More "Notes on constant 158" [and how does floor sampling work?]

Question: what's the constant 158 in the floor sampling of 3D sage's raycaster series (video 2)

<u>Step 1 – consider the **top view** on the scene</u>

Diagram 1 depicts the scene <u>from above</u>. You have:

- P – player, having world coordinates (P_x, P_y)
- S – screen (a.k.a. projection plane), having width S_w
- FoV_w - Field of view angle in the horizontal plane
- p_a – player angle, r_a = ray angle (and β is difference angle $p_a - r_a$)
- R – the result point you are interested in sampling. It's some point on the floor of your world.

Let's call the length of the ray from player P to point R be d_{res}. If we can work out this d_{res}, we can also work out the coordinates of R, by simply offsetting P with the cos() and sin() of r_a multiplied by d_{res}

To work out d_{res}, we consider that it's equal to d2P (the distance to R projected on the player angle) *compensated for the difference angle β, using the cosine*:

$$\cos(\beta) = d2P / d_{res} \rightarrow d_{res} * \cos(p_a - r_a) = d2P \rightarrow d_{res} = d2P / \cos(p_a - r_a)$$

Since p_a and r_a are known, using this approach we can reduce the problem to finding d_{res} to getting the value of d2P.

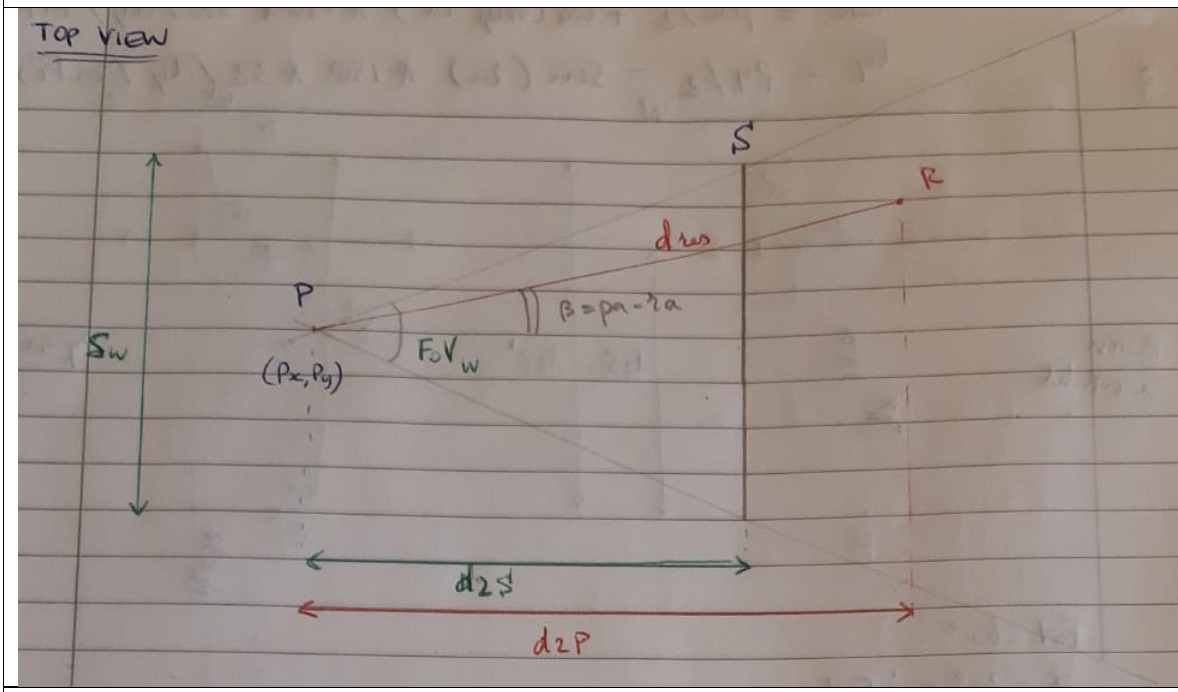


Diagram 1

Step 2 – consider the side view of the scene

We reduced the problem to calculating the distance d2P (which is the projection of d_{res} onto the ray in the direction of the player angle).

