# Visibility of a Simple Polygon from a Point

## 1. Notation and preliminary results

The **boundary** of a simple polygon $P$ consists of *a sequence of straight-line segments* such that they *form a cycle* and *no two nonconsecutive edges intersect*.

$P$ is represented by a list of vertices in **counter-clockwise order**.

Denote boundary, interior and exterior of $P$ by $Bd(P)$, $Int(P)$ and $Ext(P)$ respectively

→ $P = Bd(P) \cup Int(P)$.

Consider the vision $z$ in $Int(P)$ or on $Bd(P)$.

If $z \in Int(P)$, denote the vertices of $P$ as $v_0, v_1, \ldots, v_{n-1}$ and $v_n = v_0$.

If $z \in Bd(P)$, denote the vertices of $P$ as $z, v_0, v_1, \ldots, v_n$ ($v_n$ is the predecessor of $z$).

The edges of $P$ are $(zv_0), v_0v_1, \ldots, v_{n-1}v_n, (v_n, z)$.

---

Translate and rotate the coordinate system so that $z$ *is at the origin* and

if $z \in Int(P)$, relabel the vertices so that $v_0 = v_n$ is *on the positive x-axis* and has the *smallest x-coordinate*, we may need to add $v_0$.

if $z \in Bd(P)$, $v_0$ is *on the positive x-axis*.

Denote the polar coordinates of a point $v$ by $(r(v), \theta(v))$ where $0 \leq \theta(v) < 2\pi$.

---

A subset of $Bd(P)$ composed of the chain $sv_iv_{i+1}, \ldots, v_{k-1}v_kt$ where $s \in v_{i-1}v_i$ and $t \in v_kv_{k+1}$, is denoted as $Ch[s, t]$ or $Ch(s, t)$.

The chain $uvw$ is said to be a *left turn*, *right turn* or *no turn* if $w$ is to the left, right, or on, respectively, the line from $u$ to $v$.

---

The point $v$ is *visible* from $z$ with respect to $P$ if the line segment joining $z$ and $v$ lies completely in $Int(P)$.

Denote $V(P, z)$ as the *star-shaped simple polygon* visible from $z$, which is the closure of the set $\{v \in P | v\ is\ visible\ from\ z\}$.

Denote $\bar{V}(P, z)$ as the subset containing *all points on* $Bd(P)$ which are visible from $z$.

---

The **angular displacement** $\alpha(v)$ of a point $v$ on $Bd(P)$ with respect to $z$ is

$$\alpha(v_0) = \theta(v_0)$$

$$\forall 1 \le i \le n, \alpha(v_i) = \begin{cases} \alpha(v_{i-1}) + \widehat{(v_{i-1} z v_i)} & if\ z v_{i-1} v_i\ is\ a\ left\ turn \\ \alpha(v_{i-1}) - \widehat{(v_{i-1} z v_i)} & if\ z v_{i-1} v_i\ is\ a\ right\ turn \\ \alpha(v_{i-1}) & if\ z, v_{i-1}, v_i\ are\ colinear \end{cases}$$

For other $v \in v_{i-1} v_i$ where $1 \le i \le n$, $\alpha(v)$ can be defined as $\theta(v) + 2\pi k$ for some $k$ such that $\alpha(v)$ is a continuous function as $Bd(P)$ is traversed from $v_0$ to $v_n$.

In case $z \in Bd(P)$, $\forall v \in z v_0, \alpha(v) = \alpha(v_0)$ and $\forall v \in v_n z, \alpha(v) = \alpha(v_n)$.

Let $\beta = \widehat{(v_n z v_0)}$, the **interior angel** of $z$ in $P$.

