

Laser Conductor

(or DDR with your hands...)

James Noraky and Scott Skirlo

Introduction



Introduction



Introduction

BROKE



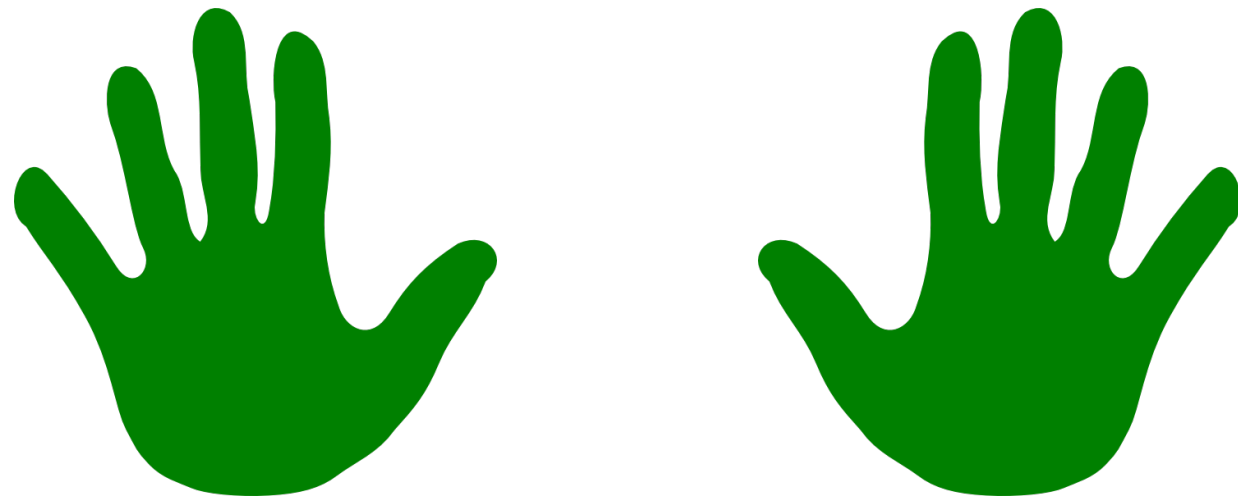
Introducing Laser Conductor

BROKE

Interactive game that combines music, lasers, and video processing as you play as a conductor!



Laser Conductor Game Play



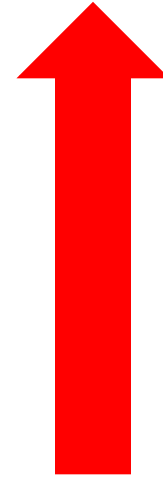
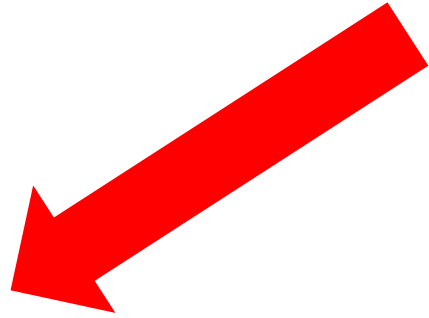
Laser Conductor Game Play



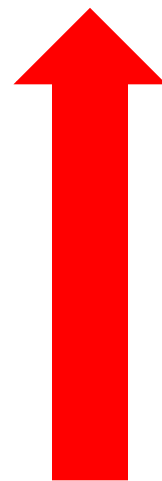
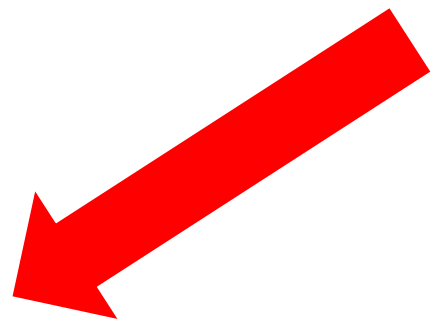
Laser Conductor Game Play



