Chapter 7.1
Hough Transforms

Image Processing and Computer Vision

LE Thanh Sach
Faculty of Computer Science and Engineering
Ho Chi Minh University of Technology, VNU-HCM

Overview

1 What is it?

2 Analytic Shape

- Principle
- Straight Lines
- Circles
- General Curves

3 Non-Analytic Shape

- A Special Case
- Generalized Hough Transforms (GHT)
- GHT with Scaling and Rotation

What is it?

Hough transforms is a method for locating objects in input images.

Questions

Q1: How do we specify objects being located?

Q2: Which information in the input image does Hough Transforms need?

Q1: How do we specify objects being located?

The objects can be expressed by one of the followings.

1. **Analytic Form**: The objects are represented by mathematical relations, for examples,

$$\text{Straight line: } y = ax + b$$

$$\text{Circle: } (x-a)^2 + (y-b)^2 = r^2$$

$$\text{Ellipse: } \left(\frac{x-x_c}{a}\right)^2 + \left(\frac{y-y_c}{b}\right)^2 = 1$$

$$\text{General form: } f(\mathbf{x}, \mathbf{a}) = 0$$

- $\mathbf{x}, \mathbf{a}$: vector of variables and parameters respectively.

What is it?

2 **Non-Analytic Form**: The objects are represented by the location and the gradient of pixels on the objects' boundary.

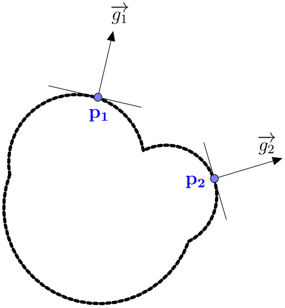

Figure 1: A shape represented by its boundary and gradients

What is it?

Q2: Which information in the input image does Hough Transforms need?

Hough Transforms needs:

1. Edge pixels (location information)
2. Gradient of edge pixels (directional information)

$\Rightarrow$ First-order derivatives can be applied to obtain the required information.

What is it?

A simple method for obtaining the required information from the input image:

Example

1. Differentiate input image $I(x, y)$ to obtain gradient image $I_g(x, y)$.
2. Find a threshold T, e.g., $T =$ percentile 90% of $|I_g(x, y)|$
3. Obtain edge map: $I_e(x, y) = |I_g(x, y)| > T$

Analytic Shape - Principle

Equation of straight lines:
Image Space: Treating a and b as constant parameters, x and y as variables

$$y = ax + b$$

Parameter Space: Treating x and y as constant parameters, a and b as variables

$$b = -xa + y$$

Analytic Shape - Principle

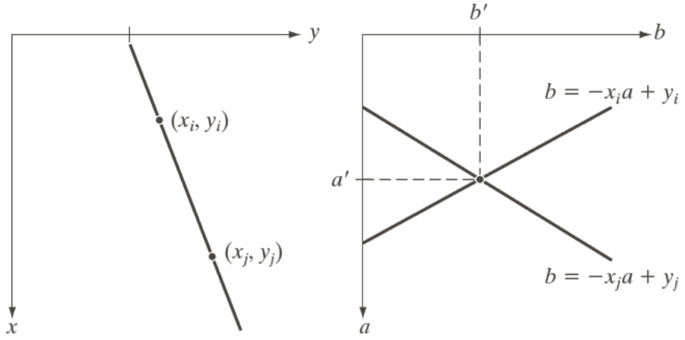

Figure 2: Left: **Image space**; Right: **Parameter space**

- **A point** in image space is corresponding to **a line** in parameter space.
- **A line** in image space is corresponding to **a point** in parameter space.

Analytic Shape - Principle

If we have N edge points on the line passing two points, called (x_i, y_i) and (x_j, y_j) (see Fig. 2), then :

- We have N lines in parameter space
- These N lines intersect at **a common point**: (a', b') in parameter space

$\Rightarrow$ Detect this **common point** in parameter space $\Rightarrow$ equation of line in image space: $y = a'x + b'$

Analytic Shape - Principle

So, the basic idea is:

1. **Discretize** paramter space into small cells. Each cell contains the number of lines passing it
    - The whole space now called the **Accumulator** $A(i, j)$
    - $i = 0, 1, .., M-1$; $i = 0, 1, .., N-1$
2. Find the common intersection point by finding the cell that contains the **largest number of lines** passing it. Assume that it is (a', b')

The equation found is: $y = a'x + b'$

Analytic Shape - Principle

Challenging Problems

1. **Accuracy**: The accurate estimation of parameter a and b depends on the resolution of the accumulator, i.e., the size of cells in the accumulator
   - $\Rightarrow$ Discretize parameter space into smaller cells.

2. **Memory cost**: The accumulator contains so many cells, especially, in the case that there are many parameters and that we use a high resolution accumulator.

