Action Tracking System! A.T.S

Shubhang Chaudhary
Raghavendra Srinivasan

Project Overview

- A.T.S is an interactive system that has a basic version of a moving Stick figure on the screen.
- Stick figure would mimic a human body performing real time actions.
- The user's actions are tracked with a camera that detects the color markers attached to the arms, legs and torso of their body.

Inspiration <= Application;

- Augmented reality
- Security & Surveillance
- Medical imaging and video editing
- Human computer interaction

SYSTEM INPUTS & OUTPUTS

Inputs?

- Video from camera.
- Color markers indicating points on body.

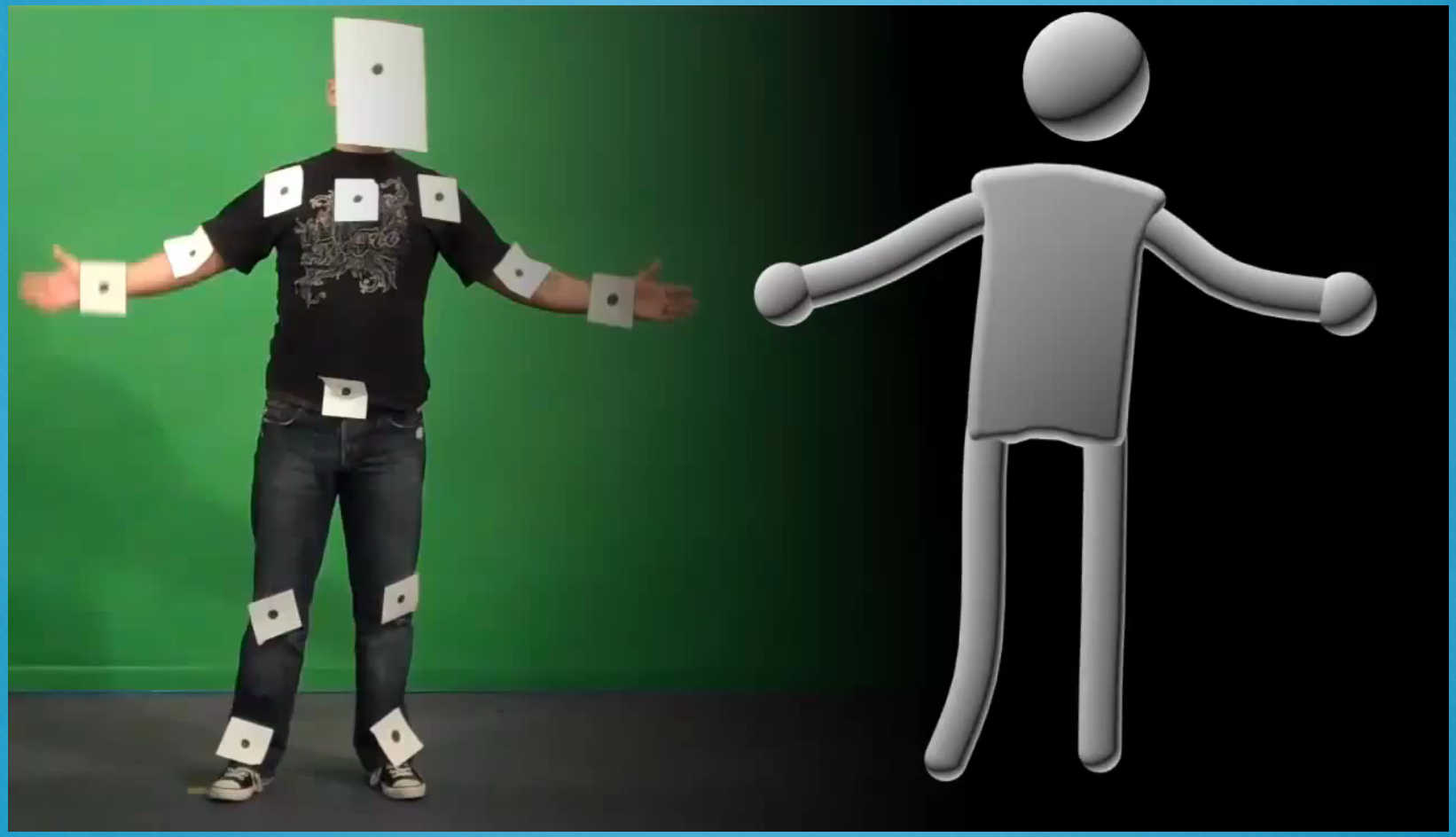
Outputs?