Laser Conductor Game Play



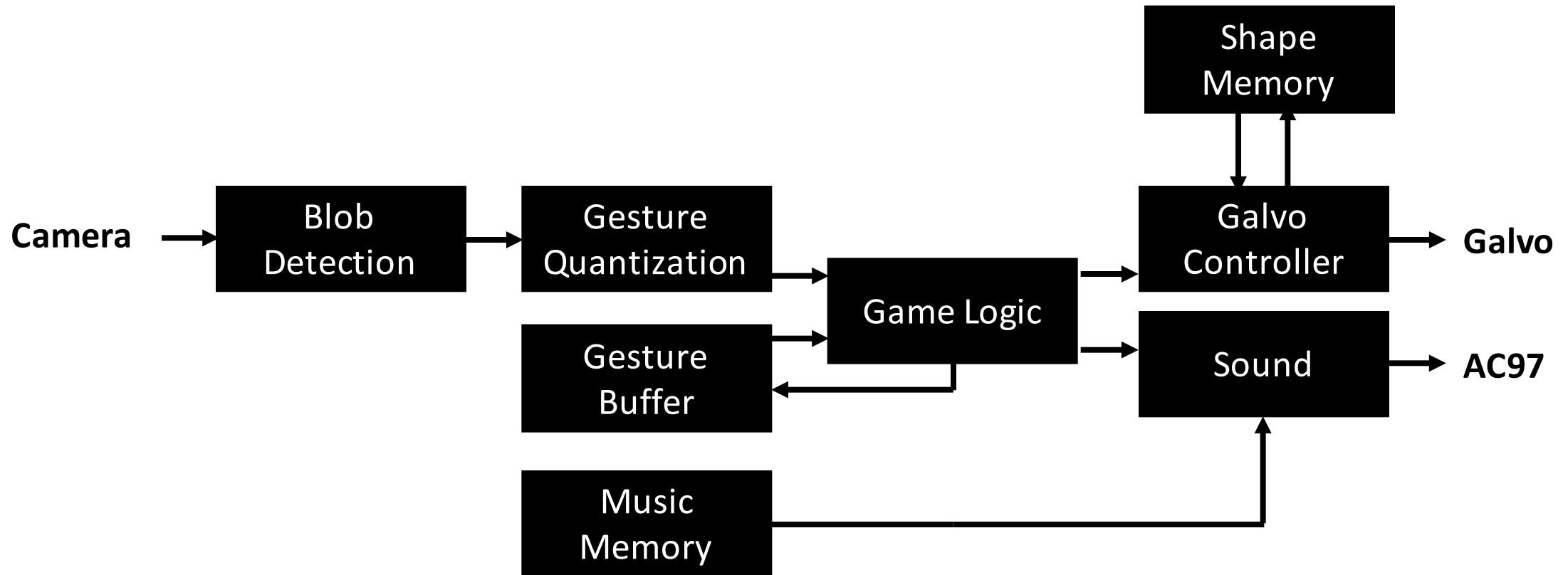
Laser Conductor Game Play



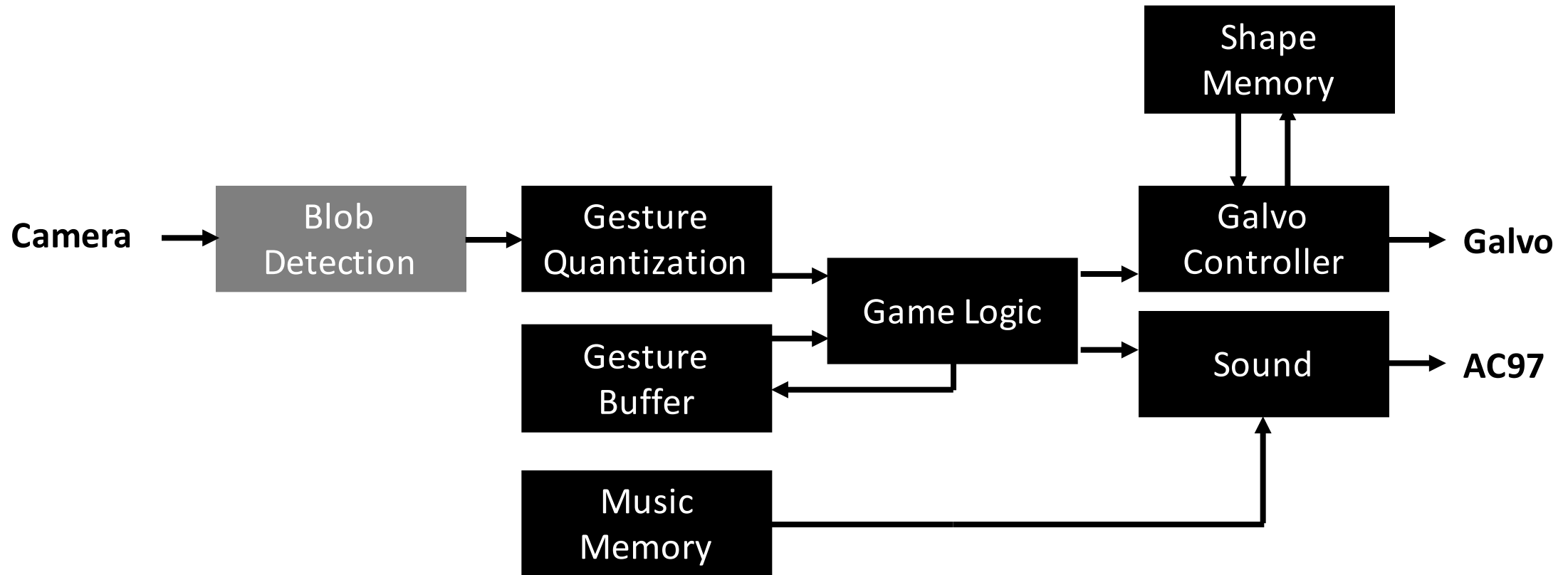
Laser Conductor Game Play



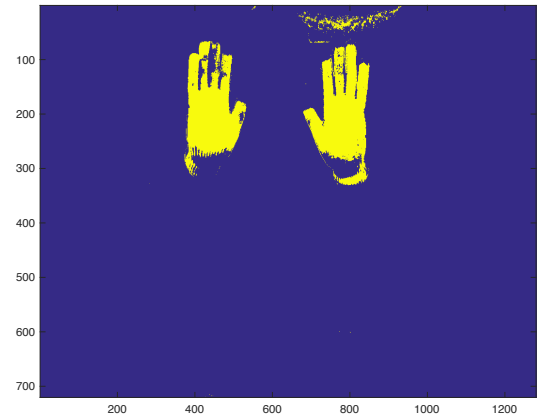
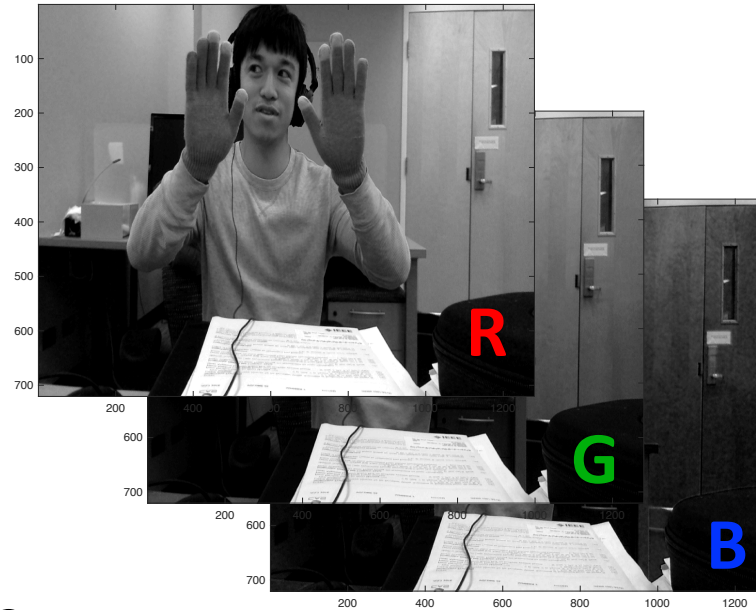
Block Diagram



