GENERALIZING THE HOUGH TRANSFORM TO DETECT ARBITRARY SHAPES

D. H. Ballard

Pattern Recognition **Vol. 13 No. 2**

1981

What is the generalized Hough (Huff) transform used for?

- Hough transform is a way of encoding and detecting an arbitrary shape for later recall.
- Used specifically for object matching.
- Hough transform is invariant to scale changes, rotations, and foreground/background reversals.
- Hough transform can handle reasonably small occlusions.
- Hough transform can be easily composed form complex image recognition.

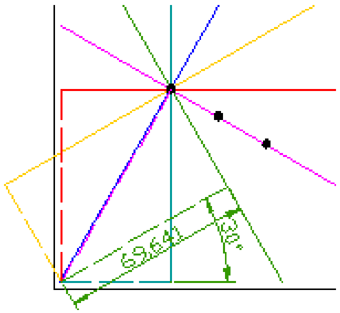
What is the procedure for a Hough transform?

- Transform
  - Threshold image to get a binary shape image.
  - Extract just the edge pixels of the shape image.
  - Create an R-Table of edge gradients that maps gradients to parameter space in parameter space.
  - Create and fill an accumulator with values from the shape.
- Recognition
  - Perform Hough transform for test shape and compare it’s accumulator to the set of possible accumulators.

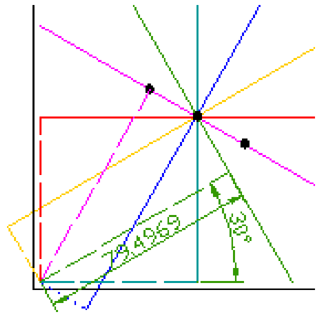
Original Hough Transform (lines and circles)

- Quick method to determine all of the lines in an edge image.
- Change general equation of a line or circle into a parametric model
- Create a parameter space accumulator.
- For each point on the edge parameterize and add a vote to the histogram.

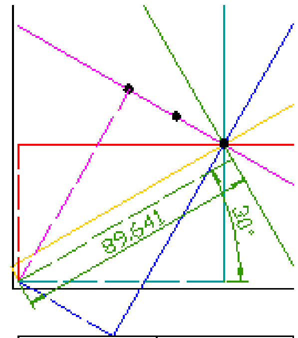
A Line Example



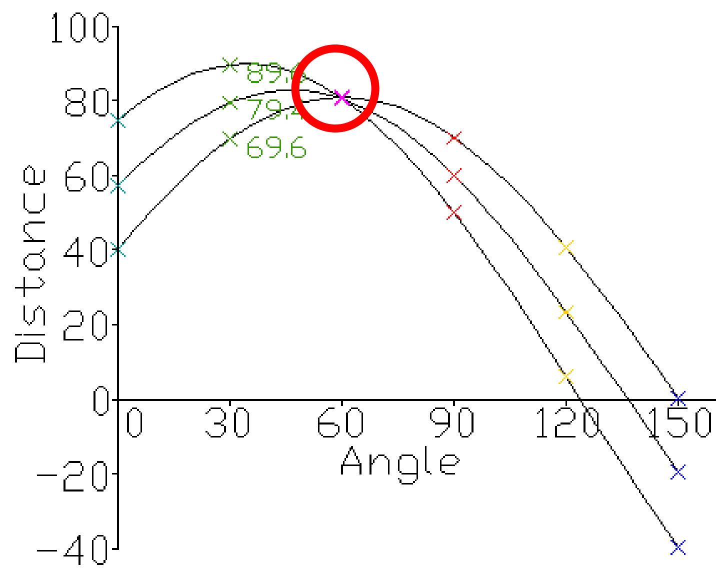
Angle	Dist.
0	40
30	69.6
60	81.2
90	70
120	40.6
150	0.4