- Stick figure that would mimic human body performing actions in real time.

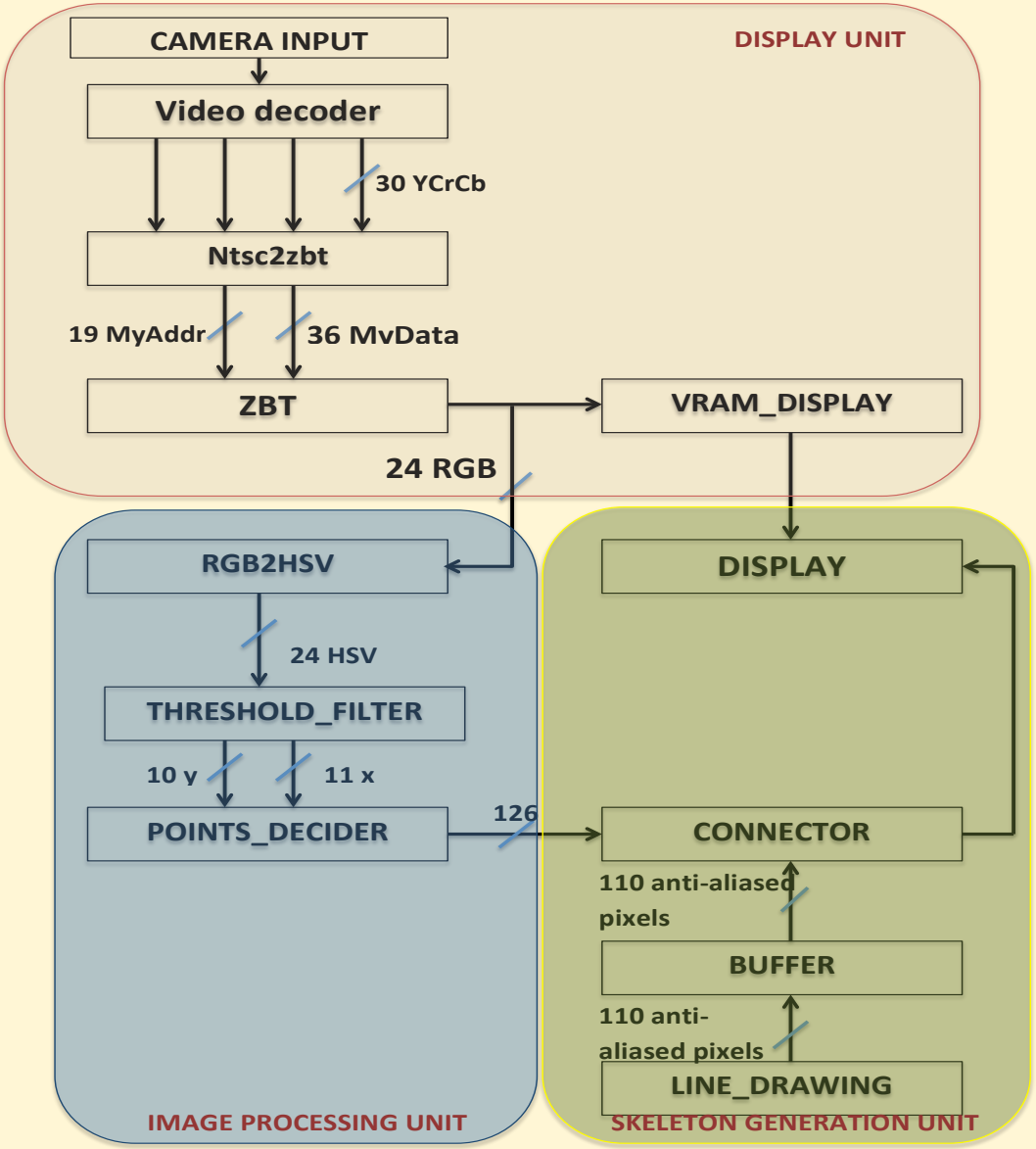
GOAL

REAL TIME INPUT

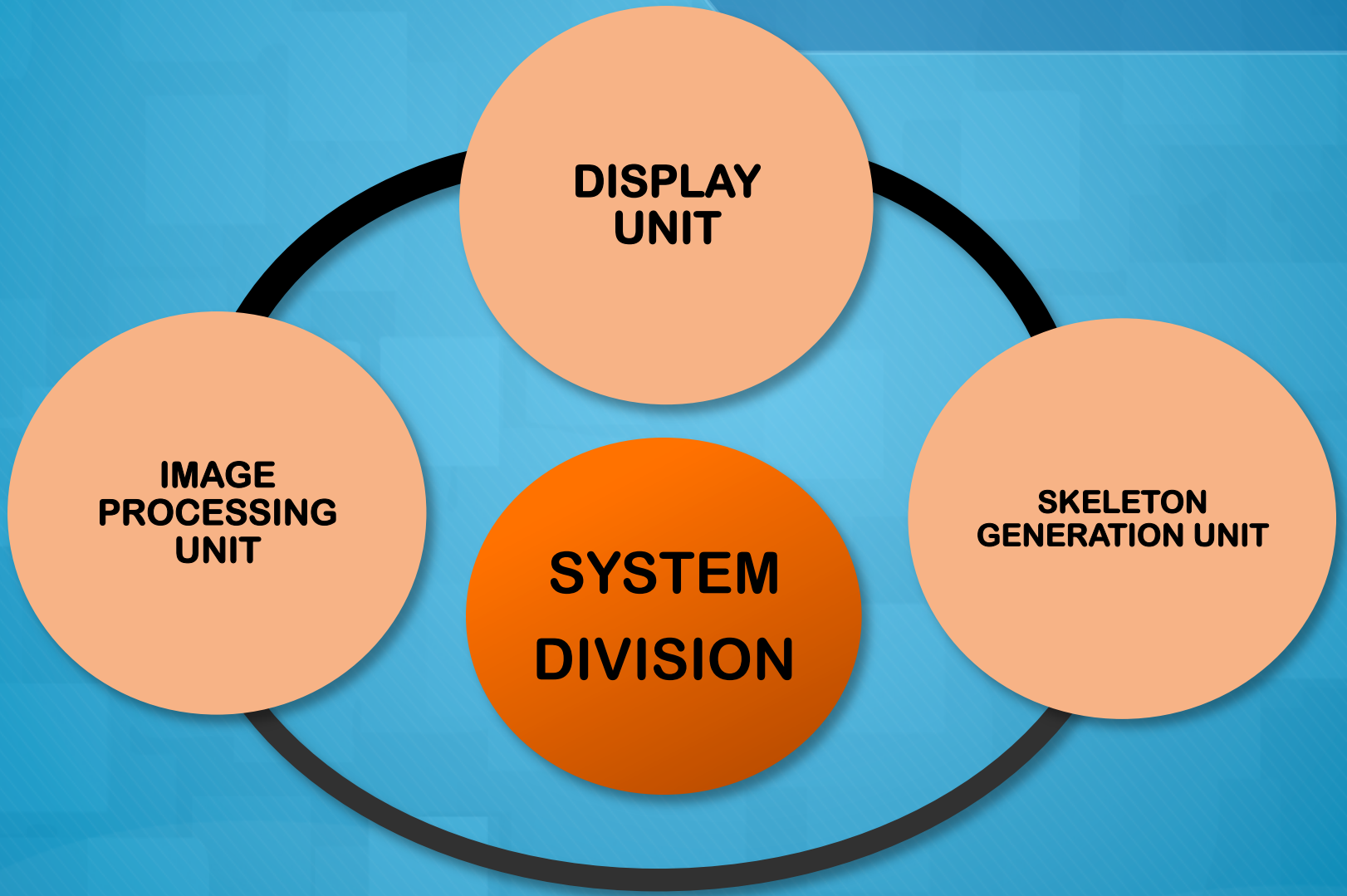