Block Diagram



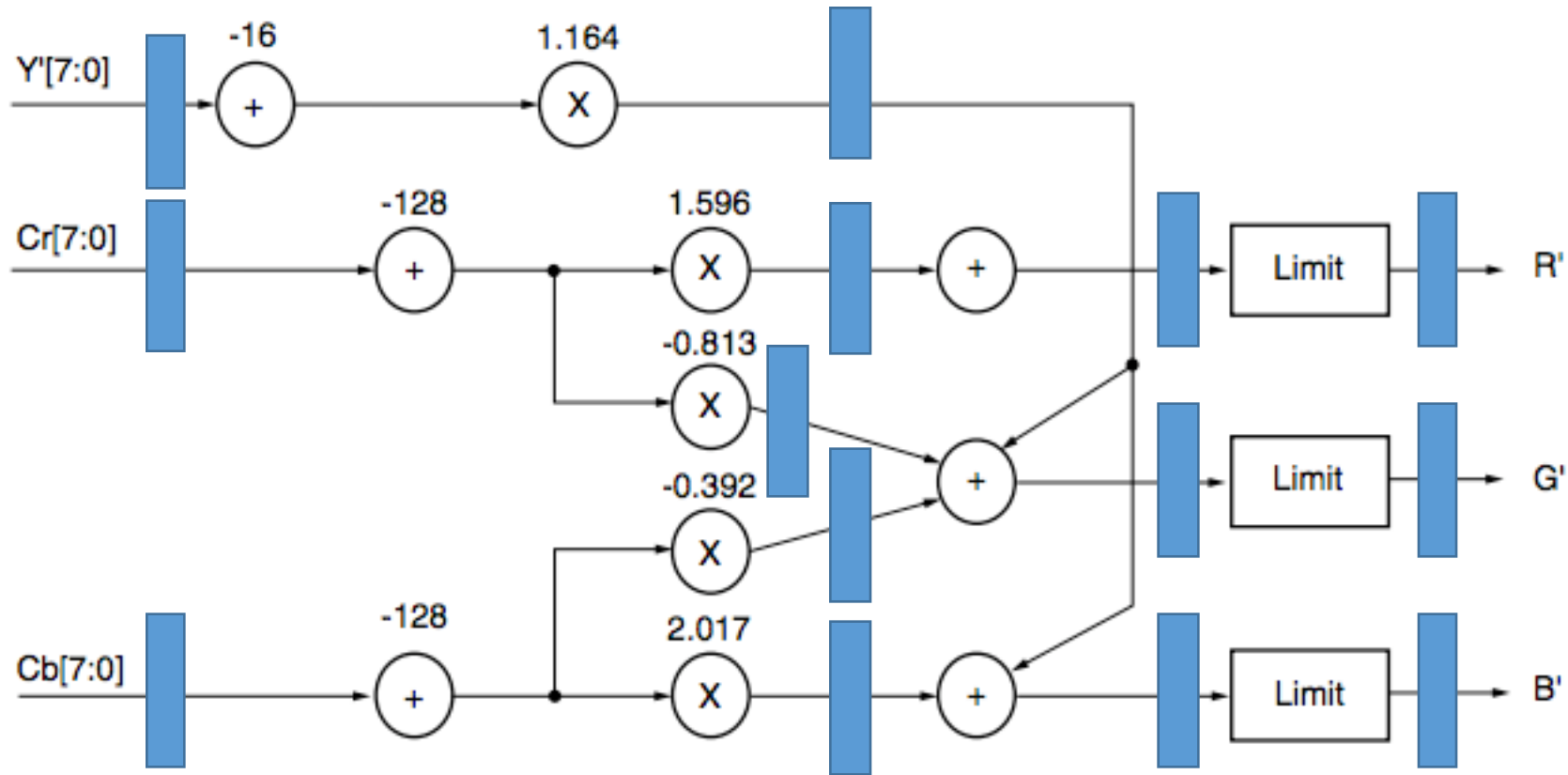
Blob Detection



Convert YCbCr
pixel from the
decoder to RGB

Threshold color
channels to get
binary image

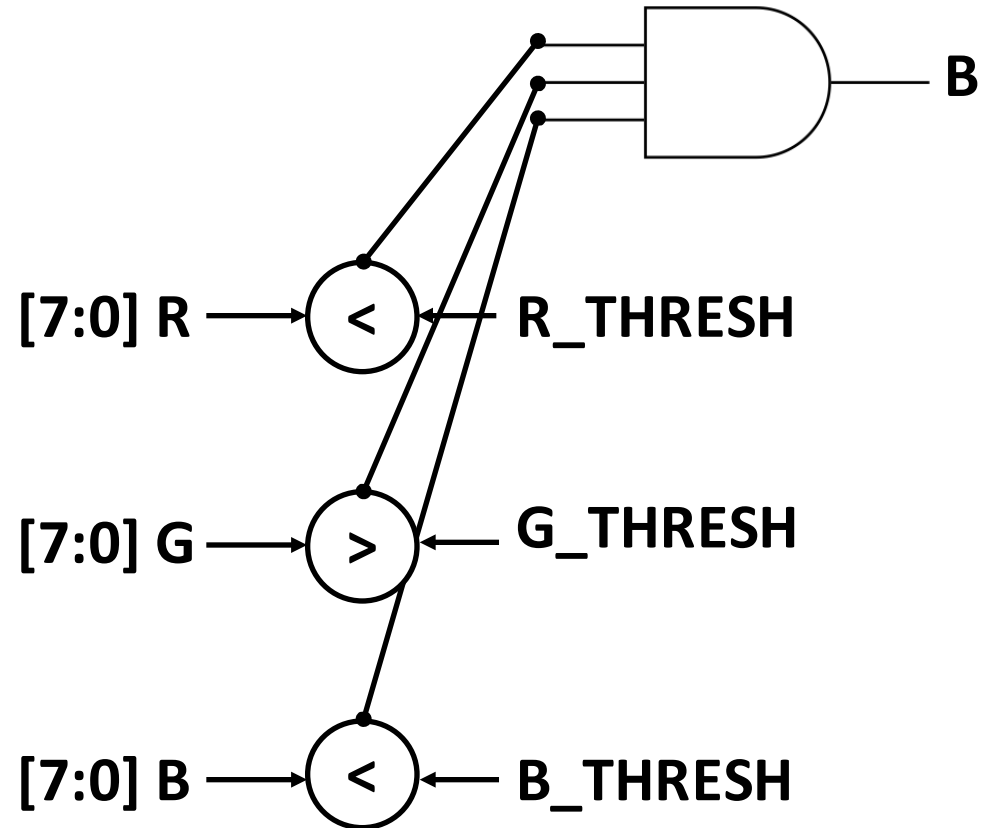
Converting YCbCr to RGB



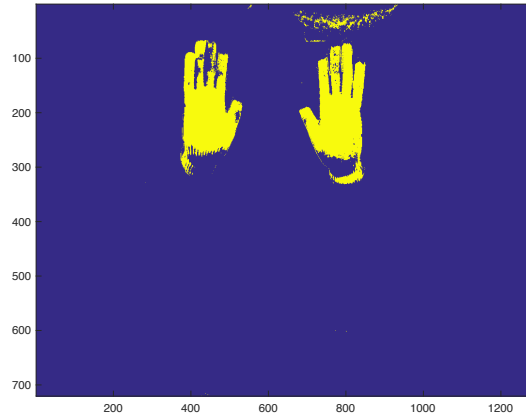
 Pipeline Register

Threshold to Obtain Binary Image

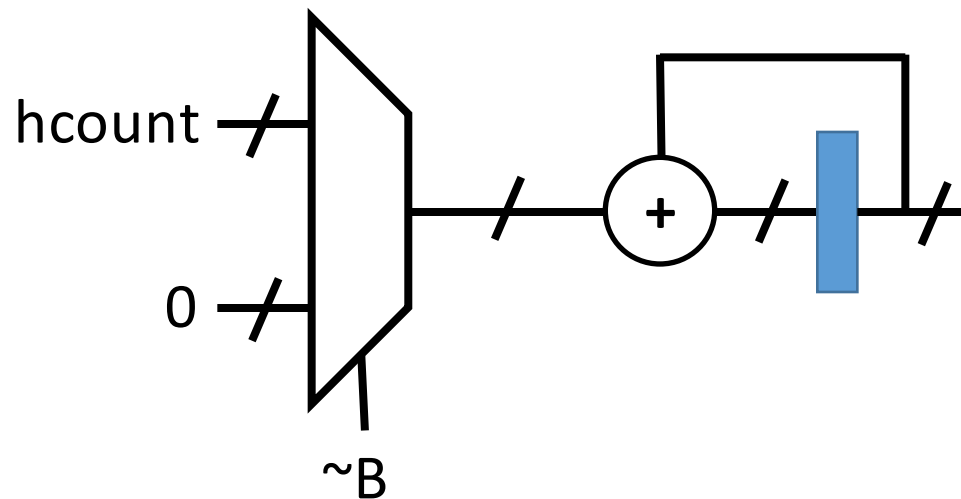
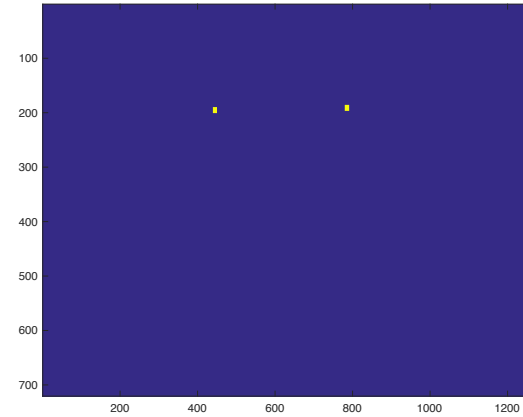
- High threshold for green and low threshold for red and blue
 - Pass green but reject white
- Compression! – [23:0] RGB is represented as a single bit!
- Even more compression! – Sequence of bits can be compressed using run length encoding