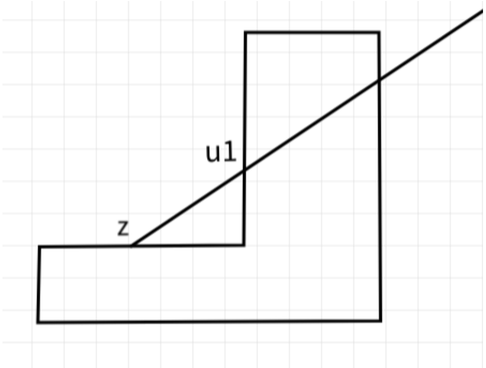
It can be derived that

$$\alpha(v_n) - \alpha(v_0) = \begin{cases} 2\pi & if\ z \in Int(P) \\ \beta & if\ z \in Bd(P) \\ 0 & if\ z \in Ext(P) \end{cases}$$

**Lemma 1**: If the intersection of the ray emanating from $z$ at polar angle $0 \le \phi < 2\pi$ and $Int(P)$ is nonempty, then it consists of line segments $u_1 u_2, u_3 u_4, \ldots, u_{2k-1} u_{2k}$, $k \ge 1$ where

1. $u_1 = z$ **or** $u_1 \in Bd(P)$, and $\forall i > 1, u_i \in Bd(P)$.

   Example of case $u_1 \neq z$:




2. The interior of $u_{2i-1}u_{2i}$ is in $Int(P)$.
3. $0 \leq r(u_1) < r(u_2) \leq r(u_3) < r(u_4) \leq \cdots \leq r(u_{2k-1}) < r(u_{2k})$.
4. If $u_{2i-1} \neq z$, it is on an edge which is oriented in the **clockwise** direction with respect to $z$.
5. $u_{2i}$ is on an edge which is oriented in the **counterclockwise** direction with respect to $z$.

In short, the ray is composed of line segments which alternate between being in $Int(P)$ and $Bd(P) \cup Ext(P)$.

The line segments in $Int(P)$ have positive length and don't include their endpoints (except $u_1 = z \in Int(P)$). The line segments in $Bd(P) \cup Ext(P)$ may have zero length: a single point.

It follows from Lemma 1 that at each polar angel $\phi$, there is **at most one** point $v$ **visible** from $z$ and on an edge which is oriented in the **counterclockwise** direction with respect to $z$.

**Lemma 2**: For $1 \leq i \leq k$, if $u_{2i-1} \neq z$, $\alpha(u_{2i-1}) = \alpha(u_{2i})$.

Proof:

If $u_{2i-1} \neq z$, the line segment $u_{2i-1}u_{2i}$ divides $P$ into two simple sub-polygons.

If $u_{2i-1}$ comes before $u_{2i}$ in the traversal of $Bd(P)$ from $v_0$ to $v_n$, then let $Q$ be the sub-polygon formed by $Ch[u_{2i-1}u_{2i}]$ and the line segment $u_{2i-1}u_{2i}$. Else, let Q be the sub-polygon formed by $Ch[u_{2i}u_{2i-1}]$ and the line segment $u_{2i-1}u_{2i}$.

→ $z \in Ext(Q)$ and $z, u_{2i-1}, u_{2i}$ are colinear → $\alpha(u_{2i-1}) = \alpha(u_{2i})$ **on** $Bd(Q)$ by the definition of angular displacement.

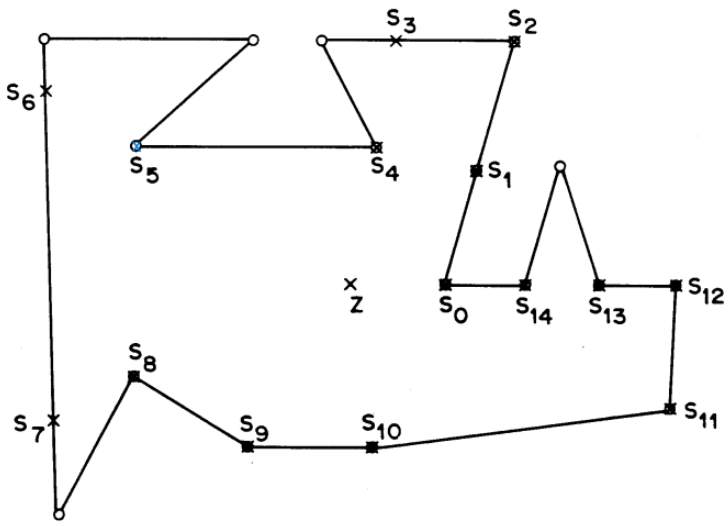
This implies $\alpha(u_{2i-1}) = \alpha(u_{2i})$ on $Bd(P)$.