A.T.S OUTPUT

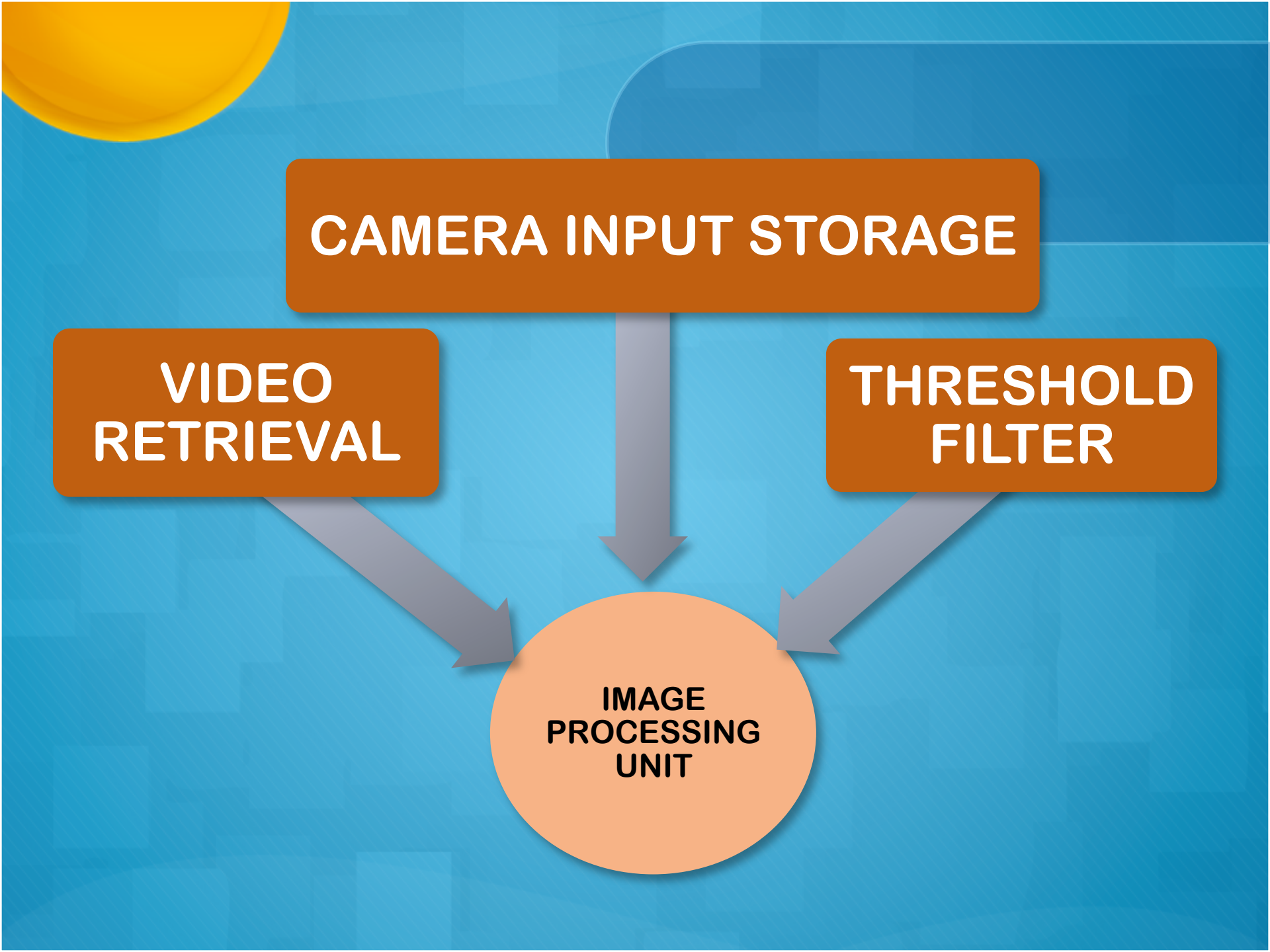


SYSTEM BLOCK DIAGRAM



HIGH LEVEL DESCRIPTION




CAMERA INPUT STORAGE
VIDEO RETRIEVAL