Centroid Calculation

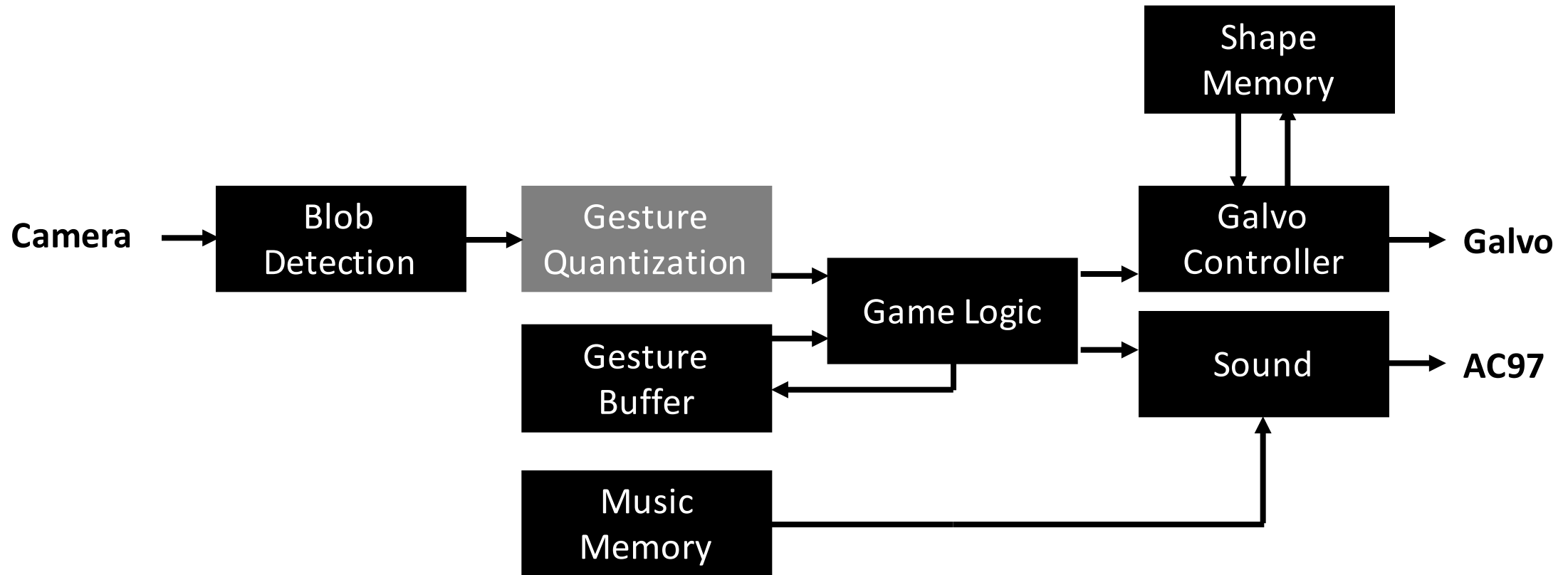


Get centroid locations



Additional circuitry and logic required to avoid normalizing the sum

Block Diagram



Gesture Classification

- Many algorithmic approaches from Hidden Markov Models, Neural Nets, Convex Hull estimation
- Challenges: (1) Computationally expensive (2) Requires a lot of data to train

Gesture Classification

- Many algorithmic approaches from Hidden Markov Models, Neural Nets, Convex Hull estimation
- Challenges: (1) Computationally expensive (2) Requires a lot of data to train

Simple solution! - Track the trajectory of the centroids

Gesture Classification Pseudocode

(x, y)



(x_start, y_start)

Gesture Classification Pseudocode

```
curr_x <= x-x_start;
```

```
curr_y <= y-y_start;
```

(x, y)

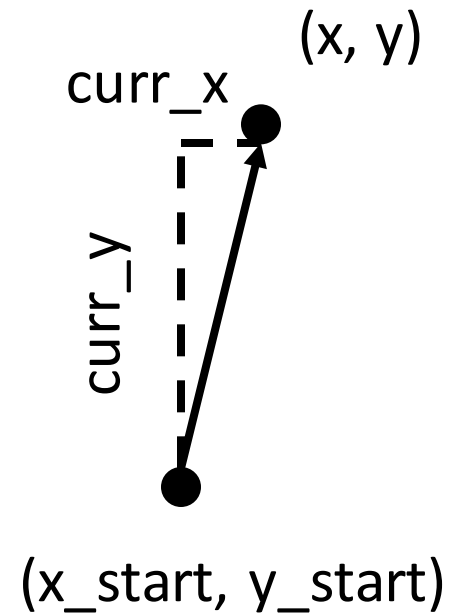


(x_start, y_start)

Gesture Classification Pseudocode

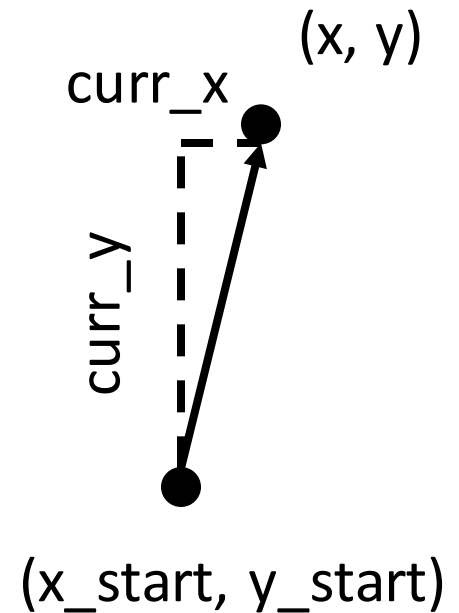
```
curr_x <= x-x_start;
```

```
curr_y <= y-y_start;
```

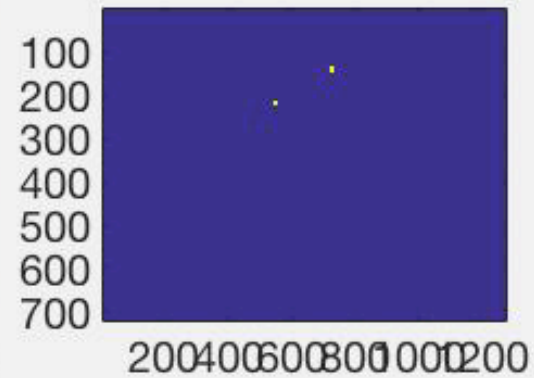
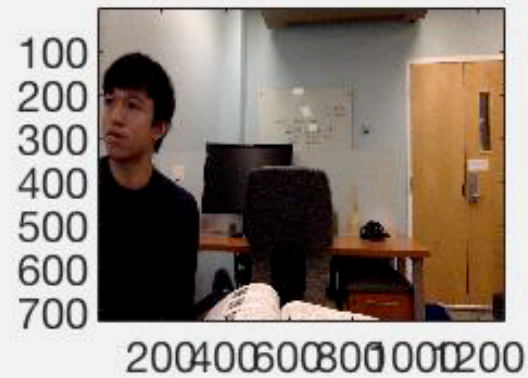
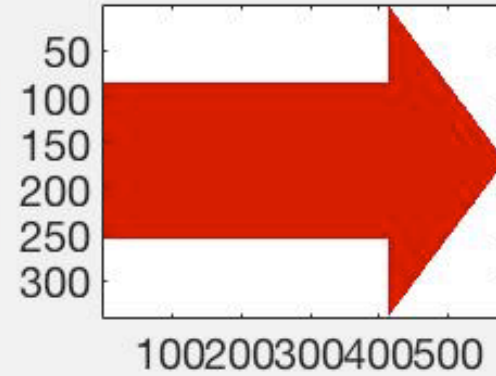
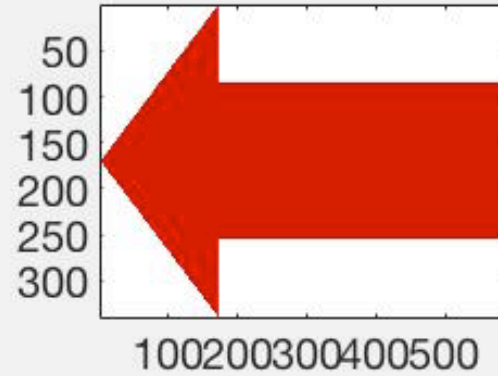