3. **Large or unlimited range**: In some cases, parameters have large ranges, for example, a in $y = ax + b$

Detection of Straight Lines

Analytic Shape - Detection of Straight lines

Questions

1. In **line detection**, Parameter a, in $y = ax + b$, has an infinite range. How do we solve this problem?

Equation of straight lines:

$$y = ax + b$$

Vertical line: $a \rightarrow \infty$
$\Rightarrow$ use the following form

Analytic Shape - Detection of Straight lines

$$\mathbf{x}\cos(\theta) + \mathbf{y}\sin(\theta) = \rho$$

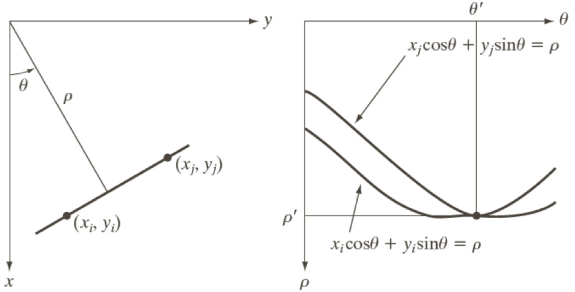

Figure 3: Hough transforms: (a) Image space, (b) Parameter space

Analytic Shape - Detection of Straight lines

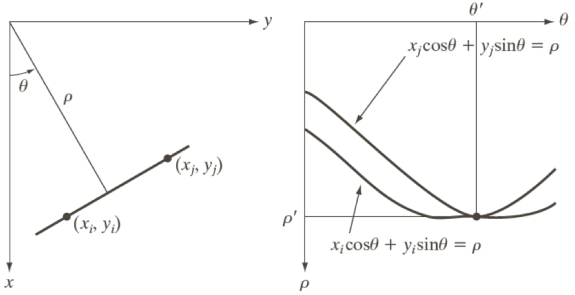

Figure 4: Hough transforms: (a) Image space, (b) Parameter space

- A fixed point in image space ⇔ A curve in parameter space
- A line in image space ⇔ A fixed point in parameter space

Analytic Shape - Detection of Straight lines

Advantages of the expression with (θ, ρ)

Range of θ and ρ is **limited**.

- $-\pi \leq \theta \leq \pi$
- $-D \leq \rho \leq D$

D: The maximum distance between two corners in images. Image's size: $R \times C$, then

$$D = \sqrt{R^2 + C^2}$$

Analytic Shape - Detection of Straight lines

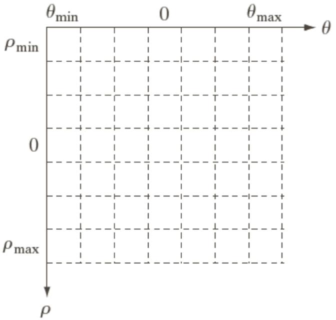

Figure 5: Demonstration of the discretization into $M \times N$ cells

Questions

2. After discretization, how does cell's indices (i, j) relate to parameter θ and ρ?

Analytic Shape - Detection of Straight lines

Along ρ-direction:

- $D = \sqrt{R^2 + C^2}$
- $\rho_{min} = -D; \rho_{max} = D$: Left and right bound of the range
- $L_\rho = 2D$: range's width
- M : number of rows along ρ axis
- $i = 0, 1, .., M - 1$: the index of cells along ρ axis
- $\Rightarrow$ Quantization step along ρ axis: $\Delta_\rho = L_\rho / M$

Analytic Shape - Detection of Straight lines

Along ρ-direction:

- $\Rightarrow$ From cell's index to ρ (at the center of the cell):

$$\rho \triangleq \text{dcm}_{idx2\rho}(i) = \rho_{min} + i \times \Delta_\rho + \frac{\Delta_\rho}{2}$$

- $\Rightarrow$ From ρ to cell's index

$$i \triangleq \text{cdm}_{\rho 2idx}(\rho) = \text{round}\left(\frac{\rho - \rho_{min}}{\Delta_\rho}\right)$$

1. cdm: **c**ontinuous to **d**iscrete **m**apping
2. dcm: **d**iscrete to **c**ontinuous **m**apping

Analytic Shape - Detection of Straight lines

Along θ-direction:

- $\theta_{min} = -\pi/2; \rho_{max} = \pi/2$: Left and right bound of the range
- $L_\theta = \pi$: Range's width
- N : number of columns along θ axis
- $j = 0, 1, .., N-1$: the index of cells along θ axis
- $\Rightarrow$ Quantization step along θ axis: $\Delta_\theta = L_\theta / N = \pi / N$

Analytic Shape - Detection of Straight lines

Along θ-direction:

- $\Rightarrow$ From cell's index to θ (at the center of the cell):

$$\theta \stackrel{\triangle}{=} \text{dcm}_{idx2\theta}(j) = \theta_{min} + j \times \Delta_\theta + \frac{\Delta_\theta}{2}$$

- $\Rightarrow$ From θ to cell's index

$$j \stackrel{\triangle}{=} \text{cdm}_{\theta 2idx}(\theta) = \text{round}\left(\frac{\theta - \theta_{min}}{\Delta_\theta}\right)$$

Analytic Shape - Detection of Straight lines

Questions

3. How can we detect a straight line with Hough Transforms?

Analytic Shape - Detection of Straight lines

Algorithm: An Informal representation

Algorithm 1 Hough Line Detection - PART1

Create an accumulator, referred to as A
Set 0 for all cells in the accumulator
for all edge point in $I_e(x, y)$ **do**
    for all $j \in [0, N-1]$ **do** ▷ iterate on each cell along θ-direction
        $\theta = \text{dcm}_{idx2\theta}(j)$
        $\rho = x\cos(\theta) + y\sin(\theta)$
        $i = \text{cdm}_{\rho 2idx}(\rho)$
        $A(i, j) = A(i, j) + \Delta(x, y)$
    end for
end for

Analytic Shape - Detection of Straight lines

Algorithm: An Informal representation

Algorithm 2 Hough Line Detection - PART2

11: Find the largest value in the accumulator, assume at (s, t)
12: $\rho^* = \text{dcm}_{idx2\rho}(s)$
13: $\theta^* = \text{dcm}_{idx2\theta}(t)$

The detected line has following equation:

$$\mathbf{xcos}(\theta^*) + \mathbf{ysin}(\theta^*) = \rho^*$$

Analytic Shape - Detection of Straight lines

What is meaning of $\Delta(x, y)$?