THRESHOLD FILTER
IMAGE PROCESSING UNIT

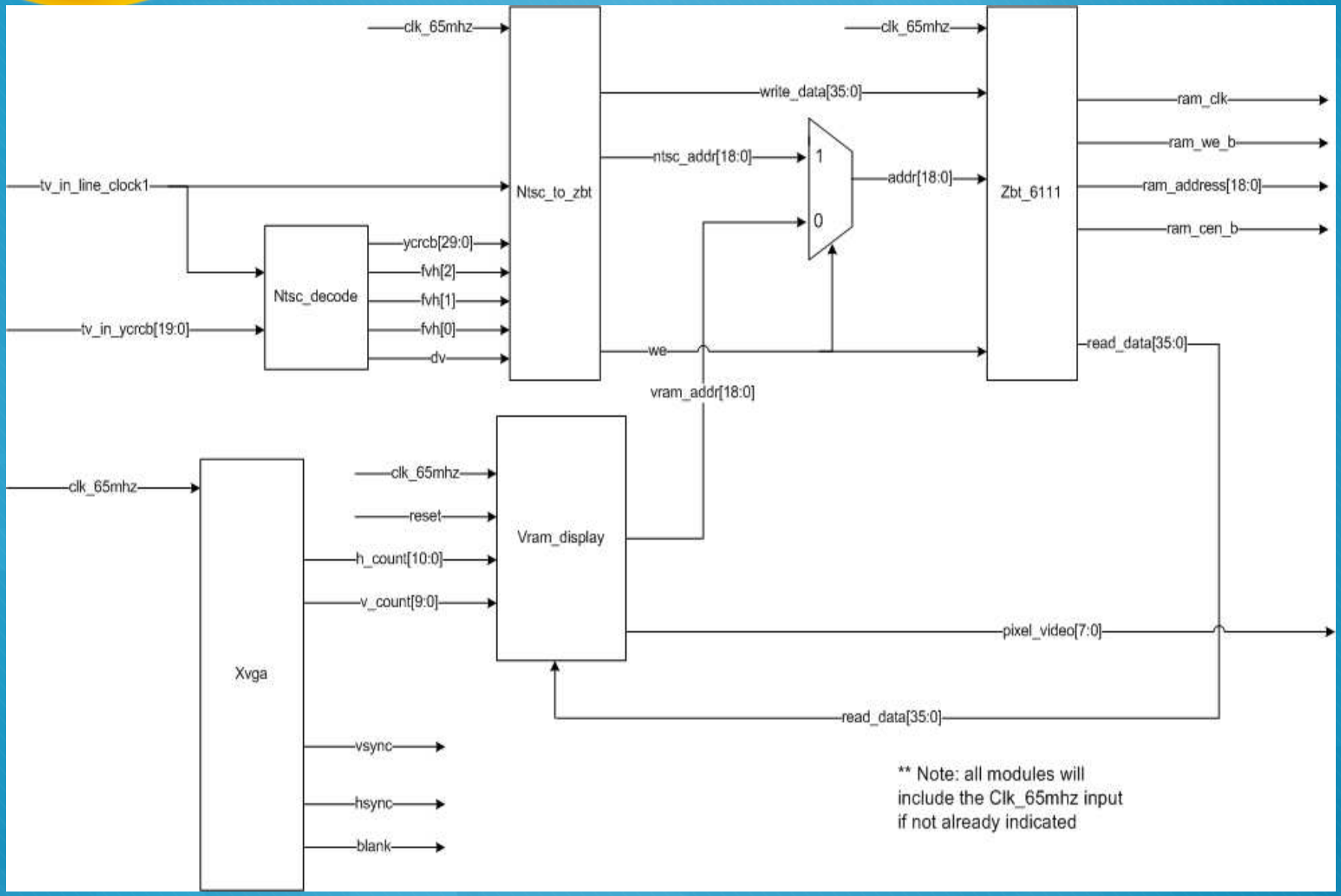
CAMERA INPUT STORAGE

1. Receives 20bit Y, Cr, Cb from the Camera.
2. Converts each pixel into a 30bit YCrCb value.
3. Converts YCrCb value into 24bit RGB value.
4. Stores 6bits of R, G, and B for each pixel.
5. Stores 2 pixel per memory location of ZBT.