Angle	Dist.
0	57.1
30	79.5
60	80.5
90	60
120	23.4
150	−19.5

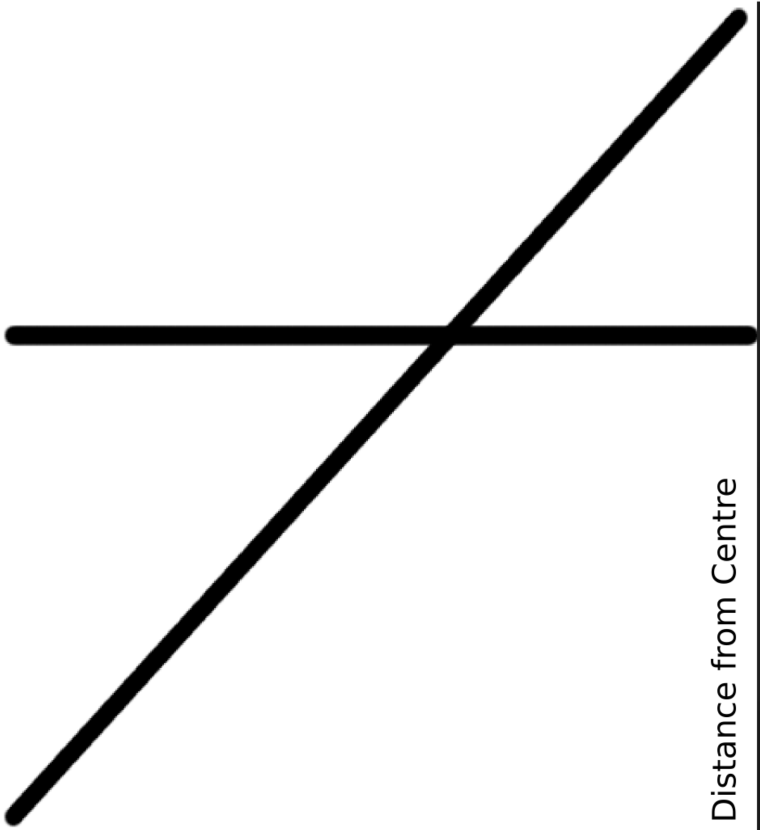


Angle	Dist.
0	74.6
30	89.6
60	80.6
90	50
120	6.0
150	−39.6

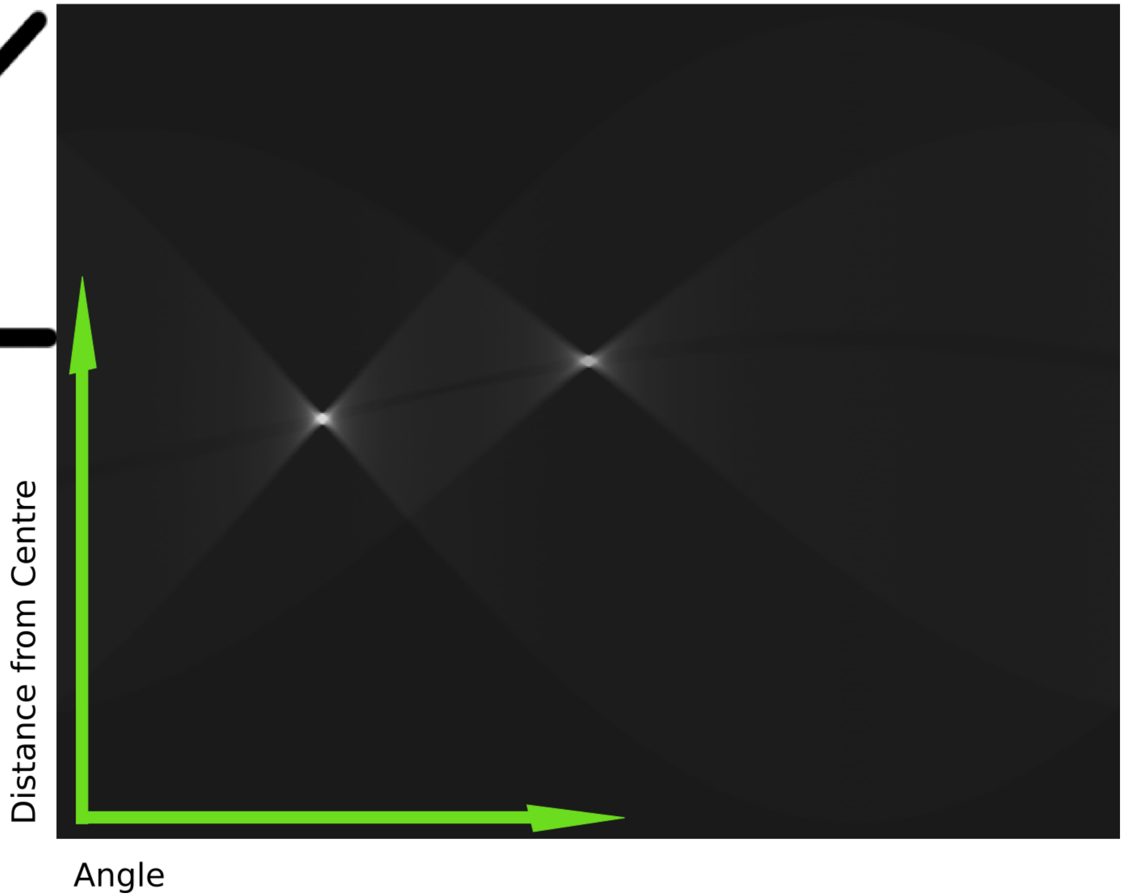


Another Example

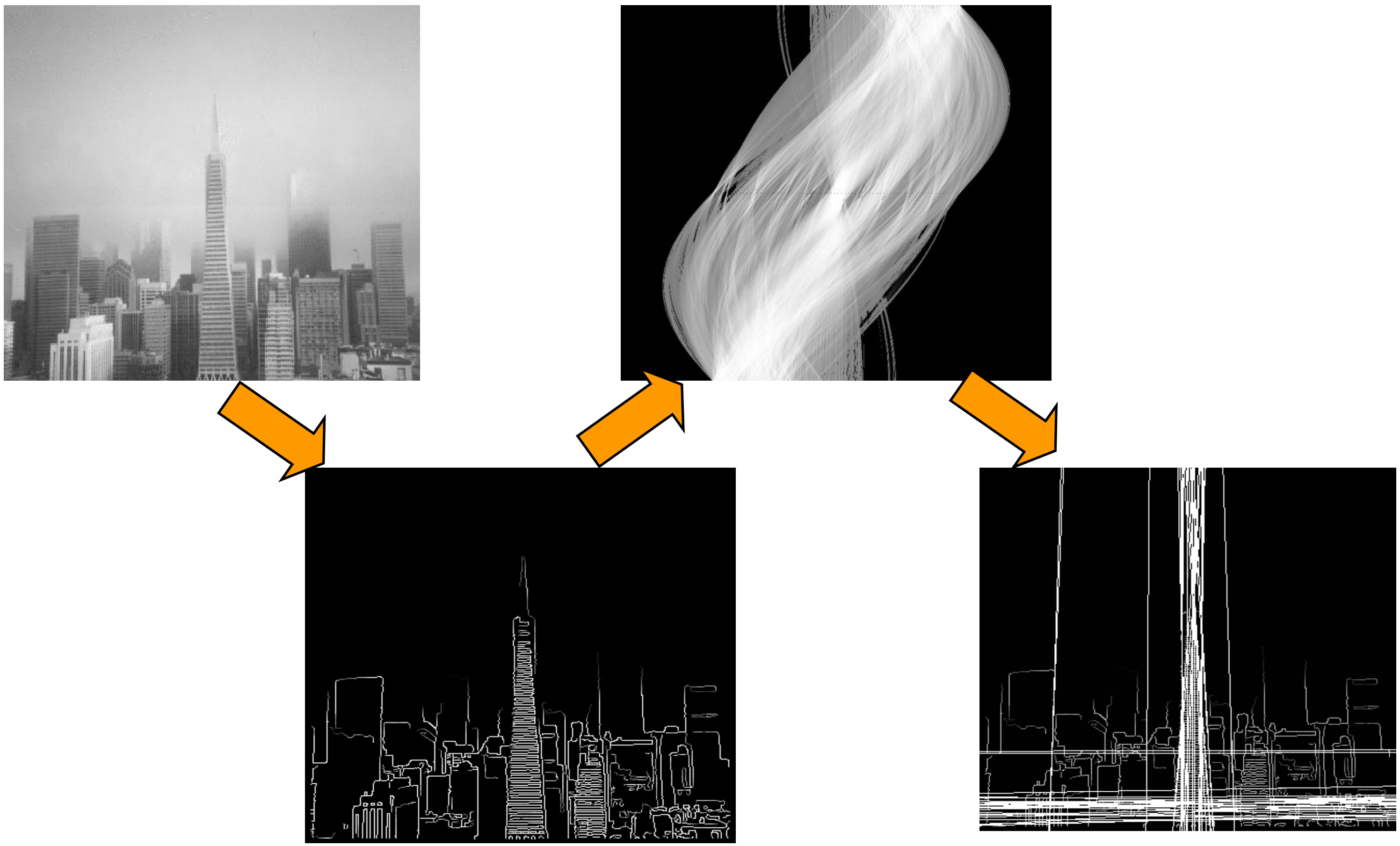
Input Image



Rendering of Transform Results



General Hough Process



Now that we have lines, let's do an analytic curve

An analytic curve: $f(\mathbf{a},\mathbf{x}) = 0$, $\mathbf{a}$ is parameter vector, $\mathbf{x}$ is pixel position

Let's let $\varphi(\mathbf{x})$ be the direction of the gradient.