Gesture Classification Pseudocode

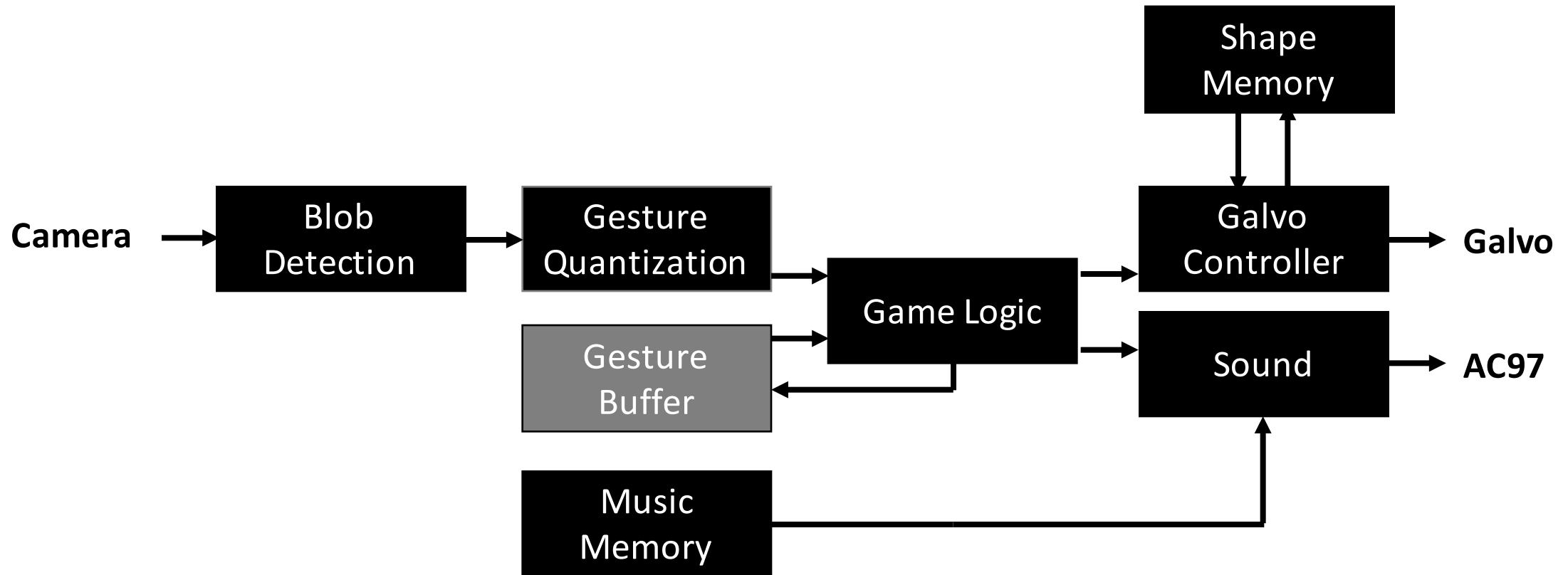
```
curr_x <= x-x_start;  
curr_y <= y-y_start;  
  
if ( (curr_y > 0) &&  
    ( (curr_x < thresh1) && (curr_x > thresh2))  
begin  
    gesture <= UP;  
end  
//... continue for other gestures
```



Matlab Demo

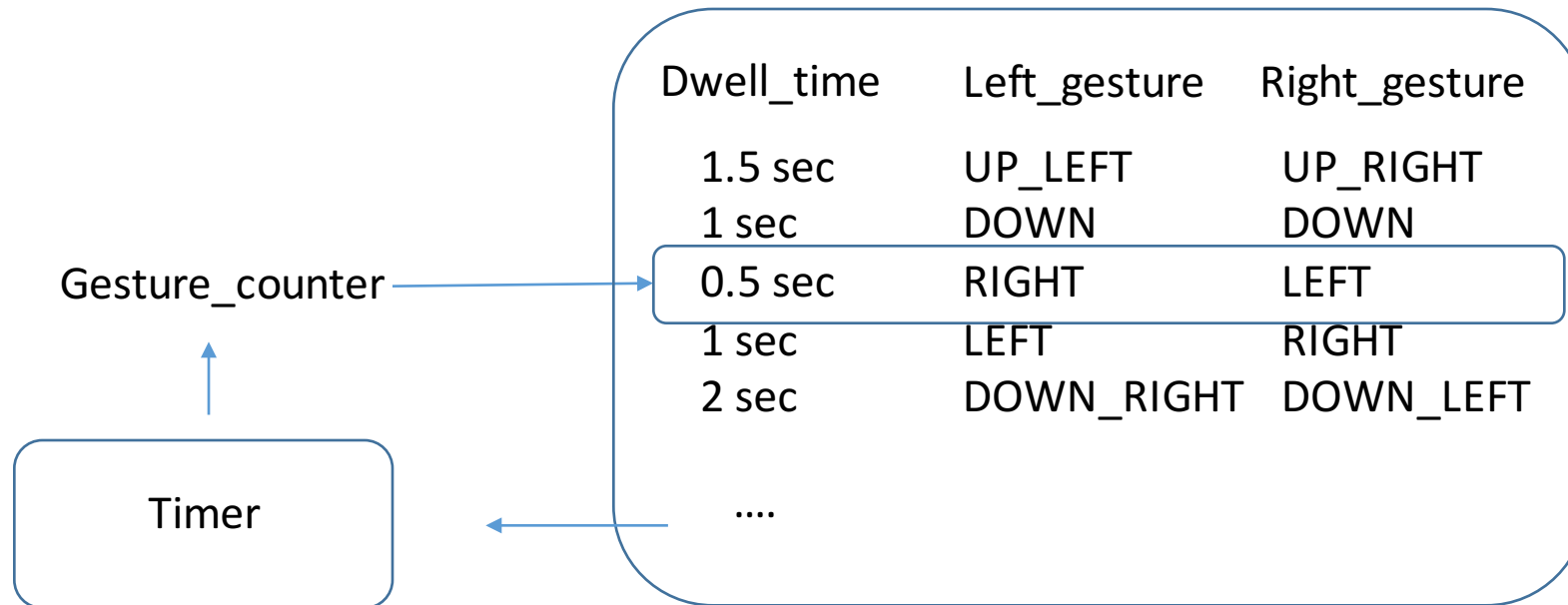


Block Diagram

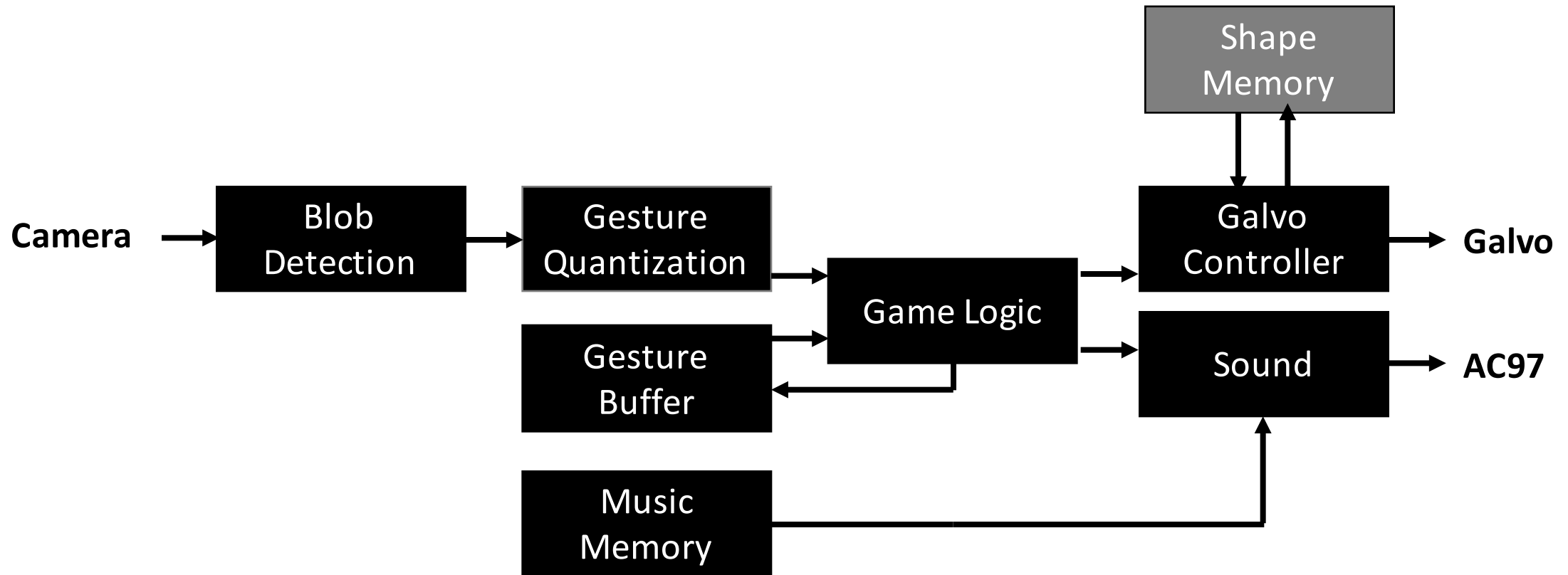


Gesture Buffer

- Stores a sequence of gestures to be recognized in DRAM and their dwell time (in 0.5 sec increments), @(posedge timer) update