1. $\Delta(x, y) = 1$ for any edge point
   - Accumulator A (with normalization) shows the probability of having a lines at each pair of (ρ, θ)

2. $\Delta(x, y) = |\overrightarrow{g}(x, y)|$, where $\overrightarrow{g}(x, y)$ is the gradient vector at edge point $I_e(x, y)$
   - Accumulator A shows the strengthen of the dis-continued information (edge) along pixels on the straight line with parameter (ρ, θ)

3. $\Delta(x, y) = |\overrightarrow{g}(x, y)| + c$, where c is a constant.
   - A variation from the previous

Analytic Shape - Detection of Straight lines

Exercise

1. Implement line detection with **Matlab** and **C/C++**
2. Assume that $\phi(x, y)$ is the angle of the gradient vector at $I_e(x, y)$ and that the estimation error of the gradient's angle is $[-\Delta_\phi, +\Delta_\phi]$. How does $\phi(x, y)$ relate to parameter θ?
3. Using on $\phi(x, y)$ and $[-\Delta_\phi, +\Delta_\phi]$, which cells in A should be increased for each ρ?

Analytic Shape - Detection of Straight lines

Questions

5. **How can we detect K straight lines with Hough Transforms in the input image?**

Analytic Shape - Detection of Straight lines

Questions

5. **How can we detect K straight lines with Hough Transforms in the input image?**

Guideline

- Create a accumulator, same as detecting 1 straight line.
- Use **non-maxima suppression** to remove (suppress) non-maxima cells.
- Find K **largest local maxima** by using **max-heap**

Analytic Shape - Detection of Straight lines

Algorithm 3 Hough Line Detection - PSEUDO-CODE

```
function DETECT_LINE(
        REF I_e(x, y): edge map,
        R, C: num of rows and cols of the edge map,
        M, N: num of rows and cols the accumulator A(i, j),
        REF K: num of straight lines,
        REF R_θ, R_ρ: array of θ and ρ detected )

    Create Accumulator A with size M × N
    COMP_ACCUMULATOR(I_e, R, C, A, M, N);
    APPLY_NONMAXIMA_SUPPRESSION(A, M, N)
    FIND_MAXIMA(A, M, N, R_θ, R_ρ, K)
end function
```

Analytic Shape - Detection of Straight lines

Algorithm 4 Updating the Accumulator, PART 1

```
1: function COMP_ACCUMULATOR(
       REF I(x, y): edge map,
       R, C: num of rows and cols of the edge map,
       REF A : Accumulator
       M, N: num of rows and cols the accumulator A )

2:     for r=0 to M-1 do
3:         for c=0 to N-1 do
4:             A(r, c) = 0;            ▷ Initialize the accumulator
5:         end for
6:     end for
```

Analytic Shape - Detection of Straight lines

Algorithm 5 Updating the Accumulator, PART 2

$\Delta_\theta = L_\theta / N$
$\Delta_\rho = L_\rho / M$
$\rho_{min} = -D$; $\theta_{min} = -\pi/2$
for $x = 0$ to $R - 1$ **do**
  for $y = 0$ to $C - 1$ **do**
    if $I_e(x, y) \neq 0$ **then** ▷ (x, y): edge point
      for $j = 0$ to $M - 1$ **do**
        $\theta = \text{dcm}_{idx2\theta}(j)$
        $\rho = x\cos(\theta) + y\sin(\theta)$
        $i = \text{cdm}_{\rho 2idx}(\rho)$
        $A(i, j) = A(i, j) + \Delta(x, y)$ ▷ Voting
      end for
    end if
  end for
end for
end function

Analytic Shape - Detection of Straight lines

Algorithm 6 Removing Non-maxima

```
function APPLY_NONMAXIMA_SUPPRESSION(
        REF A : Accumulator
        M, N: num of rows and cols the accumulator A )

    for all cell (i, j), accept the border do
        NW = A(i − 1, j − 1)
        N = A(i − 1, j)
        NE = A(i − 1, j + 1)
        E = A(i, j + 1)
        SE = A(i + 1, j + 1)
        S = A(i + 1, j)
        SW = A(i + 1, j − 1)
        W = A(i, j − 1)
        C = A(i, j)
```

Analytic Shape - Detection of Straight lines

Algorithm 7 Non-maxima Suppression

```
        C1 = (C < NW) OR (C < N)
        C2 = (C < NE) OR (C < E)
        C3 = (C < SE) OR (C < S)
        C4 = (C < SW) OR (C < W)
        if (C1 OR C2 OR C3 OR C4) then
            A(i, j) = 0                ▷ suppress non-maxima
        end if
    end for
end function
```