Diagram 2 depicts the same scene looking from the side. We have:

- P – the players location. The player is assumed to have height P_h
- S – the screen (or projection plane). It has height S_h
- dy – the difference between half the screen height and the pixel that you are currently considering
- d2S – the distance from the player's viewpoint to the screen

From geometry we know that if two triangles have the same angles, then the ratio's of their sides must be equal. Applied in this side view diagram:

The ratio between dy and d2S is equal to the ratio between P_h and d2P

Hence we can find d2P as follows:

$P_h / d2P = dy / d2S \rightarrow d2P = P_h * d2S / dy$

Since we know P_h (which is half a block size = 32) and we can calculate dy (the difference between the current pixel and half the screen size), we now have reduced the problem of finding d2P to finding d2S, the distance to the projection plane.

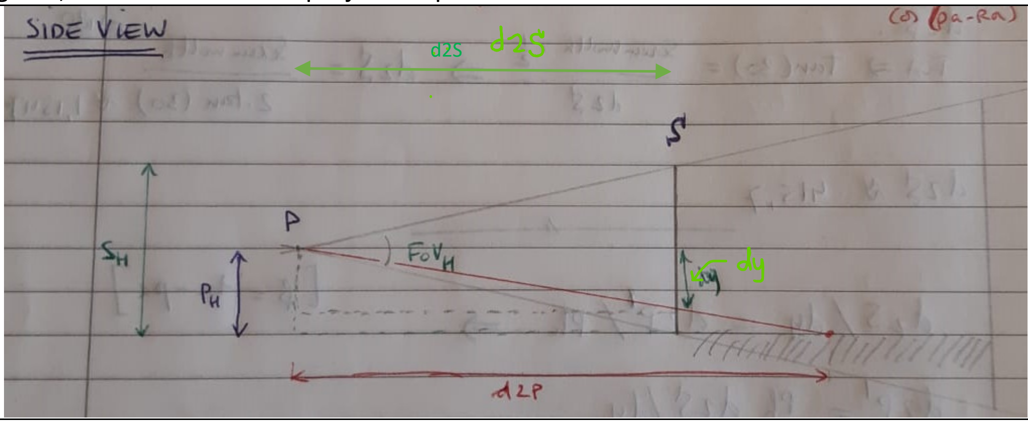


Diagram 2

Step 3 – back to top view again

We want to calculate d2S, the distance to the projection plane. Consider the right angled triangle with angle of half FoV_w, and having half the screen width as the opposite side (this triangle is highlighted in yellow in Diagram 3). We can calculate the adjacent side (which is d2S) using the tangent function:

$tan(FoV_w / 2) = opposite / adjacent = 0.5 * S_w / d2S \rightarrow d2S = S_w / (2 * tan(FoV_w / 2)$

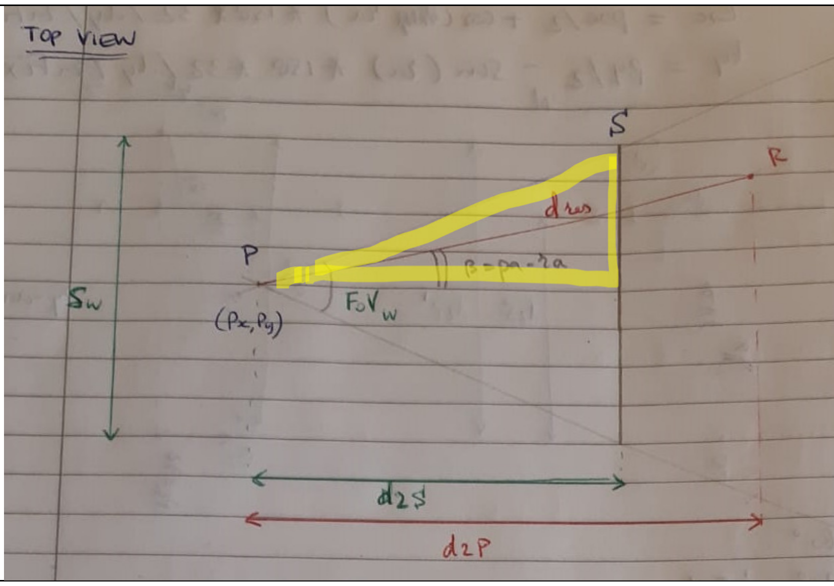


Diagram 3

<u>Step 4 – applying this in the texture coordinates calculation</u>

We now have all the info to work out the texture coordinates.