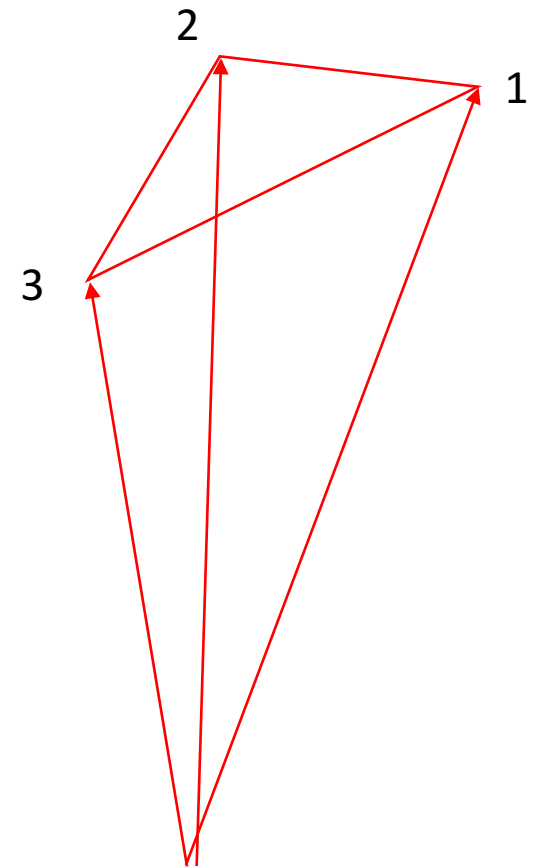
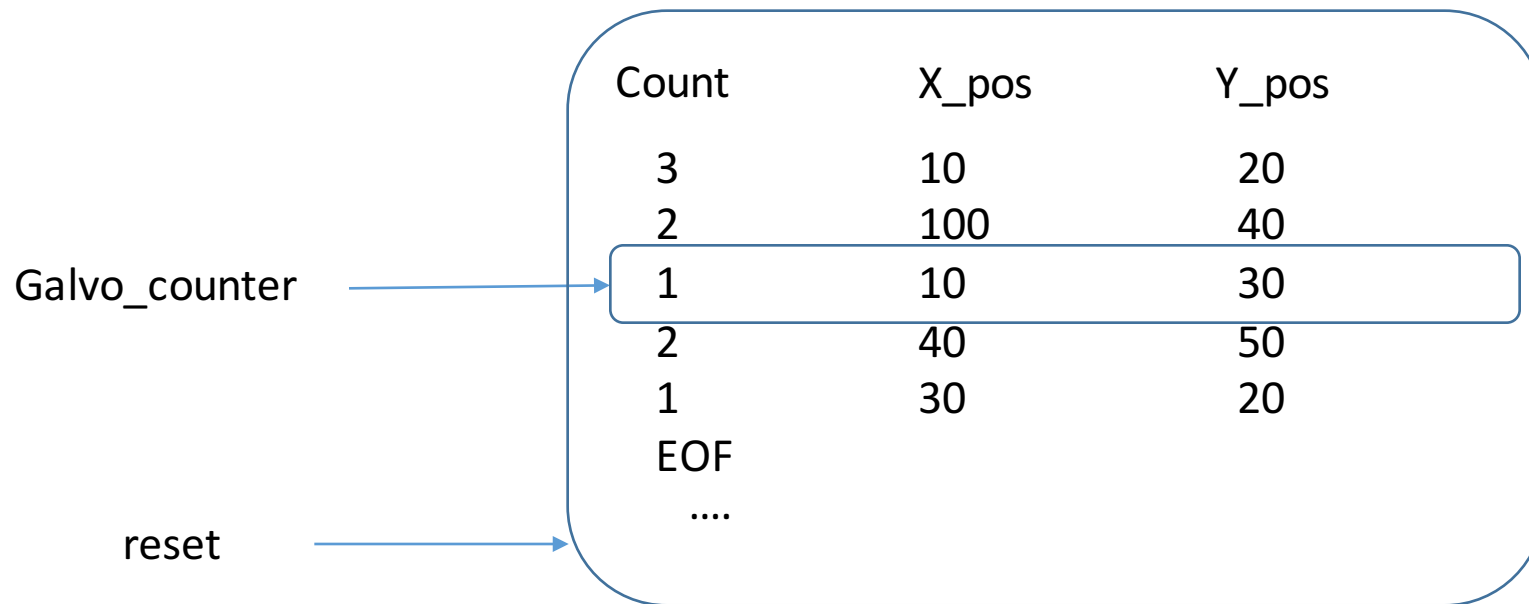