Analytic Shape - Detection of Straight lines

Algorithm 8 Finding K Maxima, PART1

```
function FIND_MAXIMA(
        REF A : Accumulator
        M, N: num of rows and cols the accumulator A
        REF R_θ, R_ρ: array of θ and ρ detected
        REF K: num of straight lines )

    Create an empty max-heap, referred to as H_max
    for all cell (i, j) in A, accept the border do
        if A(i, j) ≠ 0 then                          ▷ an extreme
            key = A(i, j)
            data.rho = dcm_idx2ρ(i)
            data.theta = dcm_idx2θ(j)
            E = { key, data.rho, data.theta }
            Add E to H_max                           ▷ re-heap up
        end if
    end for
```

Analytic Shape - Detection of Straight lines

Algorithm 9 Finding K Maxima, PART2

```
    k = 0
    while (H_max is not empty) AND (k < K) do
        E = Remove maximum element from H_max
        R_ρ[k] = E.data.rho
        R_θ[k] = E.data.theta
        k = k + 1                          ▷ next maximum
    end while
    K = k;                    ▷ Update num of lines found
end function
```

Detection of Circles

Analytic Shape - Detection of Circles

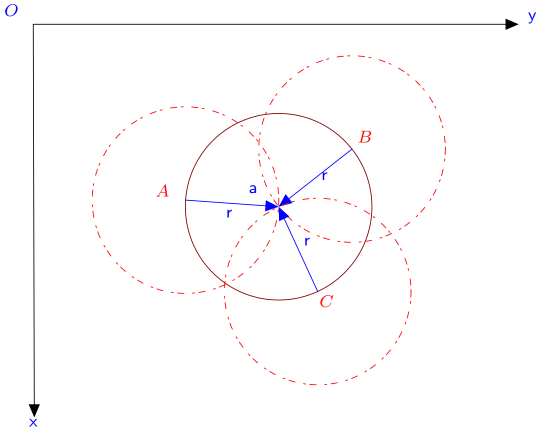

Figure 6: A circle centered at **a**, radius **r**, contains three points **A**, **B**, and **C**. We need to detect this circle.

Analytic Shape - Detection of Circles

Assumption: We DO NOT know where is the center **a**, but we know radius **r** in advance.

Facts

- **A** is on circle (**a**, **r**) $\Rightarrow$ **a** is on the circle centered at **A**, radius **r**. See Fig. 6

Analytic Shape - Detection of Circles

Assumption: We DO NOT know where is the center **a**, but we know radius **r** in advance.

Facts

- **A** is on circle (**a**, **r**) $\Rightarrow$ **a** is on the circle centered at **A**, radius **r**. See Fig. 6
- **B** is on circle (**a**, **r**) $\Rightarrow$ **a** is on the circle centered at **B**, radius **r**. See Fig. 6

Analytic Shape - Detection of Circles

Assumption: We DO NOT know where is the center **a**, but we know radius **r** in advance.

Facts

- **A** is on circle (**a**, **r**) $\Rightarrow$ **a** is on the circle centered at **A**, radius **r**. See Fig. 6
- **B** is on circle (**a**, **r**) $\Rightarrow$ **a** is on the circle centered at **B**, radius **r**. See Fig. 6
- **C** is on circle (**a**, **r**) $\Rightarrow$ **a** is on the circle centered at **C**, radius **r**. See Fig. 6
- **Center a** is the intersection of the three circles. See Fig. 6

We can use the **voting-technique**, as used in line detection, to solve the detection

Analytic Shape - Detection of Circles

Circle equation:

Explicit form

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

- Circle: has three parameters, x_c, y_c, and r
- $\Rightarrow$ Accumulator A is an array of 3 dimensions, indexed by x_c, y_c, and r
- $\Rightarrow$ A is a function of x_c, y_c, and r

Analytic Shape - Detection of Circles

Circle equation:

Explicit form

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

- Range of $x_c : 0, 1, .., R - 1$
- Range of $y_c : 0, 1, .., C - 1$
- Range of $r : 0, 2, .., R_{max} = \frac{\min(R,C)}{2}$
- Given an edge point (x_i, y_i) in image space:
  - For all points (x_c, y_c) in parameter space, compute dependent parameter r as follows:

$$r = \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2}$$

Analytic Shape - Detection of Circles

Algorithm 10 Hough Circle Detection, Using explicit form

```
Create a 3D-Accumulator, referred to as A
Set 0 for all cells in the accumulator
for all edge point in I_e(x, y) do
    for x_c = 0 to R − 1 do
        for y_c = 0 to C − 1 do
            Compute r = sqrt((x_c − x)^2 + (y_c − y)^2)
            A(x_c, y_c, r) = A(x_c, y_c, r) + 1
        end for
    end for
end for
Find the largest cell in A(x_c, y_c, r), assume at (x_c*, y_c*, r*)
```

The detected circle has following equation:
$(x - x_c^*)^2 + (y - y_c^*)^2 = r^{*2}$

Analytic Shape - Detection of Circles

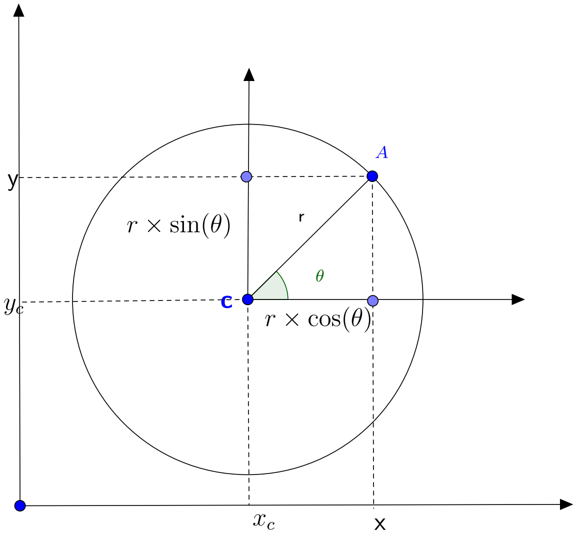

Figure 7: Parametric form: θ varies from 0 to $2\pi \rightarrow$ **A** draws a circle