Let's also say that changes around the pixel in the x direction are so small we don't care about them (i.e. $df/dx=0$) since we know the gradient we also know that $dy/dx = \tan(\varphi(\mathbf{x}) - \pi/2)$.

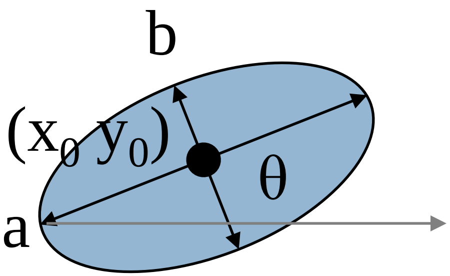
Our algorithm for finding the shape histogram in parameter is space becomes:

For each $\mathbf{x}$:

Find $\mathbf{a}$ such that, $f(\mathbf{a},\mathbf{x}) = 0$, and $df/dx=0$

Add 1 to $\mathbf{a}$'s entry in the accumulator ($\mathbf{A}[\mathbf{a}]$++).

So let's show an example for an analytic curve (the ellipse)



$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1.$$

procedure HoughEllipse (integer $X_{min}X_{max}$, $Y_{min}Y_{max}$, $\theta_{min}\theta_{max}$, $a_{min}a_{max}$, $b_{min}b_{max}$, x, y, x_0, y_0, dx, dy; real angle, ξ;
integer array A, P);
begin;
for $x := x_{min}$ step dx to x_{max} do
for $y := y_{min}$ step dy to y_{max} do
begin
$dX := P(x+\text{delta}, y) - P(x, y)$;
$dY := P(x, y+\text{delta}) - P(x, y)$;

For every edge element calculate get edge dx and dy

for $a := a_{min}$ step da until a_{max} do
for $b := b_{min}$ step db until b_{max} do
for $\theta := \theta_{min}$ step $d\theta$ until θ_{max} do

For every element in the parameter space (a,b,θ)

begin;
angle $:= \arctan\left(\frac{dY}{dX}\right) - \theta - \frac{\pi}{2}$;
$\xi := \tan(\text{angle})$;

Calculate gradient

$$dx := \text{Sign}\,X(dX, dY)\frac{a^2}{\sqrt{\left(1 + \frac{b^2}{a^2\xi^2}\right)}};$$

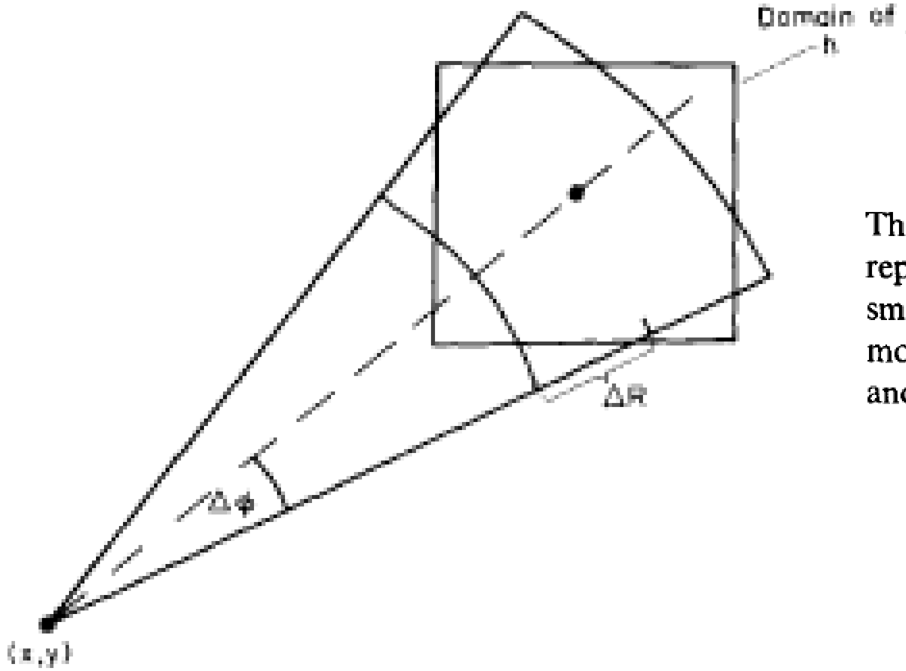
$$dy := \text{Sign}\,Y(dX, dY)\frac{b^2}{\sqrt{\left(1 + \frac{a^2}{b^2}\right)^2}};$$