Block Diagram

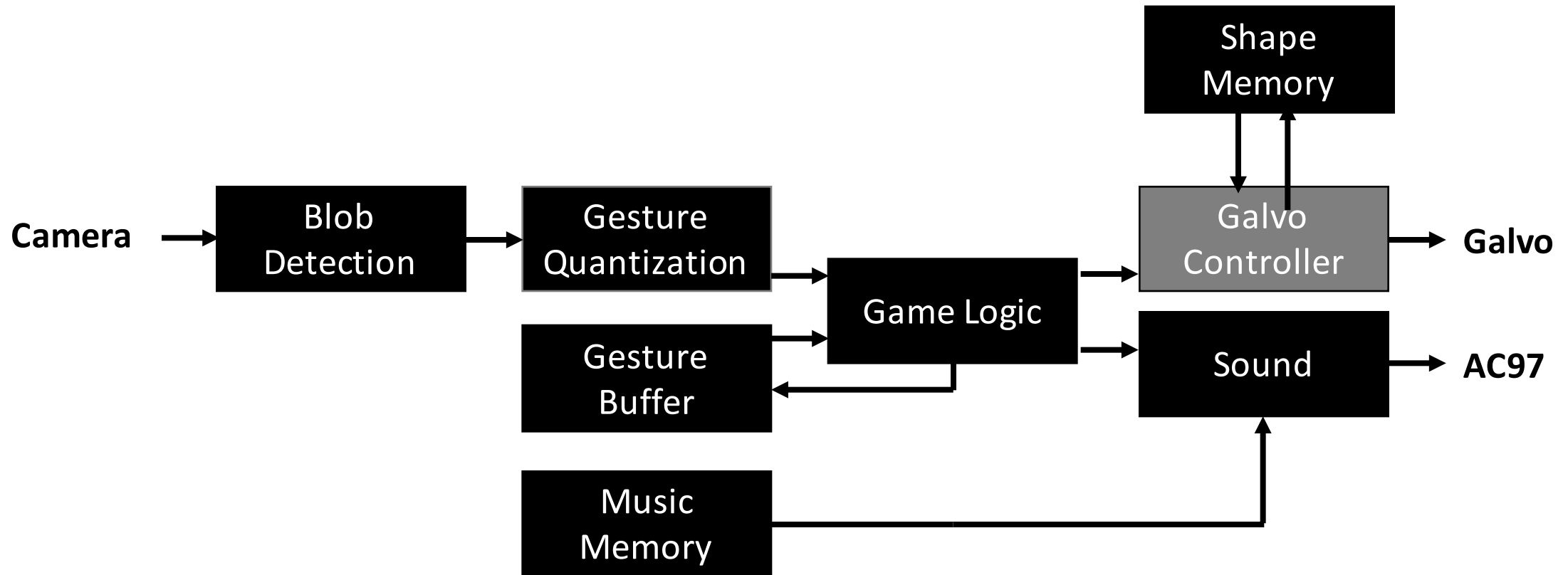


Shape memory

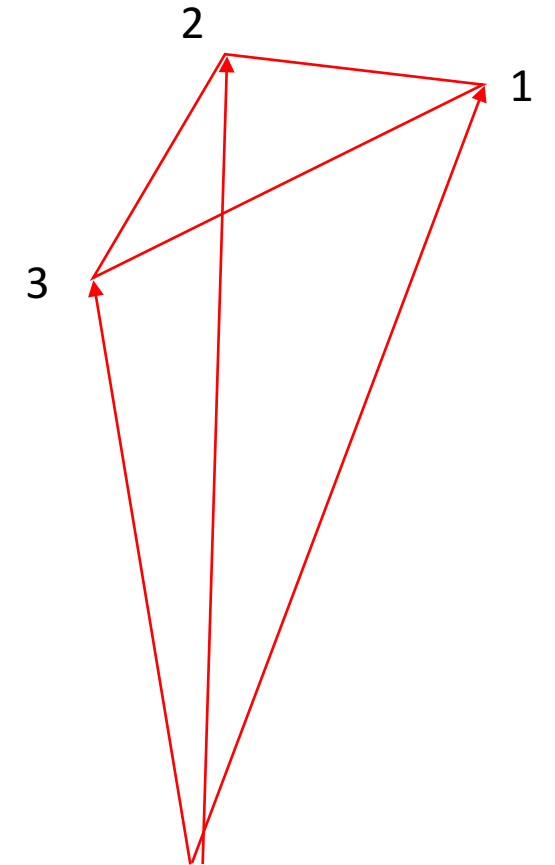
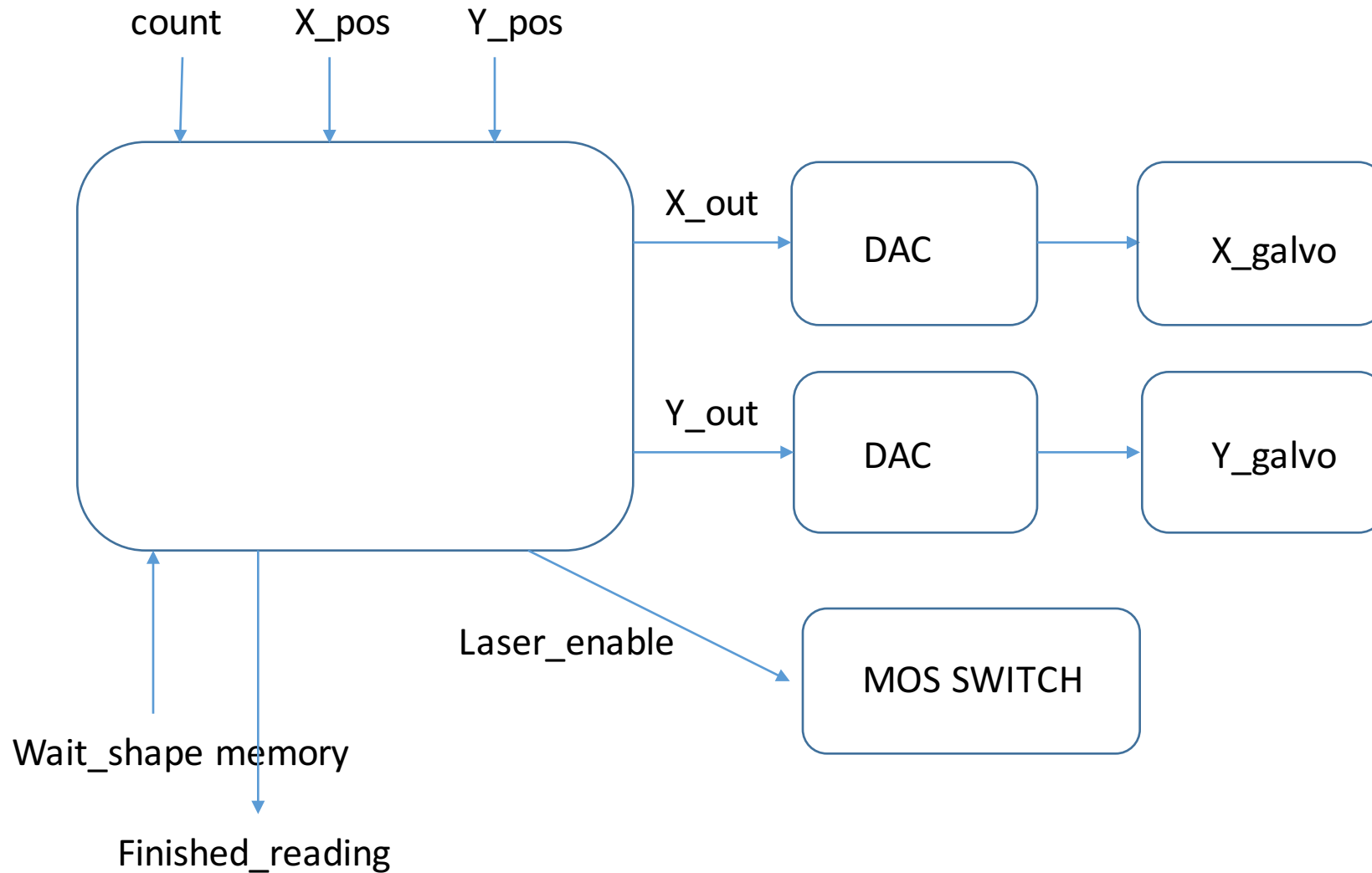
- Stores a sequence of points for plotting vector graphics



