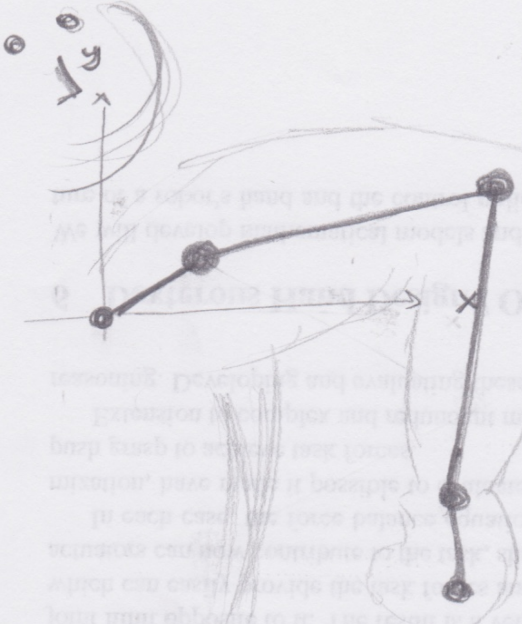
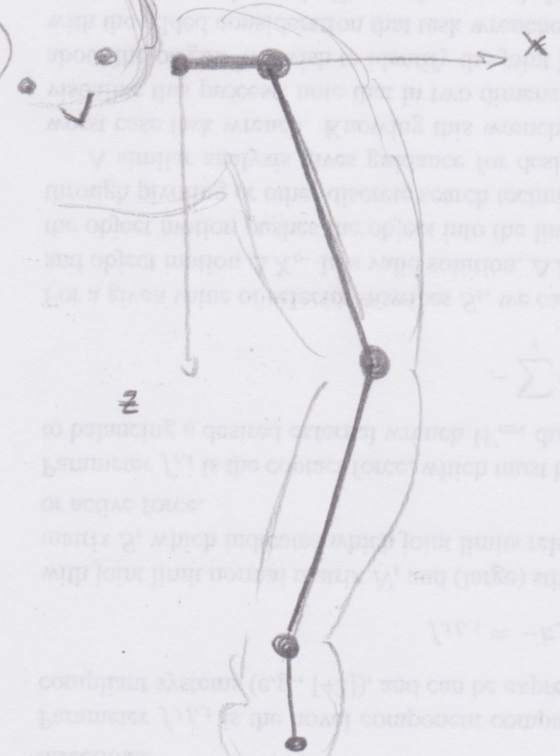
SOLUTIONS

1a.)



1b.)



1c.) $R_z(\theta_z)$

$$\boxed{\begin{bmatrix} C\theta_z & -S\theta_z & 0 & 0 \\ S\theta_z & C\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

1d.) $T(t_x, t_y, t_z)$

$$\boxed{\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}$$

1e.) $\boxed{P' = T(4,2,0)\,R_S\,T(2,-10,0)\,R_E\,T(-2,-10,0)\,R_W\,P}$

where $R_S = R_z(\Theta_{sz})\,R_x(\Theta_{sx})\,R_y(\Theta_{sy})$

$$R_E = R_z(\Theta_{ez})\,R_x(\Theta_{ex})\,R_y(\Theta_{ey})$$

$$R_W = R_z(\Theta_{wz})\,R_x(\Theta_{wx})\,R_y(\Theta_{wy})$$

1f.) $\boxed{\begin{aligned}\Theta_{sz} &= 0\\ \Theta_{sx} &= \tfrac{\pi}{2}\\ \Theta_{sy} &= 0\end{aligned}}$

Any solution with $\Theta_{sx} = \pm\frac{\pi}{2}$ and $\Theta_{sz} = 0$ is fine

1g.) $\boxed{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}}$

(The exact matrix here depends on the values chosen in 1f)

1h.) It is the $\boxed{\text{y-axis}}$

Incremental changes in $\underline{\Theta_{sx}}$ result in rotation about $\underline{\text{world } x}$

" $\underline{\Theta_{sy}}$ " $\underline{\text{world } z}$

" $\underline{\Theta_{sz}}$ " $\underline{\text{world } z}$

Rotate by ε about y

1i) $$\begin{bmatrix} c\epsilon & 0 & s\epsilon \\ 0 & 1 & 0 \\ -s\epsilon & 0 & c\epsilon \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \cos(\epsilon) & \sin(\epsilon) & 0 \\ 0 & 0 & -1 \\ -\sin(\epsilon) & \cos(\epsilon) & 0 \end{bmatrix}$$

1j) Intuitively, we will have to rotate first about some axis other than y, at least a very tiny bit, in order to kick the joint out of gimbal lock.

