# Slitscan Project Outline

By Monica Quinlan

# Outline of Project

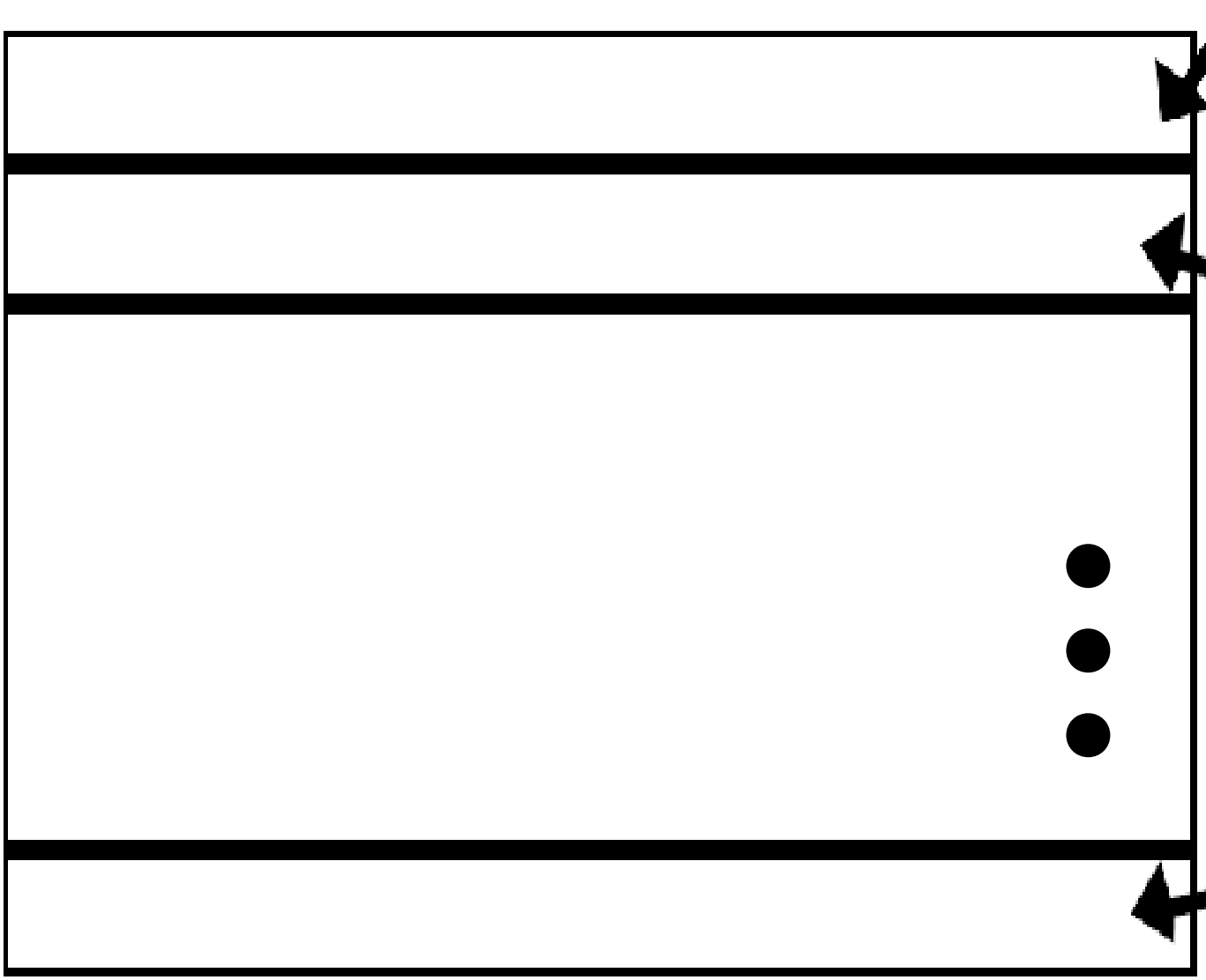
- Based on Zbig Rybczynski's experimental film, The Fourth Dimension
- Each frame is divided into 240* horizontal slices, 1 pixel high
- The highest slice (y=0) is in real time. The next slice down has a delay of one frame, and so on. 240 frames later, the lowest slice gets to the first frame taken.