Rotate-by-Theta(dx, dy);
$x_0 := x + dx$;
$y_0 := y + dy$;
$A(x_0, y_0, \theta, a, b) := A(x_0, y_0, \theta, a, b) + 1$;
end;
end.

Get x_0 and y_0 and update accumulator **A**[**a**]++

So what about sample noise?

- Ballard essentially shows that smoothing the accumulator is compensates for small changes in the parameter space.



Thus within the approximation of letting the square represent the shaded band shown in Fig. 3, the smoothing procedure is equivalent to an accommodation for uncertainties in the gradient direction and radius.

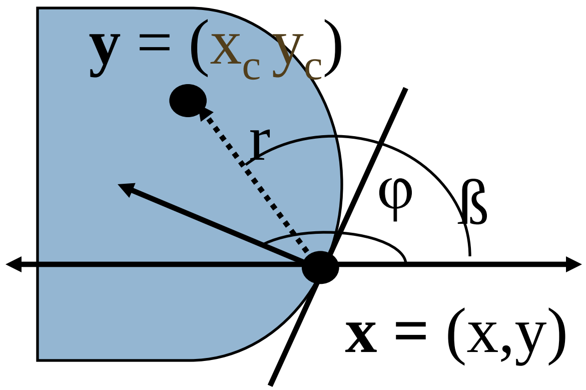
Speed of Hough

- But how long does this take? If we have m parameters, each of which has M values, we have M^{m-1} entries in our accumulator.

- However, if we have the gradient direction we can reduce the parameter space so it only takes $O(M^{m-2})$ for $m \geq 2$!

Now for a non-analytic curve (i.e. an arbitrary shape)

Let **a** = (**y**,s,θ) where **y** is a reference point, s is a scale factor, θ is a rotation factor.



$r = \sqrt{((x - x_c)^2 + (y - y_c)^2)}$

$\beta = \mathrm{atan}((y - y_c)/(x - x_c))$

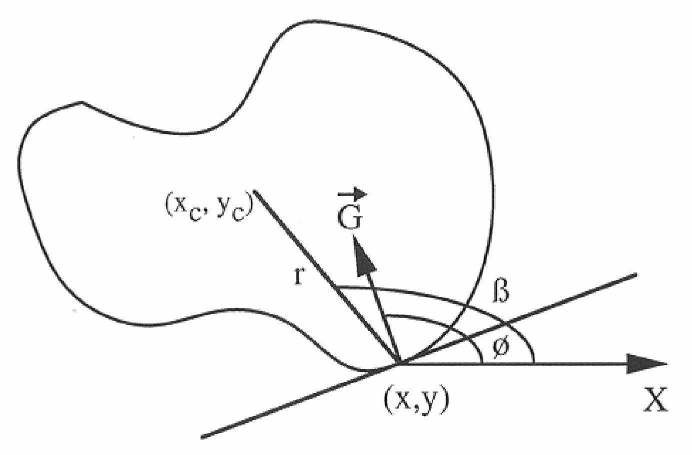
$x_c = x + r\cos(\beta)$

$y_c = y + r\sin(\beta)$

This is slightly different than Ballard but is more intuitive.

Computing R-Tables:
Associating edge gradient to reference point

Associate gradient direction with a set of parameters



$0\Delta\varphi = 0°$	(r, β)
$1\Delta\varphi = 10°$	(r, β)
$2\Delta\varphi = 20°$	(r, β) (r, β)
…	(r, β) (r, β)
$(k-1)\Delta\varphi = 170°$	(r, β) (r, β)
$k\Delta\varphi = 180°$	(r, β)

Generalized Hough Transform

Build the R-Table

Build a 4 dimensional accumulator over (x_c,y_c,s,θ) $(A[x_c][y_c][\theta][s])$

For each edge point (x,y) calculate the surface normal φ.

