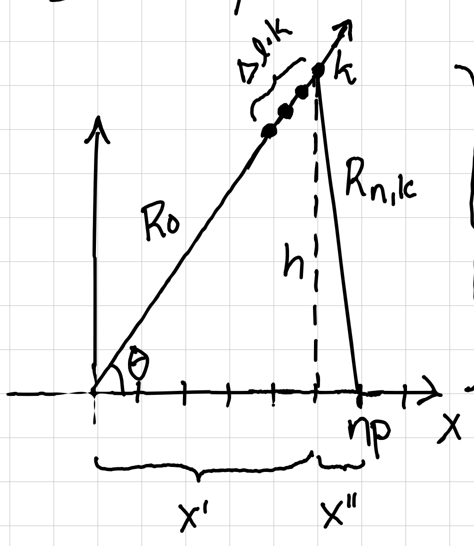
Delay to point in scanline



$$R_n^2 = x''^2 + h^2 \qquad x'' = np - x'$$

$$h = (R_0 + k\cdot\Delta l)\sin\theta \qquad x'' = np - (R_0 + k\cdot\Delta l)\cos\theta$$

$$x' = (R_0 + k\cdot\Delta l)\cos\theta$$

The distance from element n to point on scanline k:

$$\begin{aligned} R_{n,k}^2 &= (np - (R_0 + k\cdot\Delta l)\cos\theta)^2 + ((R_0 + k\cdot\Delta l)\sin\theta)^2 \\ &= (np)^2 - 2np(R_0 + k\cdot\Delta l)\cos\theta + (R_0 + k\cdot\Delta l)^2(\cos^2\theta + \sin^2\theta) \end{aligned}$$

$$R_{n,k}^2 = (np)^2 - 2np(R_0 + k\cdot\Delta l)\cos\theta + (R_0 + k\cdot\Delta l)^2$$

The distance from element n+1 to point on scanline k:

$$\begin{aligned} R_{n+1,k}^2 &= (np+p)^2 - 2(n+1)p(R_0 + k\cdot\Delta l)\cos\theta + (R_0 + k\cdot\Delta l)^2 \\ &= (np)^2 + 2np^2 + p^2 - 2np(R_0 + k\cdot\Delta l)\cos\theta \\ &\qquad - 2p(R_0 + k\cdot\Delta l)\cos\theta + (R_0 + k\cdot\Delta l)^2 \\ &= (np)^2 - 2np(R_0 + k\cdot\Delta l)\cos\theta + (R_0 + k\cdot\Delta l)^2 \\ &\qquad + 2np^2 + p^2 - 2p(R_0 + k\cdot\Delta l)\cos\theta \end{aligned}$$

$$R_{n+1,k}^2 = R_{n,k}^2 + p^2(2n+1) - 2p(R_0 + k\cdot\Delta l)\cos\theta$$

The distance from element n to point on scanline k+1:

$$\begin{aligned} R_{n,k+1}^2 &= (np)^2 - 2np(R_0 + k\cdot\Delta l + \Delta l)\cos\theta + (R_0 + k\cdot\Delta l + \Delta l)^2 \\ &= (np)^2 - 2np(R_0 + k\cdot\Delta l)\cos\theta - 2np\cdot\Delta l\cos\theta \\ &\qquad + (R_0 + k\cdot\Delta l)^2 + 2(R_0 + k\cdot\Delta l)\Delta l + \Delta l^2 \end{aligned}$$

$$R_{n,k+1}^2 = R_{n,k}^2 - 2np\cdot\Delta l\cos\theta + 2R_0\Delta l + \Delta l^2(2k+1)$$

Using the relation $N_{n,k} = \frac{f_s}{V_s} R_{n,k}$ we get the delay in sample frequency cycles:

$$N_{n+1,k}^2 = N_{n,k}^2 + \left(\frac{f_s}{V_s}p\right)^2(2n+1) - 2p\left(\frac{f_s}{V_s}\right)^2(R_0 + k\cdot\Delta l)\cos\theta$$

$$N_{n,k+1}^2 = N_{n,k}^2 - 2np\left(\frac{f_s}{V_s}\right)^2\Delta l\cos\theta + 2R_0\left(\frac{f_s}{V_s}\right)^2\Delta l + \left(\frac{f_s}{V_s}\Delta l\right)^2(2k+1)$$

Setting step size $\Delta l = \frac{V_s}{f_s}$ we get:

$$N_{n,k+1}^2 = N_{n,k}^2 - 2np\frac{f_s}{V_s}\cos\theta + 2R_0\frac{f_s}{V_s} + 2k + 1$$