*assuming video input of 240x320 pixels

# The Fourth Dimension (Rybczynski's method and demonstration of aesthetics)

- Rybczynski's description of his work: “I shot all the actors in their sets with all the movements, then in the printing phase I visualized the image in 480 lines and reproduced the images delaying, for example, each frame by one line. Thus, the last line ended up being 480 images later in respect to the higher one, so that when the head of a character is rotated, his feet are still in their original position.”

# Diagram of Time Delay



Slice 1: taken from current frame

Slice 2: taken from previous frame

Slice 240: frame delay, 239 frames behind current

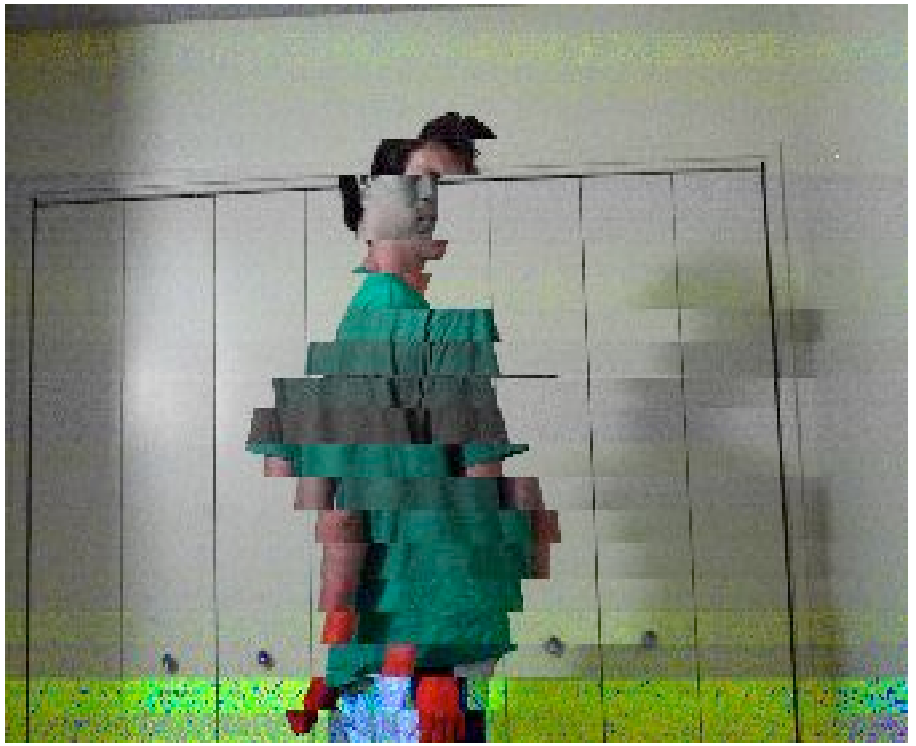
# Example of Time Delay

- Manual representation of frame delay
- Note: with smaller (1 px high) slices, the transitions between slices will appear smoother

# Variations + Interactivity

- Variations in recording will occur, based on interactive input
- One interesting variation is slice height, as it disrupts the smooth transitions of the initial code

# Variations + Interactivity



- I would also like to try variations in saturation of the individual slices
- There are many other variations of interest

# Interactivity

- There are a number of ways to implement interactivity that affects the height/saturation of each slice, based on direct input or the environment.
- Some to experiment with: color, movement (using frame differencing), sound (both pitch and volume), and mouse position for online work
- Ideally, this would be realized as an installation

# Implementing Variations

- An intermediary step might be a uniform change of slice height for each slice in the frame

# Uniform Slice Width (examples)

<< two slices

four slices >>

<< many slices

one slice >>

# Steps (figure out how to...)

- Read camera input
- Copy image to screen
- Buffer [store] images

*Store as stack or line-by-line?*

- Record
- Display [the tricky part] each frame as composite of slices

*This section needs to be further broken down*

- Produce variations

*Set as variable*

- Add interaction element

*Assign value to variable, based on interaction*