Analytic Shape - Detection of Circles

Circle equation:

Parametric form

$$x = x_c + r\cos(\theta)$$
$$y = y_c + r\sin(\theta)$$

- θ is **not a free** parameter
- Range of θ: $\theta \in [0, 2\pi]$

Advantages of parametric form

Solve free parameters easily, for examples,

$$x_c = x - r\cos(\theta)$$
$$y_c = y - r\sin(\theta)$$

Analytic Shape - Detection of Circles

Algorithm: An Informal representation

Algorithm 11 Hough Circle Detection - PART1

Create a 3D-Accumulator, named A
Set 0 for all cells in the accumulator
for all edge point in $I_e(x, y)$ **do**
  for all $\theta_i \in [0, 2\pi]$ **do** ▷ Discretization of $[0, 2\pi] \rightarrow \theta_i$
    for all $r \in [r_{min}, r_{max}]$ **do**
      $x_c = x - r\cos(\theta)$
      $y_c = y - r\sin(\theta)$
      $i, j, k \leftarrow x_c, y_c$ and r respectively.
      $A(i, j, k) = A(i, j, k) + \Delta(x, y)$
    end for
  end for
end for

Analytic Shape - Detection of Circles

Algorithm: An Informal representation

Algorithm 12 Hough Circle Detection - PART2

13: Find the largest cell in $A(i, j, k)$, assume at (i^*, j^*, k^*)

14: Determine x_c^*, y_c^* and r^* from (i^*, j^*, k^*)

The detected circle has following equation:

$$x = x_c^* + r^* \times \cos(\theta)$$
$$y = y_c^* + r^* \times \sin(\theta)$$
$$\theta \in [0, 2\pi]$$

Analytic Shape - Detection of Circles

Questions

7. **How can we speed up the circle detection using gradient vectors at edge points?**

Analytic Shape - Detection of Circles

Questions

7. **How can we speed up the circle detection using gradient vectors at edge points?**

Facts

1. **The direction of the gradient vector** at every point (x, y) on a circle **passes through the center** of that circle.
2. Angle θ_i on Line 4 in Algorithm 11 and the angle of gradient vector at edge point $I_e(x, y)$ on Line 3 must be coincided. See angle θ in Figure 8

Analytic Shape - Detection of Circles

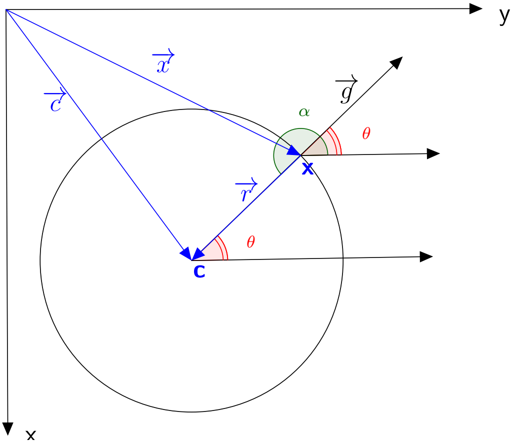

Figure 8: Circle detection: relation between gradient vectors and the center

Analytic Shape - Detection of Circles

Using gradient vectors

- θ_i in Line 4 (Algorithm 11) is equal to the angle of the gradient vector at edge point (x, y) on the circle.
- Let $\phi(x, y)$ be gradient angle at edge point (x, y)
- Let $\Delta\phi$ be the maximum estimation error for gradient angle.
- Range of anticipated gradient angle: $R_{grad} = [\phi(x, y) - \Delta\phi, \phi(x, y) + \Delta\phi]$
- So, Line 4 in previous algorithm will changed to

for all $\theta_i \in [\phi(x, y) - \Delta\phi, \phi(x, y) + \Delta\phi]$ **do**

Detection of General Curve

Analytic Shape - Detection of General Curve

General form of curves:

$$f(\mathbf{x}, \mathbf{a}) = 0$$

Where,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix}; \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \\ a_m \end{bmatrix}$$

- n : n variables
- m : m parameters

Analytic Shape - Detection of General Curve

General form of circles:

$$f(\mathbf{x}, \mathbf{a}) = 0$$
$$\equiv (x - x_c)^2 + (y - y_c)^2 - r^2 = 0$$

Where,

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} ; \mathbf{a} = \begin{bmatrix} x_c \\ y_c \\ r \end{bmatrix}$$

- $n = 2$
- $m = 3$

Analytic Shape - Detection of General Curve

Algorithm: An Informal representation

Algorithm 13 Hough Curve Detection

Create accumulator $A \equiv$ array of m-dimensions
Initialize A with 0 for all cells
for all edge point $\mathbf{x_i}$ **do**
    for all cell a_j **do**
        if $f(\mathbf{x_i}, \mathbf{a_j}) == 0$ **then**
            $A(\mathbf{a_j}) = A(\mathbf{a_j}) + \Delta(x, y)$
        end if
    end for
end for
Find the largest cell in A, referred to as $\mathbf{a}^*$
return $\mathbf{a}^*$

The detected curve: $f(\mathbf{x}, \mathbf{a}^*) = 0$

Detection of Non-Analytic Shapes

Special Case: Detection of Circles

Important Questions

1. How does the gradient direction of a circle's edge point relate to the location of the circle's center?
2. How can we generalize such the relationship for more general shapes?
3. How can we utilize the generalized relationship to detect a shape described by the shape's edge point?

