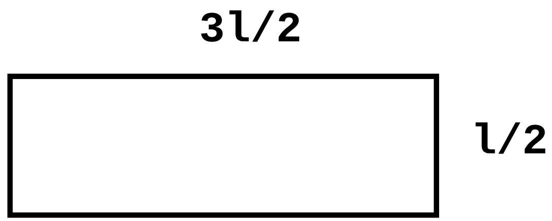
$$E =$$

$$C =$$

- A less elongated region (more compact) contains a larger area in the same perimeter:



$$E = \frac{(4l)^2}{l^2} = 16$$



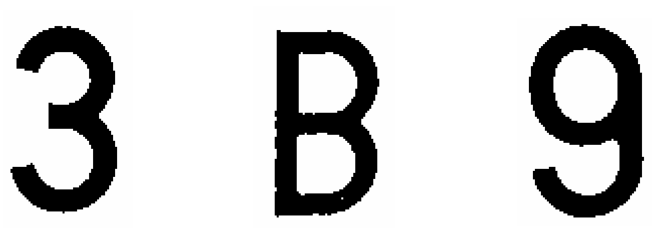
$$E = \frac{(4l)^2}{\frac{3l^2}{4}} = 64/3$$

- It is non-dimensional

Euler number

- Number of components – number of holes:

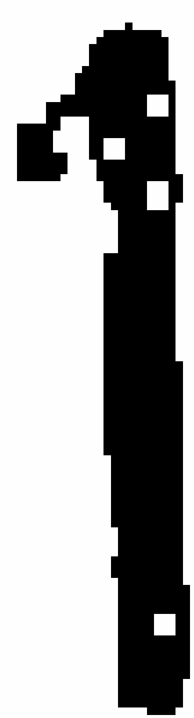
$$Eu = C - H$$



$$Eu = ?$$

- Invariant ***topological*** descriptor.
- Measurement with non random noise.

- Nevertheless, it is very sensitive to noise:



$$Eu = -3!$$

- Alternative: compute the total area of holes