---

Denote the visibility polygon by the chain $s_0 s_1 \ldots s_t$ in which $s_0 = v_0$, $s_t = v_n$, $\forall 1 \leq j \leq t, s_j \in Bd(P)$, $s_j s_{j+1} \subset Bd(P)$ if $\theta(s_j) \neq \theta(s_{j+1})$, and

if $z \in Bd(P)$, $0 = \theta(s_0) \leq \cdots \leq \theta(s_m) = \beta$, $m = t$.

if $z \in Int(P)$, $0 = \theta(s_0) \leq \cdots \leq \theta(s_m) < 2\pi$, $\theta(s_{m+1}) = \cdots = \theta(s_t) = 0$ , $m < t$.




**Lemma 3**: If $v \in Bd(P)$ is visible from $z$ and $v \neq v_n$, then $\alpha(v) = \theta(v)$.

Proof:

Firstly, prove that $\forall 0 \leq j \leq m, \alpha(s_j) = \theta(s_j)$.

For $j = 0$, $s_0 = v_0$ so $\alpha(v_0) = \theta(v_0) = 0$. Assume that this is true for $j < m$.

If $\theta(s_{j+1}) > \theta(s_j)$ then $s_j s_{j+1} \subset Bd(P)$ → $\alpha(s_{j+1}) = \alpha(s_j) + \widehat{(s_j z s_{j+1})} = \theta(s_j) + \widehat{(s_j z s_{j+1})} = \theta(s_{j+1})$.

If $\theta(s_{j+1}) = \theta(s_j)$ and $s_j s_{j+1} \subset Bd(P)$ ($s_{13}$ and $s_{14}$) → By definition, $\alpha(s_{j+1}) = \alpha(s_j) = \theta(s_j) = \theta(s_{j+1})$.

If $\theta(s_{j+1}) = \theta(s_j)$ and $s_j s_{j+1} \not\subset Bd(P)$ ($s_5$ and $s_6$), then from the proof of Lemma 2, we can deduce $\alpha(s_{j+1}) = \alpha(s_j) = \theta(s_j) = \theta(s_{j+1})$.

By induction, the above hypothesis is true.

Then, for $v$ on the interior of an edge $s_j s_{j+1}$ where $\alpha(s_j) < \alpha(s_{j+1}), \alpha(s_j) = \theta(s_j)$, and either $\alpha(s_{j+1}) = \theta(s_{j+1})$ or $\alpha(s_{j+1}) = 2\pi$, $\alpha(v) = \theta(v)$ by definition of angular displacement.

$\forall m < j < t$, $s_j$ and the interior of the segments $s_j s_{j+1}$ are blocked by $v_n$, so they aren't visible from $z$.

**Corollary**: If $v \in Bd(P)$ and either $\alpha(v) < 0$ or $\alpha(v) > 2\pi$, then $v$ is not visible from $z$.

# 2. The algorithm

The algorithm scans $Bd(P)$ monotonously from edge $v_0 v_1$ to $v_{n-1} v_n$, while manipulating a stack of vertices $s_0, s_1, \ldots, s_t$ such that ultimately the chain $(z), s_0, s_1, \ldots, s_t$ becomes $Bd(V(P, z))$.

Preprocessing: Computation of angular displacement for all $v_i$ for future reference.