Lookup R table value for φ

For each pair of (r, β) in the R-Table at φ

  Convert (r, β) to potential (x_c,y_c)

  For each value of s' between s_{max} and s_{min} in A

  For each value of θ' between θ_{max} and θ_{min} in A

    Rotate and scale (x_c,y_c) to form $(x_c,y_c)'$

    Add one to $\mathbf{A}[x_c'][y_c'][\theta'][s'] = \mathbf{A}[\mathbf{a}]{+}{+}$

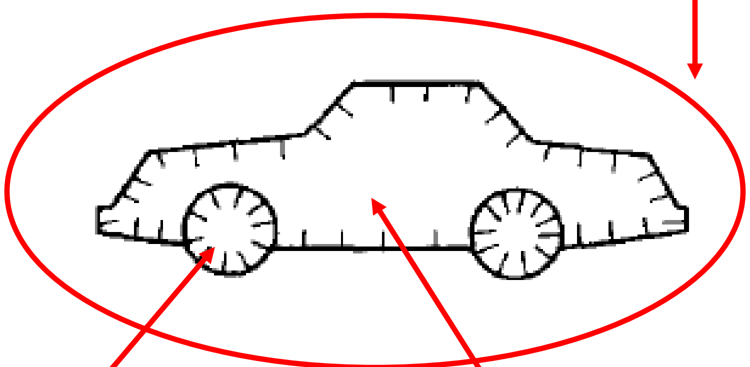
Local maxima in the accumulator represent the shape of the object!

Fun Facts about the Accumulator

It can detect shapes that have radial symmetry like donuts!

Hough Transform for Composite Shapes

Composite shape S with reference **y**



S_1 with $\mathbf{y}_1$

S_2 with $\mathbf{y}_2$

S is a shape

$R_s(\varphi)$ is the R table of shape S

y is the reference point

let $\mathbf{r}_1 = \mathbf{y} - \mathbf{y}_1$

let $\mathbf{r}_2 = \mathbf{y} - \mathbf{y}_2$

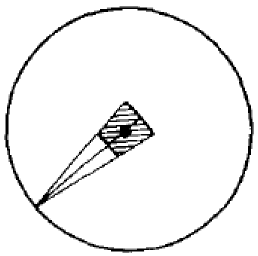
$$R_S(\phi) = [R_{S_1}(\phi) + \mathbf{r}_1] \dot{\cup} [R_{S_2}(\phi) + \mathbf{r}_2]$$

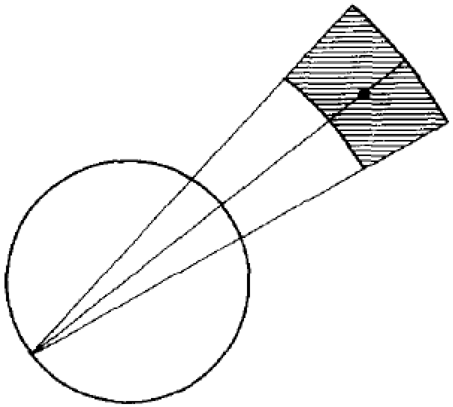
We can also define shape as the difference of R tables:

$$R_S = R_{S_1} \dot{-} R_{S_2}$$

When R Table approaches Break

Close Reference, Small Error





Far Reference, Large Error

Point is reference point

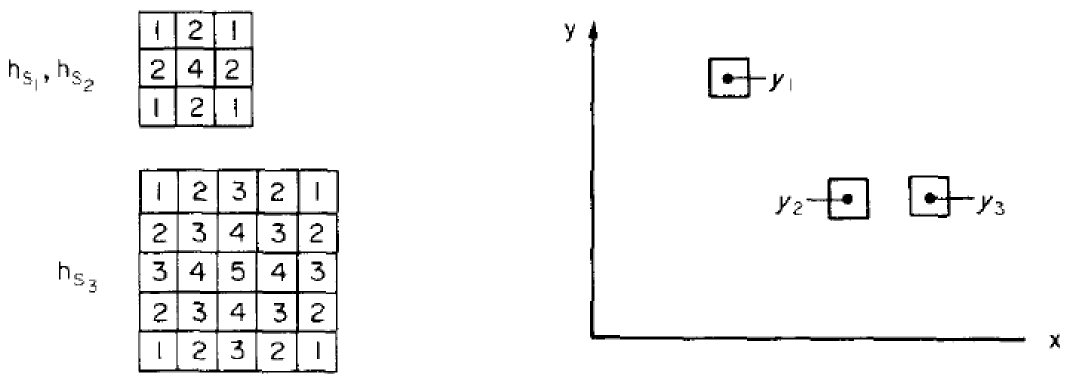
Shaded area is error.