Non-Analytic Shape - Detection of Circles

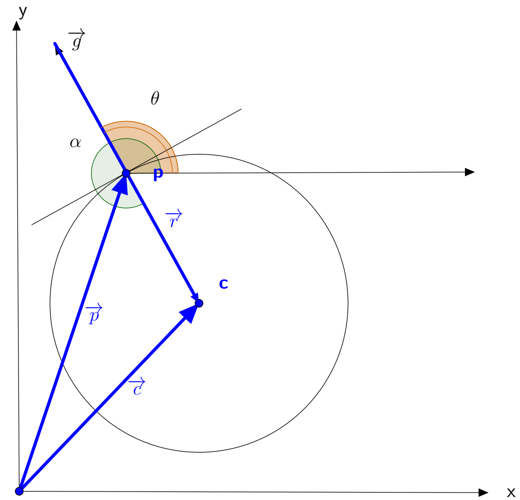

Figure 9: Circle: Center **c**, An edge point **p**, Gradient vector at **p**: $\overrightarrow{g}$, Angle of $\overrightarrow{g}$: θ

Non-Analytic Shape - Detection of Circles

What we know

1. $\mathbf{p}$ or $\overrightarrow{p}$: is an edge point. Vector form of $\mathbf{p}$ is $\overrightarrow{p}$

$$\overrightarrow{p} = \begin{bmatrix} x \\ y \end{bmatrix}$$

2. θ: angle of gradient vector
3. $|\overrightarrow{r}|$: radius of the circle being detected.
   - $\overrightarrow{r}$ is the vector from $\mathbf{p}$ to the center $\mathbf{c}$ (not known now)
   - We just know the magnitude of $\overrightarrow{r}$

Non-Analytic Shape - Detection of Circles

What we can infer

1. Angle of $\overrightarrow{r}$: $\alpha = \theta + \pi$
2. Vector $\overrightarrow{r}$:

$$\overrightarrow{r} = \begin{bmatrix} r\cos(\alpha) \\ r\sin(\alpha) \end{bmatrix} = \begin{bmatrix} r\cos(\theta + \pi) \\ r\sin(\theta + \pi) \end{bmatrix} = \begin{bmatrix} -r\cos(\theta) \\ -r\sin(\theta) \end{bmatrix}$$

Non-Analytic Shape - Detection of Circles

Finally, the location of the center can be computed by:

$$\overrightarrow{\mathbf{c}} = \overrightarrow{\mathbf{p}} + \overrightarrow{\mathbf{r}}$$

- Whenever we have $\overrightarrow{\mathbf{r}}$, we know where the circle is.

Non-Analytic Shape - Detection of Circles

Important Questions

1. How does the gradient direction of a circle's edge point relate to the location of the circle's center?

Solution:

$$\overrightarrow{r} = \begin{bmatrix} -r\cos(\theta) \\ -r\sin(\theta) \end{bmatrix}$$

- $\overrightarrow{r}$ depends on the angle of gradient vector.
- $\overrightarrow{r}$ **is a function of the angle of gradient vector**.

Non-Analytic Shape - Detection of Circles

Important Questions

1. How does the gradient direction of a circle's edge point relate to the location of the circle's center?

Solution:

$$\begin{aligned}\overrightarrow{c} &= \overrightarrow{p} + \overrightarrow{r} \\ &= \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -r\cos(\theta) \\ -r\sin(\theta) \end{bmatrix} \\ &= \begin{bmatrix} x - r\cos(\theta) \\ y - r\sin(\theta) \end{bmatrix}\end{aligned}$$

Non-Analytic Shape - Generalized Hough Transforms (GHT)

Important Questions

2. How can we generalize such the relationship for more general shapes?

Non-Analytic Shape - Generalized Hough Transforms

SOLUTION: From circle to more general shapes

CIRCLE:

- $|\overrightarrow{r}|$ is the same for every gradient vectors

MORE GENERAL SHAPES:

- $|\overrightarrow{r}|$ varies with the angle of gradient vector.

Non-Analytic Shape - Generalized Hough Transforms

SOLUTION: From circle to more general shapes

CIRCLE:

- $|\overrightarrow{r}|$ is the same for every gradient vectors
- Angle α of $\overrightarrow{r}$ is always $(\theta + \pi)$.

MORE GENERAL SHAPES:

- $|\overrightarrow{r}|$ varies with the angle of gradient vector.
- Angle α of $\overrightarrow{r}$ varies with the angle of gradient vector.

Non-Analytic Shape - Generalized Hough Transforms

SOLUTION: From circle to more general shapes

CIRCLE:

- $|\overrightarrow{r}|$ is the same for every gradient vectors
- Angle α of $\overrightarrow{r}$ is always $(\theta + \pi)$.