We now have the delay expressions for each element and point on scanline iteratively based on previous delay:

$$N_{0,0}^2 = \left(\frac{f_s}{V_s}R_{0,0}\right)^2 \qquad A_0 = \left(\frac{f_s}{V_s}p\right)^2$$

$$N_{n+1,k}^2 = N_{n,k}^2 + A_0(2n+1) - C_0 \qquad C_0 = 2p\left(\frac{f_s}{V_s}\right)^2(R_0 + k\cdot\Delta l)\cos\theta$$

$$N_{n,k+1}^2 = N_{n,k}^2 + 2k + 1 + B_n \qquad B_n = 2R_0\frac{f_s}{V_s} - 2np\frac{f_s}{V_s}\cos\theta$$

$$n = 0, \pm 1, \pm 2, \pm 3, \ldots \qquad \Delta l = \frac{V_s}{f_s}$$

$$k = 0, 1, 2, 3, \ldots$$

Making a hardware efficient implementation

$$N_{0,0}^2 = \left(\frac{f_s}{V_s} R_{0,0}\right)^2 \qquad A_0 = \left(\frac{f_s}{V_s} p\right)^2$$

$$N_{n+1,k}^2 = N_{n,k}^2 + A_0(2n+1) - C_0 \qquad C_0 = 2p\left(\frac{f_s}{V_s}\right)^2 (R_0 + k \cdot \Delta l) \cos\theta$$

$$N_{n,k+1}^2 = N_{n,k}^2 + 2k + 1 + B_n \qquad B_n = 2R_0 \frac{f_s}{V_s} - 2np\frac{f_s}{V_s}\cos\theta$$

$$n = 0, \pm 1, \pm 2, \pm 3, \ldots$$
$$k = 0, 1, 2, 3, \ldots$$

As $N_{n,k}^2$ will be very high values and square root is hard to implement efficiently in hardware, a comparator circuit will substitute the need for square root calculations:

$$N_{n+1,k}^2 = N_{n,k}^2 + A_0(2n+1) - C_0$$

$$N_{n+1,k}^2 = N_{n,k} + K_n$$

Assuming $N_{n+1,k}$ will have the solution

$$N_{n+1,k} = N_{n,k} + a_n$$

where a_n is the number of sample frequency cycles to be added/subtracted

$$\Rightarrow (N_{n,k} + a_n)^2 = N_{n,k}^2 + K_n$$
$$N_{n,k}^2 + 2a_nN_{n,k} + a_n^2 = N_{n,k}^2 + K_n$$
$$2a_nN_{n,k} + a_n^2 = K_n$$

By solving this equation for a we get

$$N_{n+1,k} = N_{n,k} + a_n$$

The following algorithm can be implemented in hardware using a comparator, bitshift and adders:

```
$a_n = a_{n-1}$
if $K_n \geq K_{n-1}$
    sign_bit = 1
else
    sign_bit = -1
```

```
if sign_bit == 1
loop {
    if $2a_nN_{n-1,0} + a_n^2 > K_n$
        $N_{n,0} = N_{n-1,0} + a_n$
        break
    else
        $a_n$ += sign_bit · inc_step
}
```

```
else if sign_bit == -1
{
    if $2a_nN_{n-1,0} + a_n^2 < K_n$
        $N_{n,0} = N_{n-1,0} + a_n$
        break
    else
        $a_n$ += sign_bit · inc_step
} loop
```

In addition, the error $2a_nN_{n-1,0} + a_n^2 - K_n$ is propagated to the next K_{n+1} to continously correct the error from the quantization of a_n.

The increment step size mirrors the maximal error to the delay values. Decreasing the step size gives a more accurate estimate of a_n at the expense of more iterations of the loop.
This means there is a speed-accuracy tradeoff that can be adjusted by tuning the step size

The same can be done for the calculation of delay in next point of scanline k+1:

$$N_{n,k+1}^2 = N_{n,k}^2 + 2k + 1 + B_n$$
$$N_{n,k+1}^2 = N_{n,k}^2 + h_{kn}$$
$$2b_kN_{n,k} + b_k^2 = h_{kn}$$

By solving this equation for b_k we get

$$N_{n,k+1} = N_{n,k} + b_k$$

The algorith is implemented in the exact same way as for a_n:

```
$b_k = b_{k-1}$
if $h_{kn} \geq h_{(k-1)n}$
    sign_bit = 1
else
    sign_bit = -1
```

```
if sign_bit == 1
if $2b_kN_{n,k-1} + b_k^2 > h_{kn}$
    $N_{n,k} = N_{n,k-1} + b_k$
    break
else
    $b_k$ += sign_bit · inc_step
```

```
else if sign_bit == -1
if $2b_kN_{n,k-1} + b_k^2 < h_{kn}$
    $N_{n,k} = N_{n,k-1} + b_k$
    break
else
    $b_k$ += sign_bit · inc_step
```

After performing the algorithm for a_n to find delay for all elements to first point in scanline, the algorithm for b_n is performed to find each elements delay to each point k in the scanline. This leads to a maximal error of 2·increment step size to all delay values $N_{n,k}$ $(k > 0)$.

The whole system will have the following steps:

1) Find reference delay
$$N_{0,0} = R_0 \cdot \frac{f_s}{V_s}$$

2) Find delay in all elements to reference point $k=0$ iteratively based on $N_{0,0}$:
$$N_{n,0} = N_{n-1,0} + a_n$$

3) Find delay in all elements to all points in scanline iteratively based on $N_{n,0}$:
$$N_{n,k} = N_{n,k-1} + b_n$$

Since the algorithm (and therefore the circuit) to find a_n and b_n are exactly the same, the blocks can be reused. The solution is therefore suited to make design choices that directly affect speed-area tradeoffs.