VIDEO RETRIEVAL

1. Obtains 24 bit RGB pixels from the ZBT.
2. Displays the pixels on the screen.

CAMERA INPUT STORAGE

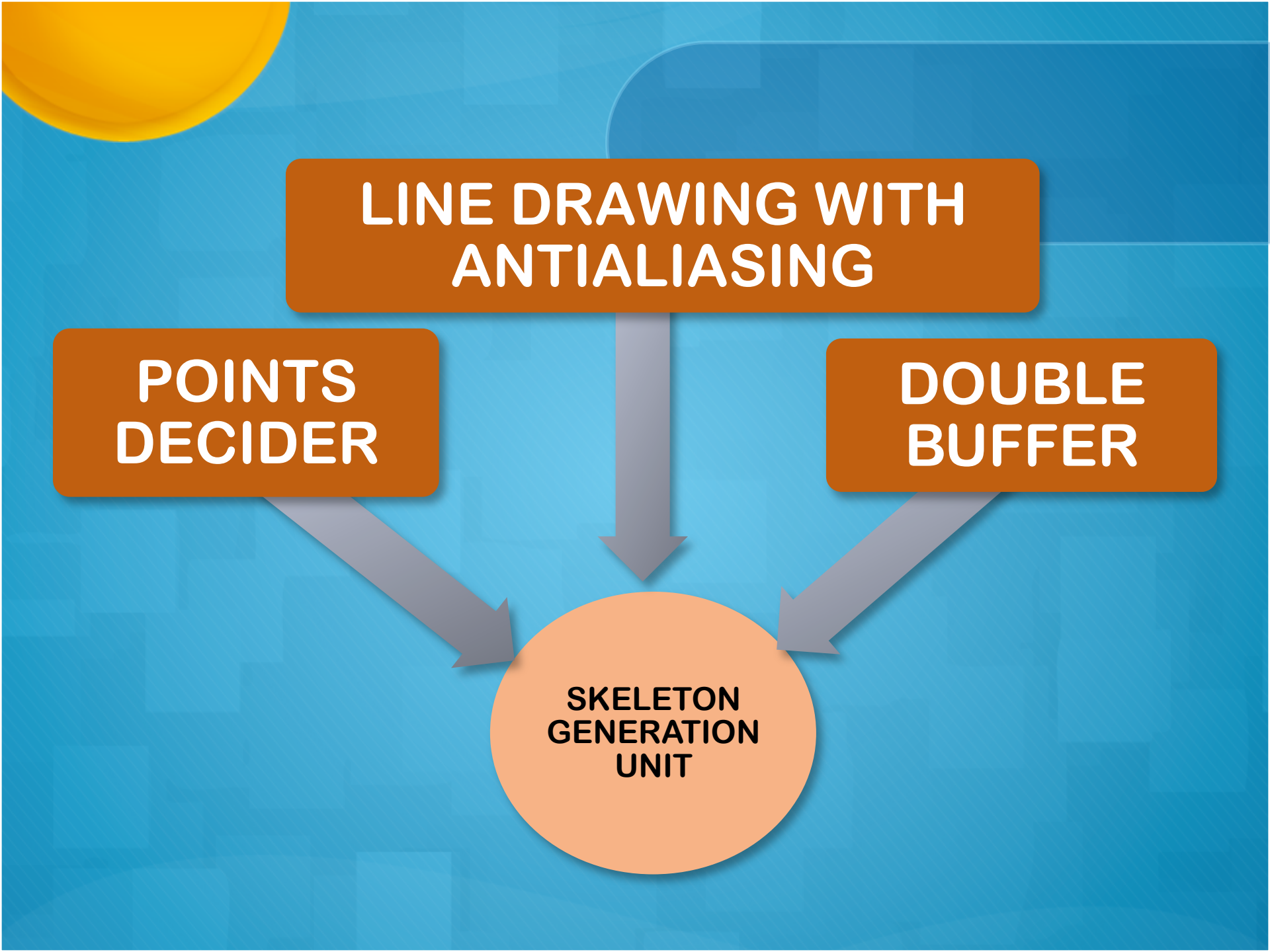


THRESHOLD FILTER

1. Implements Object Recognition and Tracking by HSV filtering.
2. Averages marker position for 16 frames.
3. Calculates the Center of masses of various markers.
4. Sends six 21 bit Center of masses as output for the six markers that are detected.

TECHNICAL APPROACH

- The brightest pixel can be determined by seeing which pixel has the highest color values;
- A "blob" of color can be determined by choosing a starting color, setting a range of variation, and checking the neighboring pixels of a selected pixel to see if they are in the range of variation.
- Areas of change can be determined by comparing one frame of video with a previous frame, and seeing which pixels have the most significantly different color values.