MORE GENERAL SHAPES:

- $|\overrightarrow{r}|$ varies with the angle of gradient vector.
- Angle α of $\overrightarrow{r}$ varies with the angle of gradient vector.
- One θ can associated with more than one $\overrightarrow{r}$

Non-Analytic Shape - Generalized Hough Transforms

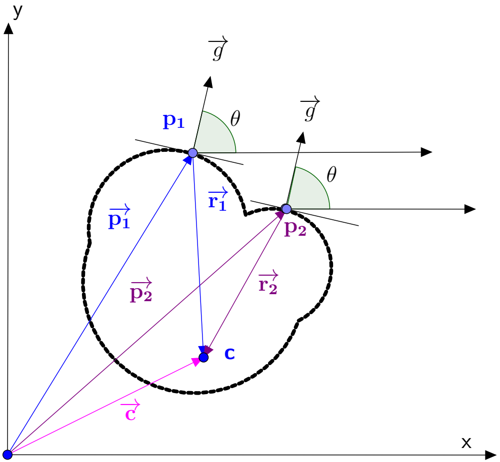

Figure 10: Generalized Shape: An angle θ can be associated with more than one vector $\overrightarrow{r}$

Non-Analytic Shape - Generalized Hough Transforms

Important Questions

3. How can we utilize the generalized relationship to detect a shape described by the shape's edge point?

Non-Analytic Shape - Generalized Hough Transforms

Input

1. An sample shape, S, described edge points on the shape boundary.
2. An image contains shape S

Method for detecting generalized shapes

1. **PHASE 1:** Describe the relationship between the gradient direction of edge points on S and a chosen point (referred to as **reference point**) $\mathbf{c}$ inside of S.
2. **PHASE 2:** Detect instances of S in the input image.

Non-Analytic Shape - Generalized Hough Transforms

PHASE 1: Description of $\theta \rightarrow \overrightarrow{r}$

1. Chose a point $\mathbf{c}$ inside of input shape S.
   - This point will be considered as the center of the shape, like center of a circle.
   - The purpose of **PHASE 2** is to detect $\mathbf{c}$
2. Build an loop-up table that maps θ (angle of gradient vectors) to $\overrightarrow{r}$
   - Name of this mapping: **R-TABLE**
   - One $\theta \rightarrow$ multiple $\overrightarrow{r}$

Non-Analytic Shape - Generalized Hough Transforms

Table 1: R-Table illustration, $R(\theta)$

i	θ_i	Content of the entry in **R-Table**
0	θ_0	$\overrightarrow{r}_{0,1}$; $\overrightarrow{r}_{0,2}$; ..; $\overrightarrow{r}_{0,N_0}$
1	θ_1	$\overrightarrow{r}_{1,1}$; $\overrightarrow{r}_{1,2}$; ..; $\overrightarrow{r}_{1,N_1}$
2 ...	θ_2 ...	$\overrightarrow{r}_{2,1}$; $\overrightarrow{r}_{2,2}$; ..; $\overrightarrow{r}_{2,N_2}$ ...
$M-1$	θ_{M-1}	$\overrightarrow{r}_{M-1,1}$; $\overrightarrow{r}_{M-1,2}$; ..; $\overrightarrow{r}_{M-1,N_{M-1}}$

1. Number of entries: M
2. θ_0 : smallest angle of gradient vectors $(-\pi)$
3. θ_{M-1} : largest angle of gradient vectors $(+\pi)$
4. N_i: number of vector $\overrightarrow{r}$ associated with θ_i
   - N_i: maybe a zero, maybe more than 1

Non-Analytic Shape - Generalized Hough Transforms

Discretization of θ

- $\theta_{min} = -\pi$
- $\theta_{max} = +\pi$
- Range of θ : $L_\theta = 2\pi$
- Number of table entries: M
- $\Rightarrow \Delta_\theta = \frac{2\pi}{M}$

Non-Analytic Shape - Generalized Hough Transforms

From i, index of R-TABLE's rows, to θ :

$$\begin{aligned}\theta &\triangleq \text{dcm}_{idx2\theta}(i) \\ &= \theta_{min} + i \times \Delta_\theta + \frac{\Delta_\theta}{2} \\ &= -\pi + i \times \Delta_\theta + \frac{\Delta_\theta}{2}\end{aligned}$$

From θ to i, index of R-TABLE's rows:

$$\begin{aligned}i &\triangleq \text{cdm}_{\theta 2idx}(\theta) \\ &= \text{round}\left(\frac{\theta - \theta_{min}}{\Delta_\theta}\right) \\ &= \text{round}\left(\frac{\theta + \pi}{\Delta_\theta}\right)\end{aligned}$$

Non-Analytic Shape - Generalized Hough Transforms

Input:

- $\mathbf{c}$: a chosen point in previous step.

Algorithm 14 Generalized Hough Transforms: Building R-Table

1: Create **R-Table** R of M rows
2: **for all** edge point p on shape S **do**
3: Compute vector $\overrightarrow{r} = \overrightarrow{c} - \overrightarrow{p}$
4: Compute gradient vector $\overrightarrow{g}$ at p
5: Compute angle θ of $\overrightarrow{g}$
6: Determine row $i = \text{cdm}_{\theta 2idx}(\theta)$
7: Add $\overrightarrow{r}$ to Row i of R
8: **end for**