Each step in the algorithm is 1 of 3 procedures:

- ADVANCE **adds vertices** to the stack.
- RETARD **removes vertices** from the stack.
- SCAN **scans invisible edges** of $Bd(P)$ without modifying the stack.

The input to these procedures includes the current edge $v_i v_{i+1}$. Each procedure *may work on successive boundary segments* until it determines one that *requires treatment by another procedure*.

SCAN requires an **interval, or 'window'**, across which a segment of $Bd(P)$ must pass in order to leave the SCAN process, and *the manner of crossing*: clockwise or counterclockwise. One end of the window is *always the top of the stack*, $s_t$, and the other end $w$ is a point on the ray $zs_t$. The direction is implied in the logical variable $ccw$ (true for counterclockwise).

---

Suppose $Ch[v_0 v_i]$ has been scanned and the stack contains $s_0, s_1, \ldots, s_t$. Let $S_i = chain\ s_0, s_1, \ldots, s_t$. Then the following stack properties are satisfied:

1. $0 = \alpha(s_0) \leq \alpha(s_1) \leq \cdots \leq \alpha(s_t) \leq 2\pi$, $s_0 = v_0$ and $0 \leq t \leq i$.
2. $s_j \in Bd(P)$ and $s_j s_{j+1} \subset Bd(P)$ if $\alpha(s_j) < \alpha(s_{j+1})$.
3. If $v \in Ch[v_0 v_i]$ but $v \notin S_i$ then $v$ is not visible from $z$.

---

def VISPOL($z, [v], n$):

# Input: vision point $z$ and $n$ vertices $[v]$ satisfying assumption in Section 2

# Output: visibility polygon vertices $[s] = s_0, s_1, \ldots, s_t$ where $s_0 = v_0, s_t = v_n$ (initially empty)

$i := 0; t := 0;$

$[s] := n$-array; $s_0 := v_0;$

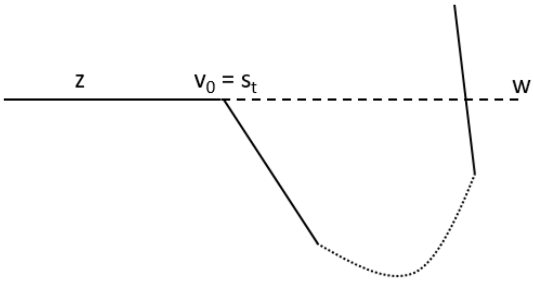
if $\alpha(v_1) \geq \alpha(v_0)$:

$action :=$ '$advance$';

else:

$action :=$ '$scan$'; $ccw = True$;

$w = (\infty, \theta(v_0))$; # polar coordinates




while $action \neq$ '$finish$':

match $action$:

case '$advance$': ADVANCE($z, [v], n, [s], t, i, action, ccw, w$);

case '$retard$': RETARD($z, [v], n, [s], t, i, action, ccw, w$);

case '$scan$': SCAN($z, [v], n, [s], t, i, action, ccw, w$);

return ($[v], t$);

---

def ADVANCE($z, [v], n, [s], t, i, action, ccw, w$):

# Modify: $[s], t, i, action, ccw, w$

# Pre-condition: $\alpha(v_{i+1}) \geq \max(\alpha(v_i), \alpha(s_t))$ and $s_t \in v_i v_{i+1}, s_t \neq v_{i+1}$.

while $action =$ '$advance$':

if $\alpha(v_{i+1}) \leq 2\pi$:

$i := i + 1; t := t + 1;$

$s_t := v_i;$

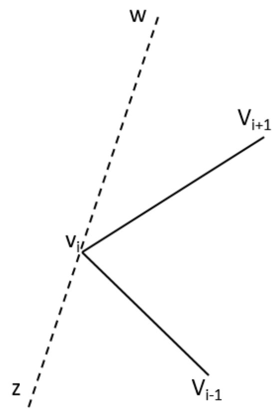
if $i = n$:

$action =$ '$finish$';

else if $\alpha(v_{i+1}) < \alpha(v_i)$ and $v_{i-1}v_iv_{i+1}$ is a right turn:

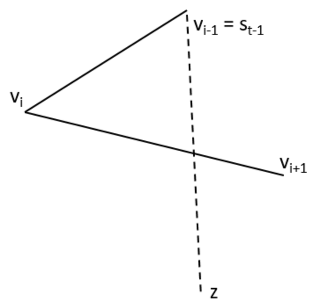
$action = \text{'}scan\text{'}; ccw = True;$

$w = (\infty, \theta(v_0));$




else if $\alpha(v_{i+1}) < \alpha(v_i)$ and $v_{i-1}v_iv_{i+1}$ is a left turn:

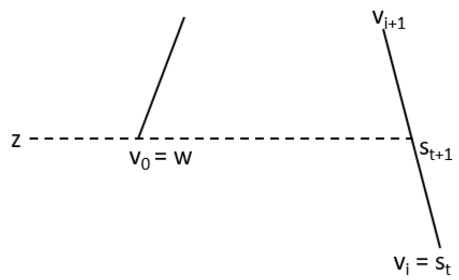
$action = \text{'}retard\text{'};$




else: # $\alpha(v_{i+1}) > 2\pi$

if $\alpha(v_i) < 2\pi$:

$t := t + 1; s_t :=$ intersection of $v_iv_{i+1}$ and $\overrightarrow{zv_0}$;




$action = \text{'}scan\text{'}; ccw = False; w = v_0;$

---

def RETARD($z, [v], n, [s], t, i, action, ccw, w$):

# Pre-condition:

# If RETARD is entered after exiting ADVANCE, then $s_t = v_i$ and $\alpha(v_{i+1}) < \alpha(v_i)$.

# If RETARD is entered after exiting SCAN, then $\alpha(v_{i+1}) \leq \alpha(s_t) < \alpha(v_i)$.

# → Either $v_{i+1} \in z\bar{s}_t$ or part of edge $v_iv_{i+1}$ is in front of chain $s_0, s_1, \ldots, s_t$ with respect to $z$.

while $action =$ '$retard$':

scan backward $s_{t-1}, \ldots, s_0$ for **first** vertex $s_j$ such that either

(a) $\alpha(s_j) < \alpha(v_{i+1}) \leq \alpha(s_{j+1})$, or

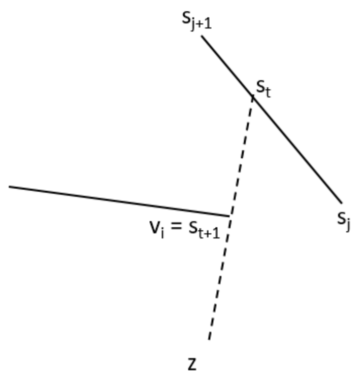
(b) $\alpha(v_{i+1}) \leq \alpha(s_j) = \alpha(s_{j+1})$ and $v_iv_{i+1}$ intersects $s_js_{j+1}$.

if $\alpha(s_j) < \alpha(v_{i+1})$ (a):

$i := i + 1; t := j + 1;$

$s_t :=$ intersection of $s_js_{j+1}$ and $\overrightarrow{zv_i}$;

$s_{t+1} := v_i; t := t + 1;$

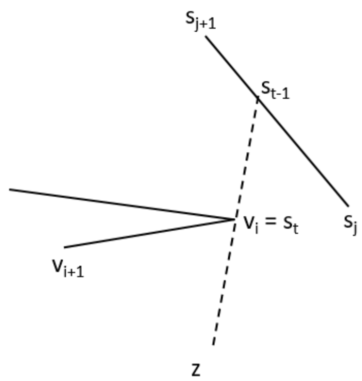



if $i = n$:

$action :=$ '$finish$';

else if $\alpha(v_{i+1}) \geq \alpha(v_i)$ and $v_{i-1}v_iv_{i+1}$ is a right turn:

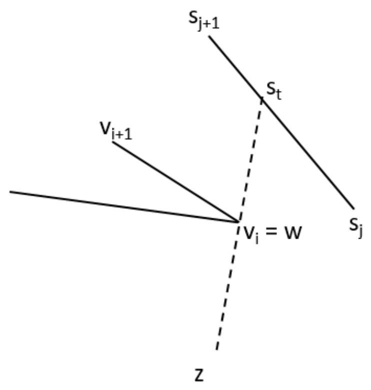
$action :=$ '$advance$';



else if $\alpha(v_{i+1}) > \alpha(v_i)$ and $v_{i-1}v_iv_{i+1}$ is a left turn:

$action = \text{'}scan\text{'}; ccw = False;$

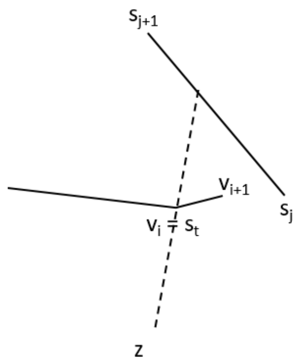
$t := t - 1; w = v_i;$

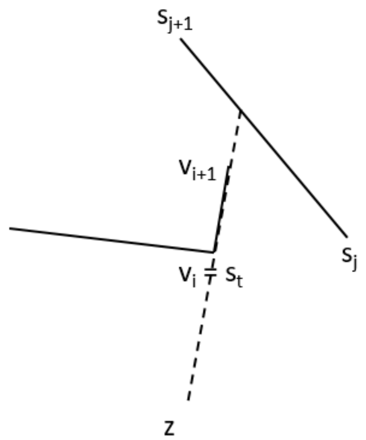



else:

$t := t - 1;$

# continue to 'retard' if either $\alpha(v_{i+1}) < \alpha(v_i)$, or $\alpha(v_{i+1}) = \alpha(v_i)$ and $r(v_{i+1}) > r(v_i)$.



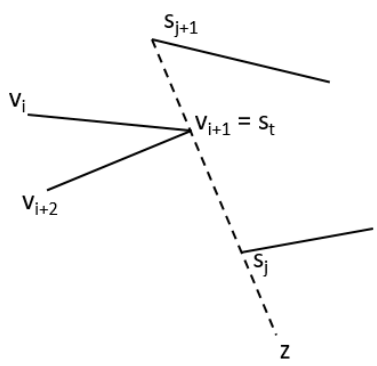




else (b):

if $\alpha(v_{i+1}) = \alpha(s_j)$ and $\alpha(v_{i+2}) > \alpha(v_{i+1})$ and $v_iv_{i+1}v_{i+2}$ is a right turn:

$action := \text{'}advance\text{'}; i := i + 1;$

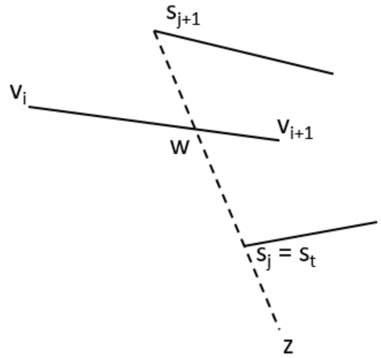
$t := j + 1; s_t := v_i;$



else:

$action = \text{‘}scan\text{’}; ccw = True;$

$t := j; w =$ intersection of $s_j s_{j+1}$ and $v_i v_{i+1}$;




def SCAN($z, [v], n, [s], t, i, action, ccw, w$):

# Pre-condition: $\theta(s_t) = \theta(w)$

while $action = \text{‘}scan\text{’}$:

$i := i + 1;$

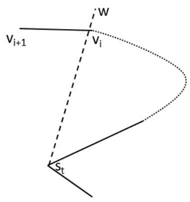
if $ccw$ and $\alpha(v_{i+1}) > \alpha(s_t) \geq \alpha(v_i)$:

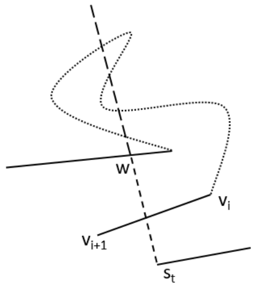
if $v_i v_{i+1}$ intersects $s_t w$:

$s_{t+1} :=$ intersection of $v_i v_{i+1}$ and $s_t w$;

$action = \text{‘}advance\text{’}; t := t + 1;$

# Let $u$ be the intersection and $v$ be a point on $Ch(s_t, u)$, then $v$ lies in the region bounded by $Ch(s_t, u)$ and line segment $us_t$ → $v$ is not visible from $z$.



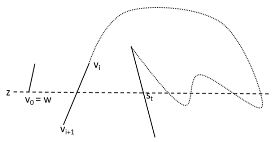



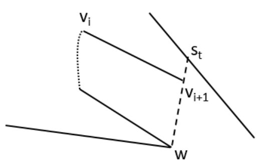

else if not $ccw$ and $\alpha(v_{i+1}) \leq \alpha(s_t) < \alpha(v_i)$:

if $v_i v_{i+1}$ intersects $s_t w$:

$action = \text{‘}retard\text{’};$

# A point $v \neq v_{i+1}$ on edge $v_i v_{i+1}$ is not visible from $z$.



# 3. Correctness and running time

**Theorem 1**: $\forall 0 \le i \le n$, the stack properties are satisfied after $Ch[v_0, v_i]$ has been scanned, and the algorithm terminates with $s_t = v_n$.

Proof:

Initially, $i = 0, s_0 = v_0, \alpha(s_0) = 0$.

In ADVANCE, $s_t$ satisfies $\alpha(s_{t-1}) \le \alpha(s_t) \le 2\pi$ and $s_{t-1}s_t \subset Bd(P)$.

In RETARD, vertices $s_{j+1}, \ldots, s_t$ are popped from the stack but $s_0 = v_0$ remains, so $t \ge 0$. $s_t$ <u>is either on edge</u> $s_j s_{j+1}$ such that $s_{t-1} = s_j, \alpha(s_t) > \alpha(s_{t-1}), s_{t-1}s_t \subset Bd(P)$ or <u>is</u> $v_i$ <u>such that</u> $\alpha(s_t) = \alpha(s_{t-1})$.

In SCAN, $s_t$ <u>is on edge</u> $v_i v_{i+1}$ <u>such that</u> $\alpha(s_t) = \alpha(s_{t-1})$.

$t \le i$ since $$t is increased by at most once each time we increment $i$.

**Theorem 2**: $Bd(V(P, z)) = S_n \cup \{v_n z, z v_0 \ if \ z \in Bd(P)\}$ where $S_n$ is the chain $s_0, s_1, \ldots, s_t$ when the algorithm finishes.

Proof:

Let $v$ be a point on $Ch[v_0, v_n]$.

If $v \notin S_n$, then $v$ is not visible from $z$ by the third property.

If $v \in S_n$ and $v$ is the closet point on $S_n$ to $z$ with polar angle $\theta(v)$, then $v \in Bd(P)$ and visible from $z$.

If $v \in S_n$ but $v$ is not the closet point to $z$ with polar angle $\theta(v)$, then $v$ is on an edge $s_j s_{j+1}$ in which $\alpha(s_j) = \alpha(s_{j+1})$. These edges are added to connect the visible points on $Bd(P)$.

**Theorem 3**: The algorithm runs in $O(n)$.

Proof:

Each edge $v_i v_{i+1}$ on $Bd(P)$ is processed once in the algorithm and at most 2 points on this edge are pushed onto the stack. Therefore, at most $2n$ points are popped from the stack in RETARD. The computation of angular displacement, the intersection of two lines and determining whether $v_{i-1}v_i v_{i+1}$ is a left, right or no turn can each be done in constant time.