1k.) The interesting thing for animation is that there is no real problem for standard keyframing that is associated specifically with gimbal lock. We can create arbitrary keyframes and set them. Even interpolation in this case will produce visually pleasing results. Problems occur with odd interpolation because linearly interpolating Euler angles does not linearly interpolate rotations, although this is not specifically a gimbal lock problem. Gimbal lock causes other problems, however, such as trackballs and orientation widgets that get "stuck".

1l.) If we solve for the Euler angles corresponding to the matrix in 1i) we get: (not a unique solution)

<u>Before</u>

$\Theta_y = 0$

$\Theta_x = \frac{\pi}{2}$

$\Theta_z = 0$

<u>After</u>

$\Theta_y = -\frac{\pi}{2}$

$\Theta_x = \frac{\pi}{2} + \epsilon$

$\Theta_z = \frac{\pi}{2}$

We can create an interpolation from one set to the next and investigate what happens at the beginning of that interpolation.

Conclusion: <u>Initial rotation is about $\hat{x}$</u>

$$R_z\left(\tfrac{\pi}{2}t\right) \qquad R_x\left(\tfrac{\pi}{2}+\epsilon t\right) \qquad R_y\left(-\tfrac{\pi}{2}t\right)$$

$$\begin{bmatrix} c(\frac{\pi}{2}t) & -s(\frac{\pi}{2}t) & 0 \\ s(\frac{\pi}{2}t) & c(\frac{\pi}{2}t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\frac{\pi}{2}+\epsilon t) & -s(\frac{\pi}{2}+\epsilon t) \\ 0 & s(\frac{\pi}{2}+\epsilon t) & c(\frac{\pi}{2}+\epsilon t) \end{bmatrix} \begin{bmatrix} c(-\frac{\pi}{2}t) & 0 & s(-\frac{\pi}{2}t) \\ 0 & 1 & 0 \\ -s(-\frac{\pi}{2}t) & 0 & c(-\frac{\pi}{2}t) \end{bmatrix}$$

$$\begin{bmatrix} c(\frac{\pi}{2}t) & -s(\frac{\pi}{2}t)c(\frac{\pi}{2}+\epsilon t) & s(\frac{\pi}{2}t)s(\frac{\pi}{2}+\epsilon t) \\ s(\frac{\pi}{2}t) & c(\frac{\pi}{2}t)c(\frac{\pi}{2}+\epsilon t) & -c(\frac{\pi}{2}t)s(\frac{\pi}{2}+\epsilon t) \\ 0 & s(\frac{\pi}{2}+\epsilon t) & c(\frac{\pi}{2}+\epsilon t) \end{bmatrix} \begin{bmatrix} c(+\frac{\pi}{2}t) & 0 & -s(+\frac{\pi}{2}t) \\ 0 & 1 & 0 \\ +s(+\frac{\pi}{2}t) & 0 & c(+\frac{\pi}{2}t) \end{bmatrix}$$

$$\begin{bmatrix} +c^2(\frac{\pi}{2}t) + s^2(\frac{\pi}{2}t)s(\frac{\pi}{2}+\epsilon t) & -s(\frac{\pi}{2}t)c(\frac{\pi}{2}+\epsilon t) & -c(\frac{\pi}{2}t)s(\frac{\pi}{2}t)\left(1-s(\frac{\pi}{2}+\epsilon t)\right) \\ c(\frac{\pi}{2}t)s(\frac{\pi}{2}t)\left(1-s(\frac{\pi}{2}+\epsilon t)\right) & c(\frac{\pi}{2}t)c(\frac{\pi}{2}+\epsilon t) & -s^2(\frac{\pi}{2}t) - c^2(\frac{\pi}{2}t)s(\frac{\pi}{2}+\epsilon t) \\ s(\frac{\pi}{2}t)c(\frac{\pi}{2}+\epsilon t) & s(\frac{\pi}{2}+\epsilon t) & c(\frac{\pi}{2}t)c(\frac{\pi}{2}+\epsilon t) \end{bmatrix}$$