LINE DRAWING WITH ANTIALIASING
POINTS DECIDER
DOUBLE BUFFER
SKELETON GENERATION UNIT

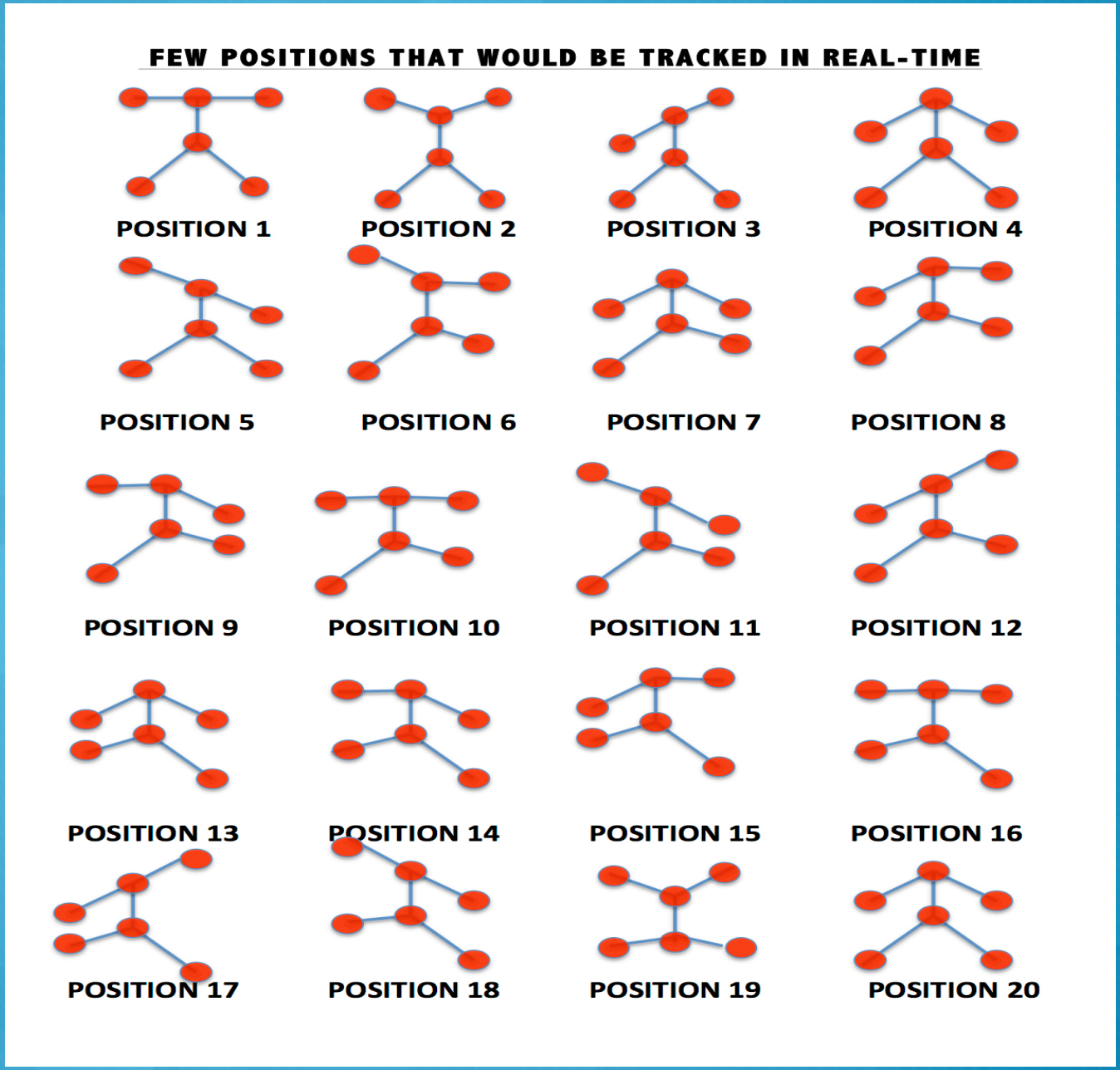
POINTS DECIDER

1. Takes six 21 bit center of masses as the input.
2. Outputs corresponding points denoting the arms, legs and the torso points. (2 points each for arms, legs and the torso.)

ALGORITHM

1. Implements Point Sorting.
2. Uses Artificial Intelligence – Uses the fact that the markers on the torso lie between the markers denoting the left and right parts of the body.
3. Assumes few Constraints –
    - Human cannot cross his/her hands or legs
    - Human cannot turn around as this a 2D motion tracking.
    - Human cannot put his/her hands below his/her legs.
    - Human cannot join their hands/legs.