Non-Analytic Shape - Generalized Hough Transforms

PHASE 2: Detecting instances of S

1. Create 2D-Accumulator A for each possible of center $\mathbf{c}(x_c, y_c)$
2. Detect the largest cell in A

Non-Analytic Shape - Generalized Hough Transforms

Algorithm 15 Detection of instances of S

Create a 2D-Accumulator, referred to as A
Set 0 for all cells in the accumulator
for all edge point in $I_e(x, y)$ **do**
    Create vector $p = [x, -y]^T$ ▷ negative y, because y-axis: upright, x-axis: to-right
    Compute angle θ of the gradient at $I_e(x, y)$
    Determine R-TABLE's row: $l = \text{cdm}_{\theta 2idx}(\theta)$
    Get List L of vector $\overrightarrow{r}$ from Row l
    for all vector $\overrightarrow{r}_i$ in L **do**
        Compute vector $\overrightarrow{c} = \overrightarrow{p} + \overrightarrow{r}_i$
        Determine corresponding cell (x_c, y_c) in A from $\overrightarrow{c}$
        $A(x_c, y_c) = A(x_c, y_c) + \Delta(x, y)$
    end for
end for
Find the largest cell in $A(x_c, y_c)$, assume at (x_c^*, y_c^*)

GHT with Scaling and Rotation

Question

How can detect a shape in that case described as follows?

Input:

- A sample of a shape S specified by the shape's edge points.
- An input image $I(x, y)$

Capability of the Detection:

- is able to detect instances S_i in the input $I(x, y)$ in the case that S_i is a rotated and/or scaled version of S by an angle α and a scaling factor s?

GHT with Scaling and Rotation

GHT without Scaling and Rotation

R-Table: $R(\theta)$ i is a multivalued vector function

- Input: θ, for example, $\theta = \theta_i$, See Table 1
- Output: zero or multiple vectors: $\overrightarrow{r}_{i,1}$; $\overrightarrow{r}_{i,2}$; ...; $\overrightarrow{r}_{i,N_i}$

Accumulator : $A(x_c, y_c)$ is a **2D-array**, indexed by the coordinates of the reference point c

GHT with Scaling and Rotation

What do Scaling and Rotation affect the shape S?

SCALING:

- Causes vector $\overrightarrow{r}$ in R-Table scaled

ROTATION:

1. Causes angle of gradient rotated an angle α
2. Causes vector $\overrightarrow{r}$ in R-Table rotated an angle α

ACCUMULATOR A:

1. Need two more parameters: rotation angle α and scaling factor s
2. $\Rightarrow A(x_c, y_c, \alpha, s)$

R-TABLE $R(\theta)$:

1. Rebuilding is NOT required

GHT with Scaling and Rotation

SCALING:

Scaling-matrix:

$$M_s = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

Scaled vector $\overrightarrow{r}^s$ of $\overrightarrow{r}$:

$$\overrightarrow{r}^s = M_s \times \overrightarrow{r}$$

Scaling and Accumulator

- Perform the scaling for all vectors in R-Table.
- Increase A for each scaling factor

GHT with Scaling and Rotation

ROTATION:

Rotation-matrix for rotation angle α:

$$M_r = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

Rotated vector $\overrightarrow{r}^{rot}$ of $\overrightarrow{r}$:

$$\overrightarrow{r}^{rot} = M_r \times \overrightarrow{r}$$

Rotation and Accumulator

- Perform the rotation for all vectors in R-Table.
- Increase A for each rotation angle α, $0 \rightarrow 2\pi$ in general case.

GHT with Scaling and Rotation

Steps to rotating the whole shape S by α

1. All R-Table's indices are increased by $-\alpha$, takes **modulo** by 2π after increasing.
   - θ : angle of gradient vector.
   - Compute $\theta^{rot} = (\theta - \alpha)$ **modulo** 2π
   - $\equiv$ Treat R-Table as a circular buffer, shift θ around the circular buffer an amount $-\alpha$
2. All vectors found at θ^r are rotated by α

GHT with Scaling and Rotation

Algorithm 16 Detection of instances of S with scaling and rotation

```
Create a 4D-Accumulator, referred to as A(x_c, y_c, r, s)
Set 0 for all cells in the accumulator
for all edge point in I_e(x, y) do
    Create vector p = [x, -y]^T    ▷ negative y, because y-axis: upright, x-axis: to-right
    Obtain gradient vector \vec{g} at I_e(x, y)
    Compute angle θ of \vec{g}
    for all rotation angle α do
        Compute θ^{rot} = (θ − α) modulo 2π
        Find R-TABLE's row: l = cdm_{θ2idx}(θ^{rot})
        Get List L of vector \vec{r} from Row l
        Compute rotation matrix M_r(α)
```

GHT with Scaling and Rotation

Algorithm 17 Detection of instances of S with scaling and rotation

- **for all** scaling factor s **do**
  - Compute scaling-matrix $M_s(s)$
  - **for all** vector $\overrightarrow{r}_i$ in L **do**
    - Transform $\overrightarrow{r}_i^{trans} = M_r \times M_s \times \overrightarrow{r}_i$
    - Compute $\overrightarrow{c} = \overrightarrow{p} + \overrightarrow{r}_i^{trans}$
    - Determine (x_c, y_c) from $\overrightarrow{c}$
    - $A(x_c, y_c, r, s) = A(x_c, y_c, r, s) + \Delta(x, y)$
  - **end for**
- **end for** ▷ scaling factor s

end for ▷ rotation angle α

end for ▷ edge point $I_e(x, y)$

Find the largest cell in $A(x_c, y_c, r, s)$, assume at (x_c^*, y_c^*, r^*, s^*)