We have the ray angle in radians (a bit stupid of 3DSage to call this one 'deg' though):

```
deg = degToRad( ra ),
```

And we have the cosine of the difference angle $\beta = p_a - r_a$:

```
raFix = cos(degToRad( FixAng( pa - ra )));
```

The formula for the x and y texture coordinates are:

```
// determine texture coordinates
tx = px / 2.0f + cos( deg ) * 158 * 32 / dy / raFix;
ty = py / 2.0f - sin( deg ) * 158 * 32 / dy / raFix;
```

1. Starting point is the players location divided by 2. This division by two is necessary since the tile size is 64 units and the texture size is 32. So this division yields the players location expressed in texture coordinates.
2. Added to the players location is an offset using the ray angle (variable deg) multiplied by a formula that represents the length d_{res} (`158 * 32 / dy / raFix`):
   a. 32 represents the player height (a cell is considered 64 x 64 x 64 units in size, and point of view of the player is halfway the cell height);
   b. dy is calculated previously as …

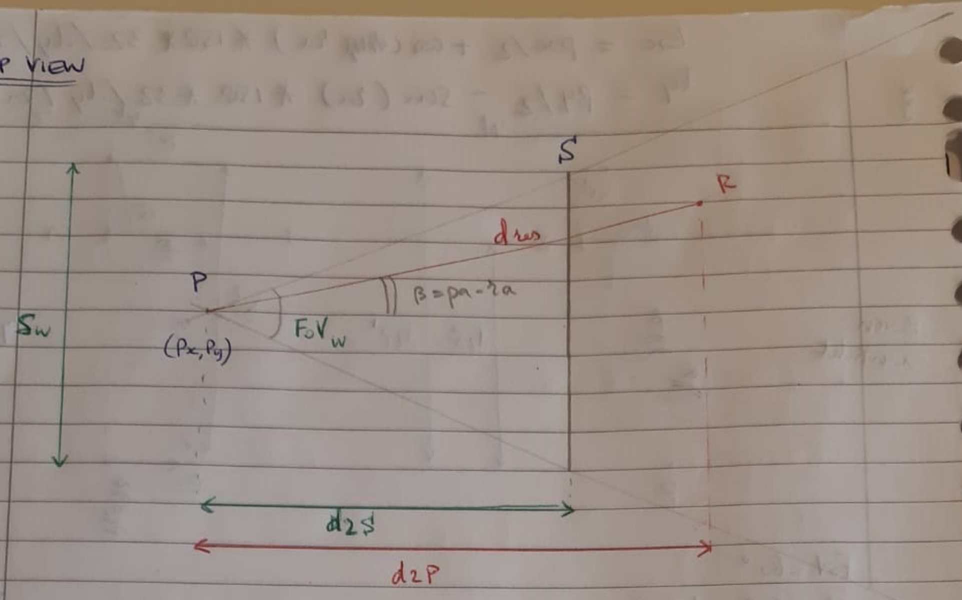
   ```
   dy = y - (320 / 2.0f),
   ```

   … where 320 is the screen height, and y is the pixel in screen coordinates we are currently considering.

The ratio P_h over dy is used to extend d2S to d2P

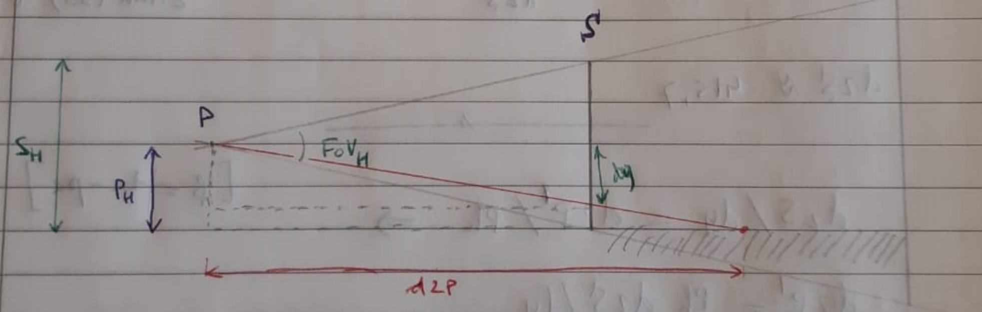
c. raFix represents the compensation factor to get the length difference between d2P and d_{res}, using the difference angle
d. This leaves 158 as the d2S value, the distance between the player and the projection screen

TOP VIEW



$d_{res} \cdot \cos(pa - ra) = d2P \Rightarrow d_{res} = \frac{d2P}{\cos(pa - Ra)}$

SIDE VIEW



$$\frac{dy}{d2S'} = \frac{P_H}{d2P} \Rightarrow d2P = \frac{P_H \cdot d_2S'}{dy}$$

$$d_{res} = \frac{d2P}{\cos(pa - ra)} = \frac{P_H \cdot d_2S'}{dy \cdot \cos(pa - ra)}$$