Chapter 2

Hough Transforms
II. Hough Transforms

1. (The Original) Hough Transform
   
   - Good for line and curve detections

(1) Slope-Intercept (m-c) parameter space

\[
y = mx + c
\]

Properties:

a) A point in the image space corresponds to a line in the m-c space.

b) A point in the m-c space corresponds to a line in the image space.

c) Points lying on the same line in the image space correspond to lines passing through a common point in the m-c space.

d) Points lying on the same line in the m-c space correspond to lines passing through a common point in the image space.

Disadvantage: vertical lines have \( m \to \infty \)

( a huge m-c space ! )
Hough Transform

\[ y = mx + c' \]

\[(x', y') (x'', y'')\]

\[ c = -mx'' + y'' \]

A line (a) in image space

(b) in parameter space

(slope - intercept)

\[ (m - c) \]
Hough Transform Algorithm (Line Detection)

1. Form a parameter space with max and min values for \( c \) and \( m \).
2. Form an accumulator array \( A(c,m) \).
3. For each edge point \((x,y)\) with its magnitude-of-gradient \( s > \text{threshold} \), increment all points in the accumulator array satisfying \( c = -xm + y \).
4. Local maxima in \( A(c,m) \) now corresponds to collinear points (line) in the image array.

(The values of peaks in \( A(c,m) \) provide a measure the number of points on the line.)
(2) Normal Parameterization

\[ \rho = x_i \cos \theta + y_i \sin \theta \]
\[
\rho = x_i \cos \theta + y_i \sin \theta \\
= \sqrt{x_i^2 + y_i^2} \left( \frac{x_i}{\sqrt{x_i^2 + y_i^2}} \cos \theta + \frac{y_i}{\sqrt{x_i^2 + y_i^2}} \sin \theta \right) \\
= \sqrt{x_i^2 + y_i^2} \left( \cos \phi \cos \theta + \sin \phi \sin \theta \right) \\
= \sqrt{x_i^2 + y_i^2} \cos (\theta - \phi) \\
= r \cdot \cos (\theta - \phi)
\]
In mathematics, a line can be represented as
\[ \rho = x \cos \theta + y \sin \theta \]
where \( 0 \leq \theta < 2\pi \), \( 0 \leq \rho \leq \rho_{\text{max}} \)

\text{OR} \quad 0 \leq \theta < \pi \quad -\rho_{\text{max}} \leq \rho \leq \rho_{\text{max}}

(\rho_{\text{max}} \geq 0)

In computer vision, no linear object can have width \( \rightarrow 0 \). A line is usually a boundary between two regions, and the gradient direction \( \theta \) is perpendicular to the line, pointing towards the brighter side.

Therefore in implementing the Hough Transform
\[ 0 \leq \theta < 2\pi \quad -\rho_{\text{max}} \leq \rho \leq \rho_{\text{max}} \]
(3) **Discussion**

The Hough method can recognize any curve \( f(x, \vec{a}) = 0 \) where \( \vec{a} \) is a parameter vector.

**Ex. Circles.**

\[
(x_i - a)^2 + (y_i - b)^2 = r^2
\]

or

\[
\begin{align*}
(a_i - x)^2 + (b_i - y)^2 &= r^2 \\
\end{align*}
\]

To find the center of a circle with radius \( r \):

(without gradient info)

**Image space**

**Parameter space**

(for a fixed radius \( r \) )
** In general, to find circles with all possible \( r \), increment all points on a right circular cone in a 3D parameter space.

Advantages and disadvantages of Hough Transform

+: works for broken lines, curves
+

-: insensitive to noise

-: could be expensive (computationally) when number of parameters is large.
(4) Use Gradient Information.

- For line detection: for each edge point \((x_0, y_0)\), only increment at one \((ρ, θ)\) point (or \((m, c)\) point) in the parameter space.
  
  \(\text{The gradient direction} \ θ \ \text{can directly be used to calculate} \quad ρ = x_0 \cdot \cos θ + y_0 \cdot \sin θ\)

- For circle detection (radius = \(r\)):

  for each edge point \((x_0, y_0)\), only increment at one \((a, b)\) point in the parameter space.

  \(\text{The gradient direction can directly be used}\)
  
  \(a = x_0 + r \cdot \cos θ\)

  \(b = y_0 + r \cdot \sin θ\)
** In general, to find all possible circles (non-fixed \( r \)), increase counts along a line in the 3D parameter space.