What happens to $\hat{x}$

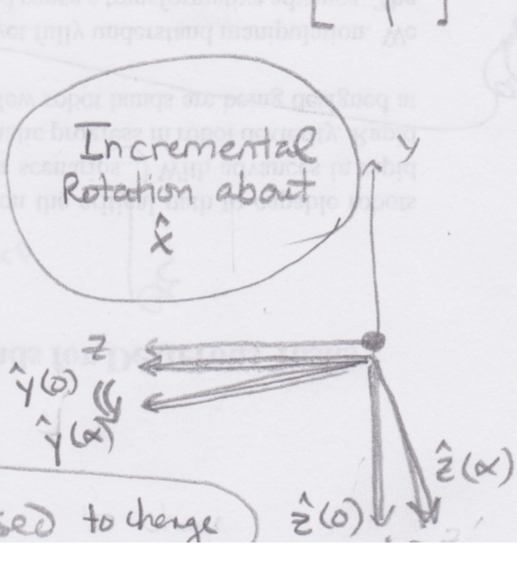
$$\hat{x}(0) = \begin{bmatrix} +1 \\ 0 \\ 0 \end{bmatrix} \qquad \hat{x}(1) = \begin{bmatrix} c\epsilon \\ 0 \\ -s\epsilon \end{bmatrix} \qquad \underset{\text{(Small)}}{\hat{x}(\alpha)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

What happens to $\hat{y}$

$$\hat{y}(0) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \hat{y}(1) = \begin{bmatrix} s\epsilon \\ 0 \\ c\epsilon \end{bmatrix} \qquad \underset{\text{small}}{\hat{y}(\alpha)} = \begin{bmatrix} 0 \\ -\beta \\ 1 \end{bmatrix}$$

What happens to $\hat{z}$

$$\hat{z}(0) = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \qquad \hat{z}(1) = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \qquad \underset{\text{small}}{\hat{z}(\alpha)} = \begin{bmatrix} 0 \\ -1 \\ -\beta \end{bmatrix}$$