Another approach: smoothing template

$H_i(\mathbf{y}_i)$= Smoothing template for a reference point

1. Construct smoothing template: $$H(y) = \sum_{i=1}^{N} h_i(\mathbf{y} - \mathbf{y}_i).$$
2. Make R table or each point, for each value of s, for each value of θ
$$R_s(\phi) = T_s\left\{T_\theta\left[\bigcup_{k=1}^{N} R_{S_k}(\phi)\right]\right\}.$$
3. Increment each entry in the parameter histogram A
4. Our new accumulator is $\mathbf{A}_s = \mathbf{A}*\mathbf{H}$

Creating the H matrix



Accumulator Addition Strategies

$\mathbf{A}[\mathbf{a}] = \mathbf{A}[\mathbf{a}] + c$

Our regular approach

$\mathbf{A}[\mathbf{a}] = \mathbf{A}[\mathbf{a}] + g(\mathbf{x})$

Where $g(\mathbf{x})$ is a function of the gradient magnitude.

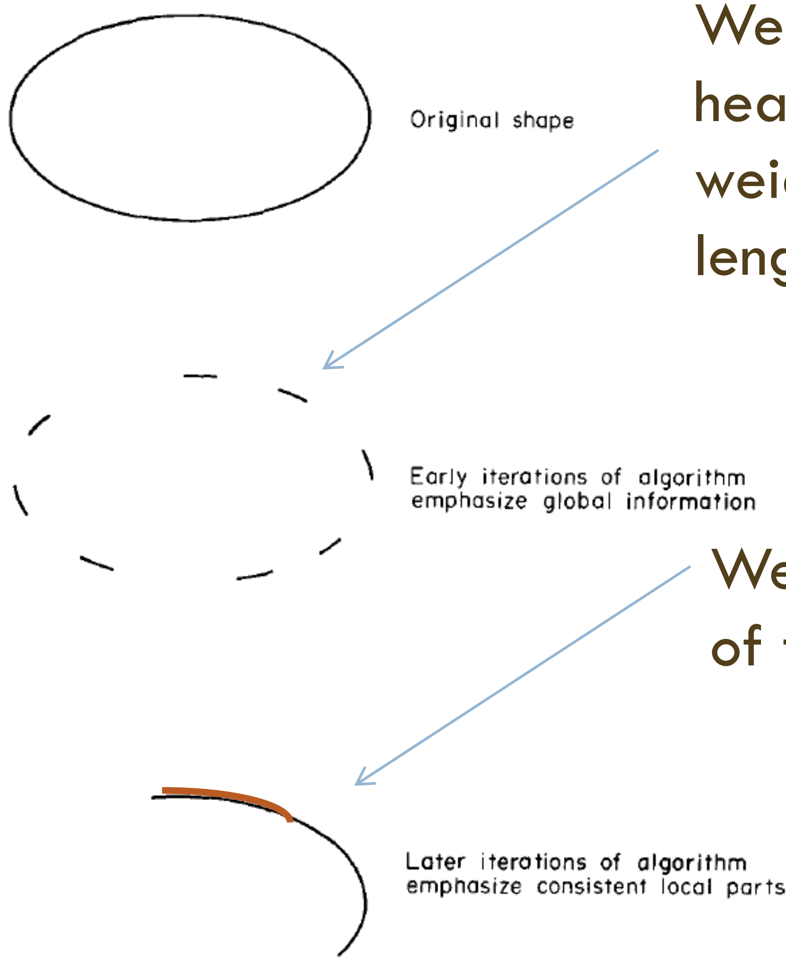
$\mathbf{A}[\mathbf{a}] = \mathbf{A}[\mathbf{a}] + g(\mathbf{x}) + c$

Similar to above

$\mathbf{A}[\mathbf{a}] = \mathbf{A}[\mathbf{a}] + \mathbf{K}$

Where K is a value for local curvature. (We have a locally good fit)

Improving Performance



Weight connected segments more heavily than single pixels this weight can be based on the length of the edge from a pixel.

Weight more interesting sections of the image