- Making use of gradient info saves one dimension in increasing the possible cells in the accumulator arrays.
** For ellipse:

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

its derivative \[
\frac{x-x_0}{a^2} + \frac{y-y_0}{b^2} \cdot \frac{dy}{dx} = 0
\]

\(\text{gradient direction}\)

\[
x = x_0 \pm \frac{a}{\sqrt{1 + \frac{b^2}{a^2} \left(\frac{dy}{dx}\right)^2}}
\]

\[
y = y_0 \pm \frac{b}{\sqrt{1 + \frac{a^2}{b^2} \left(\frac{dy}{dx}\right)^2}}
\]

Four parameters \((a, b, x_0, y_0)\)
\[ \theta \cdot p \approx \nu \]

\[ p \cdot \nu \approx \sum (\theta - \phi) \frac{\theta \cdot e}{\nu} + \nu \cdot \frac{\nu \cdot \theta}{\nu} = 0 \]

\[ (\theta - \phi) \cos \frac{\nu + \theta}{\nu} = (\theta \cdot \nu \frac{\nu + \theta}{\nu} + \theta \cos \frac{\nu + \theta}{\nu}) \frac{\nu + \theta}{\nu} = \theta \cdot \nu + \theta \cos \nu = \nu \]
Hierarchical Hough method for line detection:

![Diagram of line detection](image)

**Figure 3-3:**

The $\theta$ and $\rho$ values of the child segments are recalculated with reference to the current space's coordinate system during the merging process. It essentially involves a translation of the origin of the child space to the current space. $(x_{o_i}, y_{o_i})$ and $(x_{o_{i-1}}, y_{o_{i-1}})$ are the origins of the image spaces in the levels $i$ and $i-1$ respectively; $\theta_i$ and $\theta_{i-1}$ are the $\theta$ estimates; $\rho_i$ and $\rho_{i-1}$ are the $\rho$ estimates and $d$ measures the translational displacement of the foot of normal.
2. Generalized Hough Transform (GHT)

- For general object recognition based on the shape of its boundary.
- A generalized form of Template Matching.

Modeling:

Pre-compute locations of the boundary points with respect to a reference point, use the gradient angle as index to store all such locations in an R-table – i.e., to build a model.

Matching:

a) Increment the possible loci of the reference points in an accumulator array according to the information of the edge pixels in the image space.

b) Find peaks in the accumulator array.
\[
\begin{align*}
\langle \phi \rangle \times \sin \cdot \langle \phi \rangle \cdot r + h &= \hat{h} \\
\langle \phi \rangle \times \cos \cdot \langle \phi \rangle \cdot r + x &= \hat{x}
\end{align*}
\]

\[
\begin{array}{c|c}
 m & \phi \\
 \vdots & \vdots \\
 r^m & \phi \\
 r^1 & \phi \\
 r^1 & \phi \\
 \hline
\r = (r, \alpha) \\
\text{Set of Radii } R^r \\
\end{array}
\]

Set of Radii

\[
\text{Generalized Hough Transform}
\]

\[R - \text{table: } \text{Ballard 1981}\]
Algorithm: (Generalized Hough)

0. Make R-table for the shape to be located.
1. Form an accumulator array
   \[ A( x_{\text{min}} : x_{\text{max}}, y_{\text{min}} : y_{\text{max}}) \]
2. For each edge point \( \bar{x} \), do
   2.1. Get \( \phi(\bar{x}) \)
   2.2a. Calculate possible centers, i.e. for each table entry for \( \phi \), compute
   \[ x_c = x + r(\phi) \cdot \cos \left[ \alpha(\phi) \right], \]
   \[ y_c = y + r(\phi) \cdot \sin \left[ \alpha(\phi) \right]. \]
   2.2b. \[ A(x_c, y_c) = A(x_c, y_c) + 1 \]
3. Possible locations for the shape are given by maxima in array \( A \).
Rotate it by $\theta$ (i.e., $x' = x + \theta$).

R-table. After it is found, gradient angle to look up the

case $[\phi, \theta - \phi \mod 2\pi] 

\{ \theta [\phi - \theta \mod 2\pi], \theta \} \text{Rot}

\theta + \phi = \phi

\theta + x = x$.

Rotation in GHT.
Modeling:

R-Table:

<table>
<thead>
<tr>
<th>$0^\circ$</th>
<th>$\vec{r}_0$</th>
<th>$\vec{r}_1$</th>
<th>$\vec{r}_2$</th>
<th>$\vec{r}_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$90^\circ$</td>
<td>$\vec{r}_9$</td>
<td>$\vec{r}_1$</td>
<td>$\vec{r}_2$</td>
<td>$\vec{r}_9$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Matching:

Rotate by $\theta = 45^\circ$  Scale up by $s = 2$

Discussion:

1. Quantization
2. Error analysis
Discussion:

Use \((\Delta x, \Delta y)\) instead of \((r, \alpha)\).

The vector \(\vec{r} = (r, \alpha)\) in the \(R\)-table could be replaced by \(\vec{r} = (\Delta x, \Delta y)\).

In other words:

\[
\begin{align*}
    x_c &= x + \Delta x \\
    y_c &= y + \Delta y
\end{align*}
\]

When there is a rotation of \(\Theta\) (counter-clockwise),

\[
\begin{pmatrix}
    \Delta x' \\
    \Delta y'
\end{pmatrix} =
\begin{pmatrix}
    \cos \Theta & -\sin \Theta \\
    \sin \Theta & \cos \Theta
\end{pmatrix} \cdot
\begin{pmatrix}
    \Delta x \\
    \Delta y
\end{pmatrix}
\]
Fig. 2 Multi-level Hierarchical Modeling Using Salient Features
Figure 6: An Accumulator Array ($\theta = 180^\circ$) for the Detection of a Key Head at Level 6.

<table>
<thead>
<tr>
<th>X Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 5 10 15 20 25 30</td>
</tr>
</tbody>
</table>

```
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>