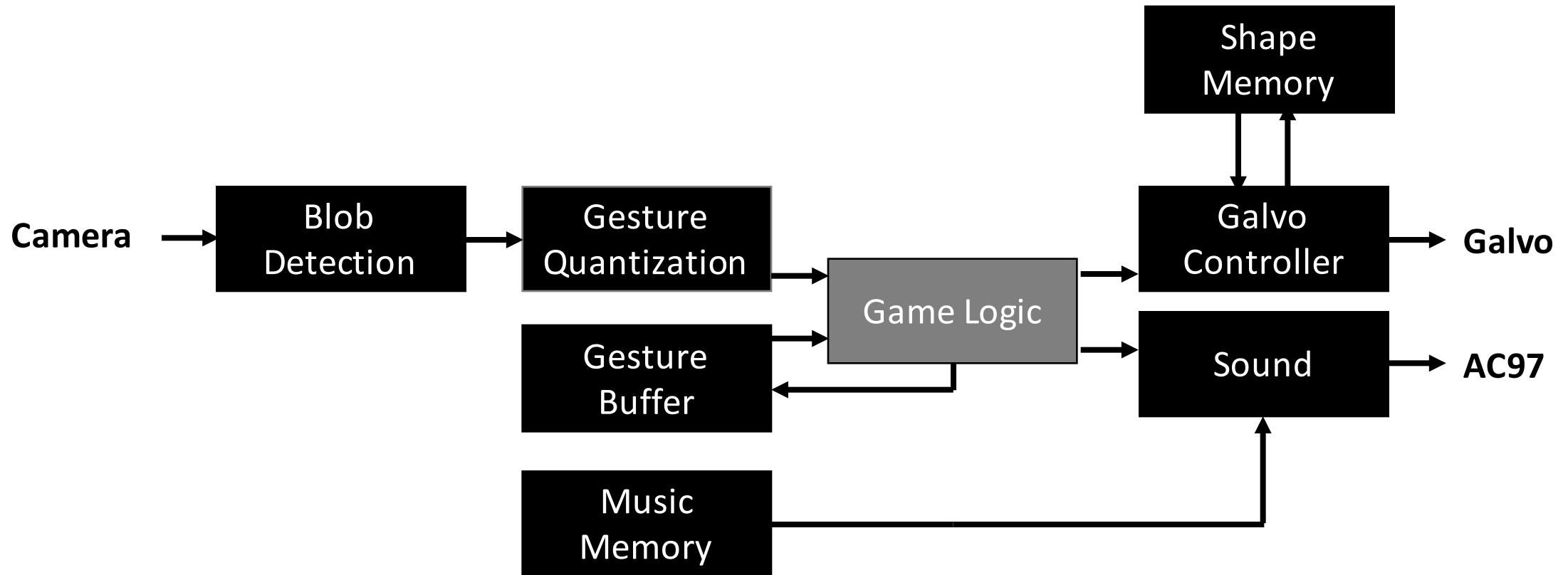
Block Diagram



Galvo controller



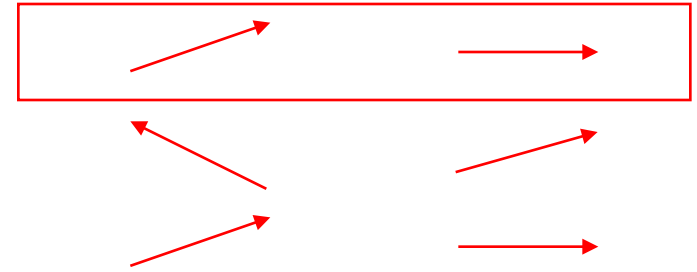
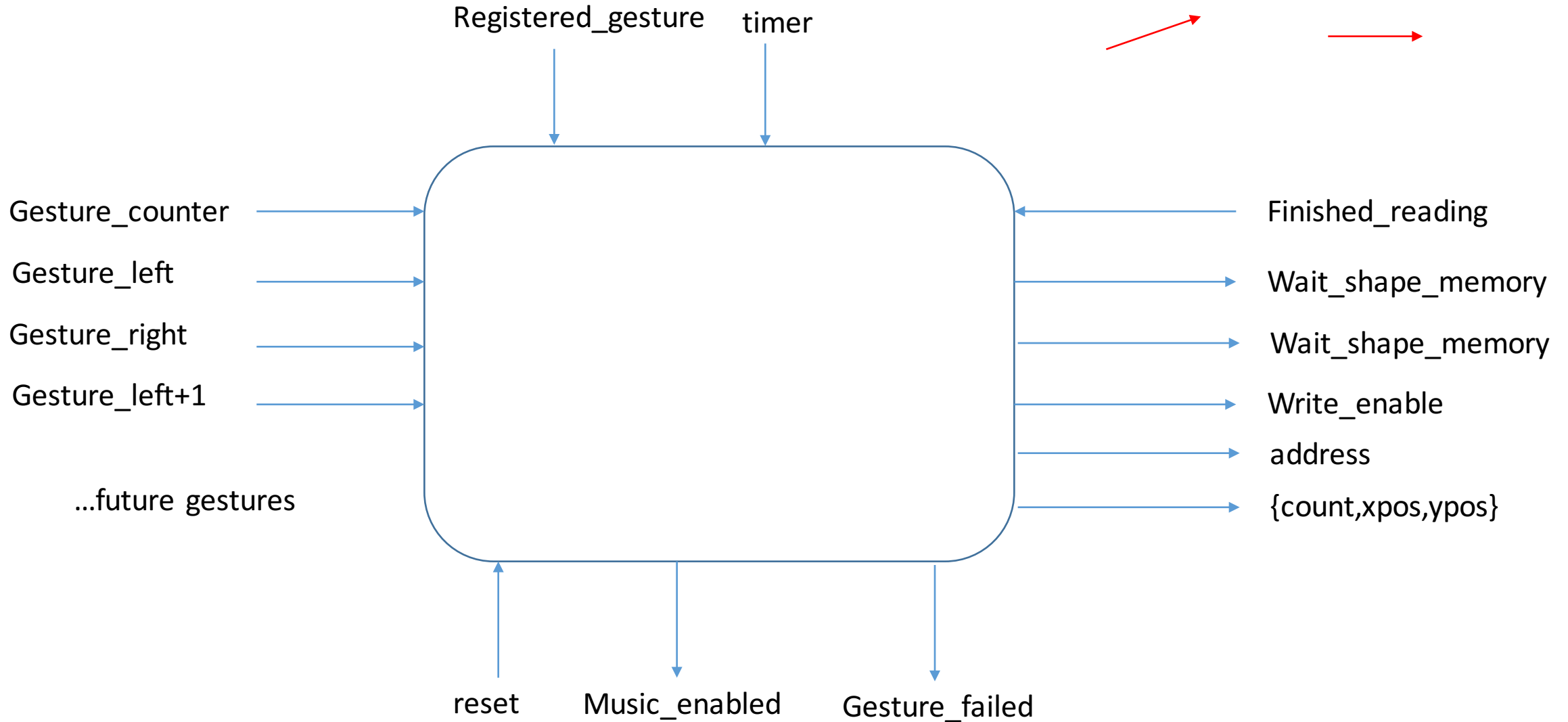
Block Diagram



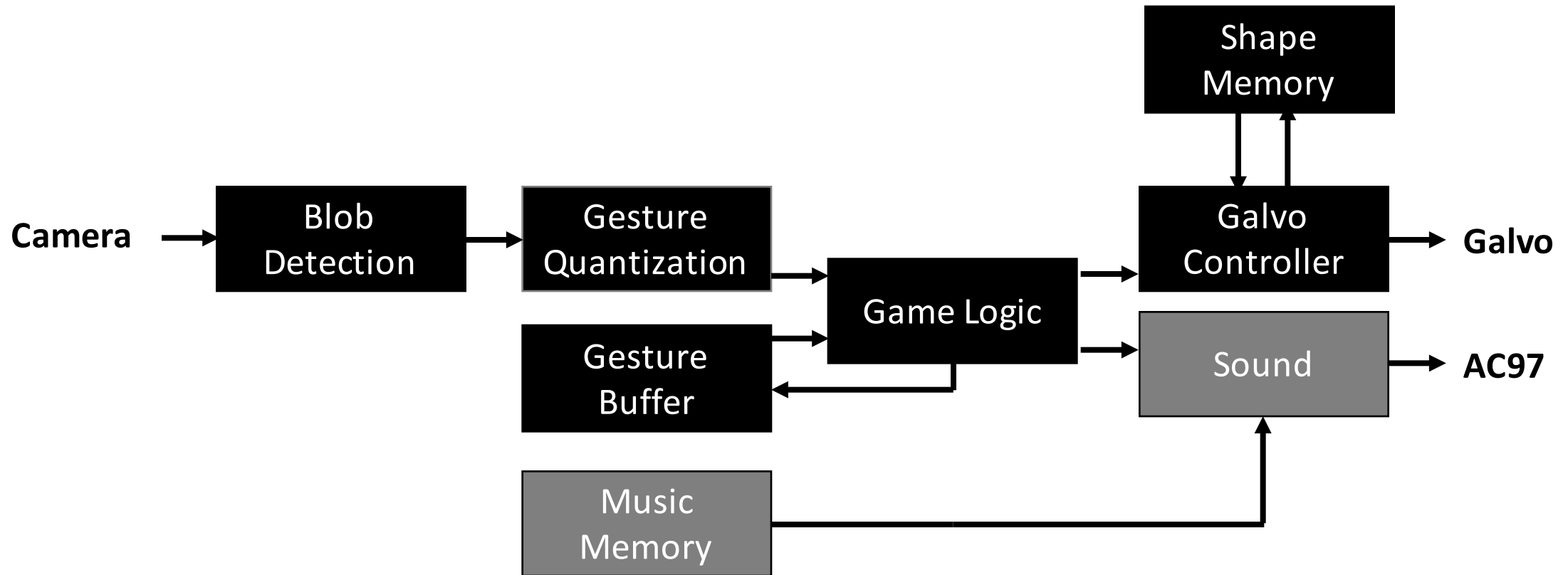
Game logic FSM

- Game logic decides whether or not valid gesture has been completed in interval, and makes buzzer sound if player has failed.
- Game logic plays music to go with gesture
- Game logic loads new moves in from gesture buffer to be displayed on screen and updates shape memory appropriately

Game logic FSM



Block Diagram



Sound and music memory