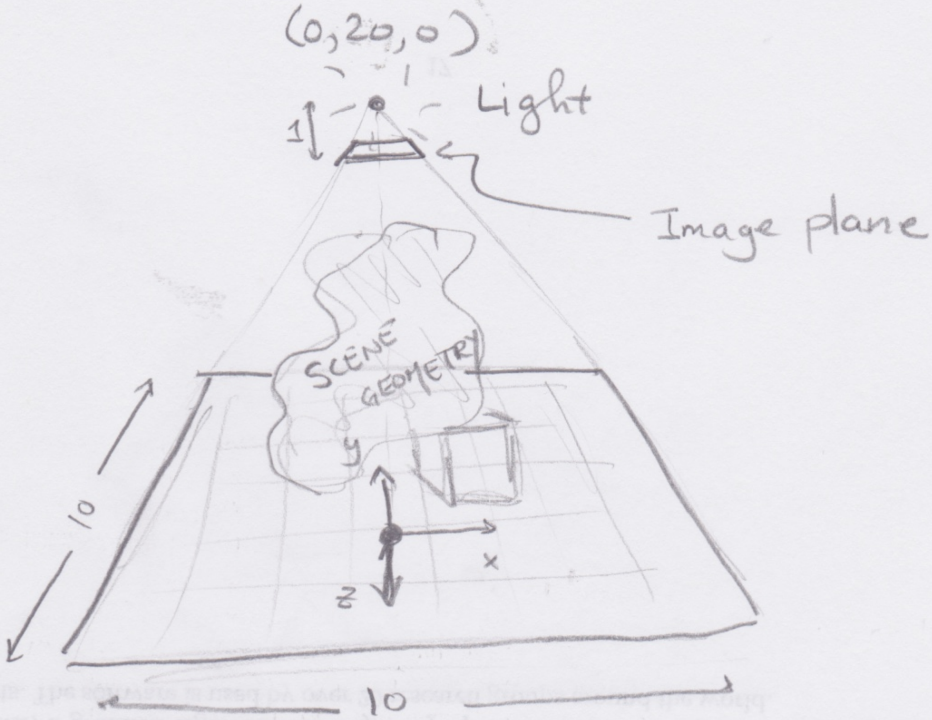


Small angle approximation

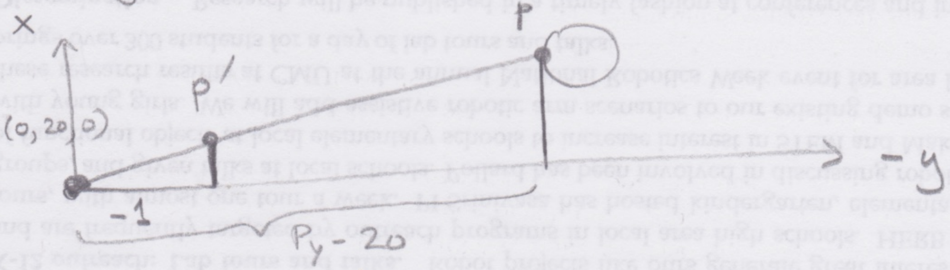
$$c(\tfrac{\pi}{2}t) = 1$$
$$s(\tfrac{\pi}{2}t) = \beta$$
$$c(\tfrac{\pi}{2}+\epsilon t) = -\beta$$
$$s(\tfrac{\pi}{2}+\epsilon t) = 1$$

(Note that $\hat{z}$ was not supposed to change)

2a.)



2b.) Light point of view



$$\frac{P'_x}{1} = \frac{P_x}{(20-P_y)} \qquad P'_x = \frac{P_x}{(20-P_y)}$$ image coord u

$$\frac{P'_z}{1} = \frac{P_z}{(20-P_y)} \qquad P'_z = \frac{P_z}{(20-P_y)}$$ image coord V

$$\boxed{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 20 \\ 0 & -1 & 0 & 20 \end{bmatrix}}$$

Some variations are also OK

- switch top 2 rows
- negate first and/or second row
- negate last 2 rows

2c) Map $y = 19$ to $z = 1$

$y = 0$ to $z = 20$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & a & 0 & b \\ 0 & -1 & 0 & 20 \end{bmatrix} \begin{bmatrix} 0 \\ 19 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 19a+b \\ 1 \end{bmatrix}$$

$z = 1 \Rightarrow$

$19a + b = 1$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & a & 0 & b \\ 0 & -1 & 0 & 20 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b \\ 20 \end{bmatrix}$$

$z = 20 \Rightarrow$

$\frac{b}{20} = 20$

$b = 400$

$19a + 400 = 1$

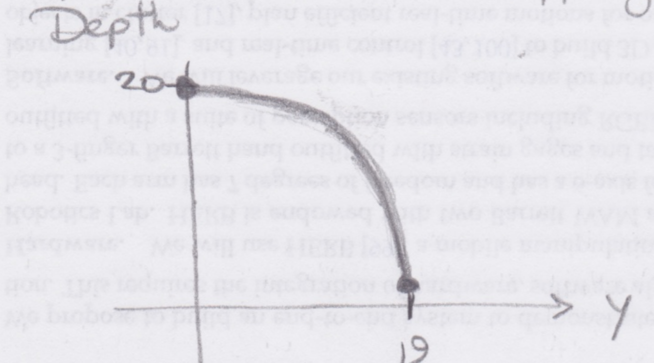
$19a = -399$

$a = -21$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -21 & 0 & 400 \\ 0 & -1 & 0 & 20 \end{bmatrix}$$

$$M \begin{bmatrix} 0 \\ 9.5 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -21(9.5) + 400 \\ -9.5 + 20 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 199.5 \\ 10.5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 19 \\ 1 \end{bmatrix}$$

My matrix has a mapping that look like this:



It is especially bad!

Most of the depth resolution is for items w/in 5 units of the light.

2d.) u and v are coordinates for the "image plane" when we treat the light source as a camera. For my matrix, u=0, v=0 falls in the center of the image (maybe not what we want).

2e.) Use the transform M computed for the light to find u,v from world coordinates g.

2f.) $$M \begin{bmatrix} g_x \\ g_y \\ g_z \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha u \\ \alpha v \\ \alpha w \\ \alpha \end{bmatrix} \xRightarrow{1/\alpha} \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} \qquad \boxed{M}$$

2g.) $\underline{\underline{w}}$ computed as in 2f is greater than the value stored at (u, v) in the shadow map

2h.) Computation of u and v can be done at the vertex level and interpolated in screen space using the perspective-correct interpolation we saw in class. This is true because u and v vary linearly over the triangle.

What about w?

In my example (part 2c), $w = \dfrac{(-21y + 400)}{(-y + 20)}$.

This is clearly non-linear across the triangle. However, both numerator and denominator are linear. Therefore, we could track numerator and denominator separately, interpolate in screen space using perspective-correct interpolation, and divide them per-fragment to get the correct answer.