FEW POSITIONS THAT WILL BE TRACKED



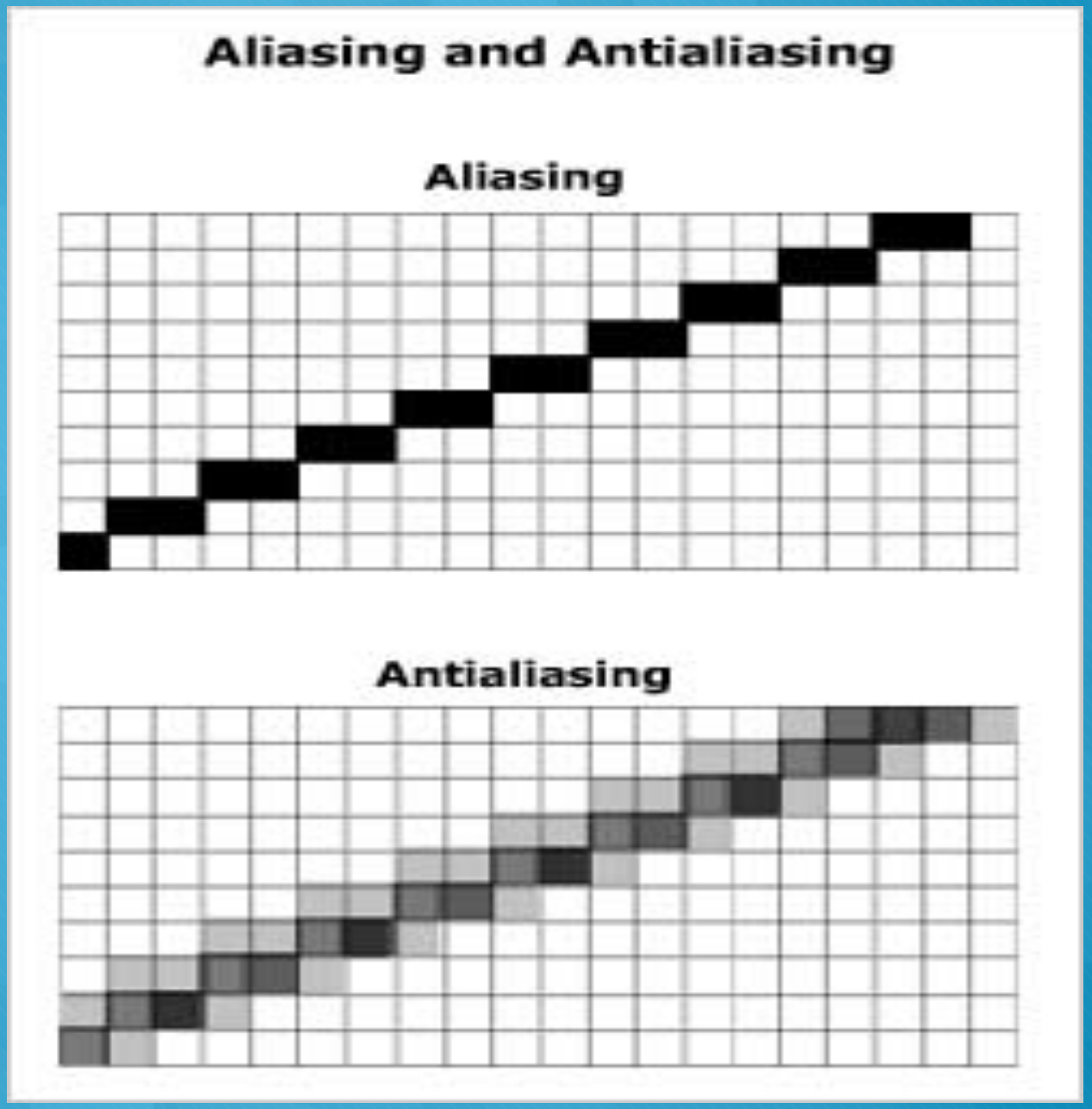
LINE DRAWER WITH ANTIALIASING

1. Takes two 21 bit center of masses as the input.
2. Outputs all points in between them. (one every 2 clock cycles)
3. Also outputs the Anti-aliased points in different color to make the line look natural.

JACK BRESHENHAM'S LINE ALGORITHM

1. Determines which points in an n-dimensional raster should be plotted in order to form a close approximation to a straight line between two given points.
2. It uses only integer addition, subtraction and bit shifting, all of which are very cheap operations in standard computer architectures.
3. Anti-aliasing is done by using the same algorithm multiple times and replacing required pixels with different colors.

SAMPLE OUTPUTS OF LINE DRAWER

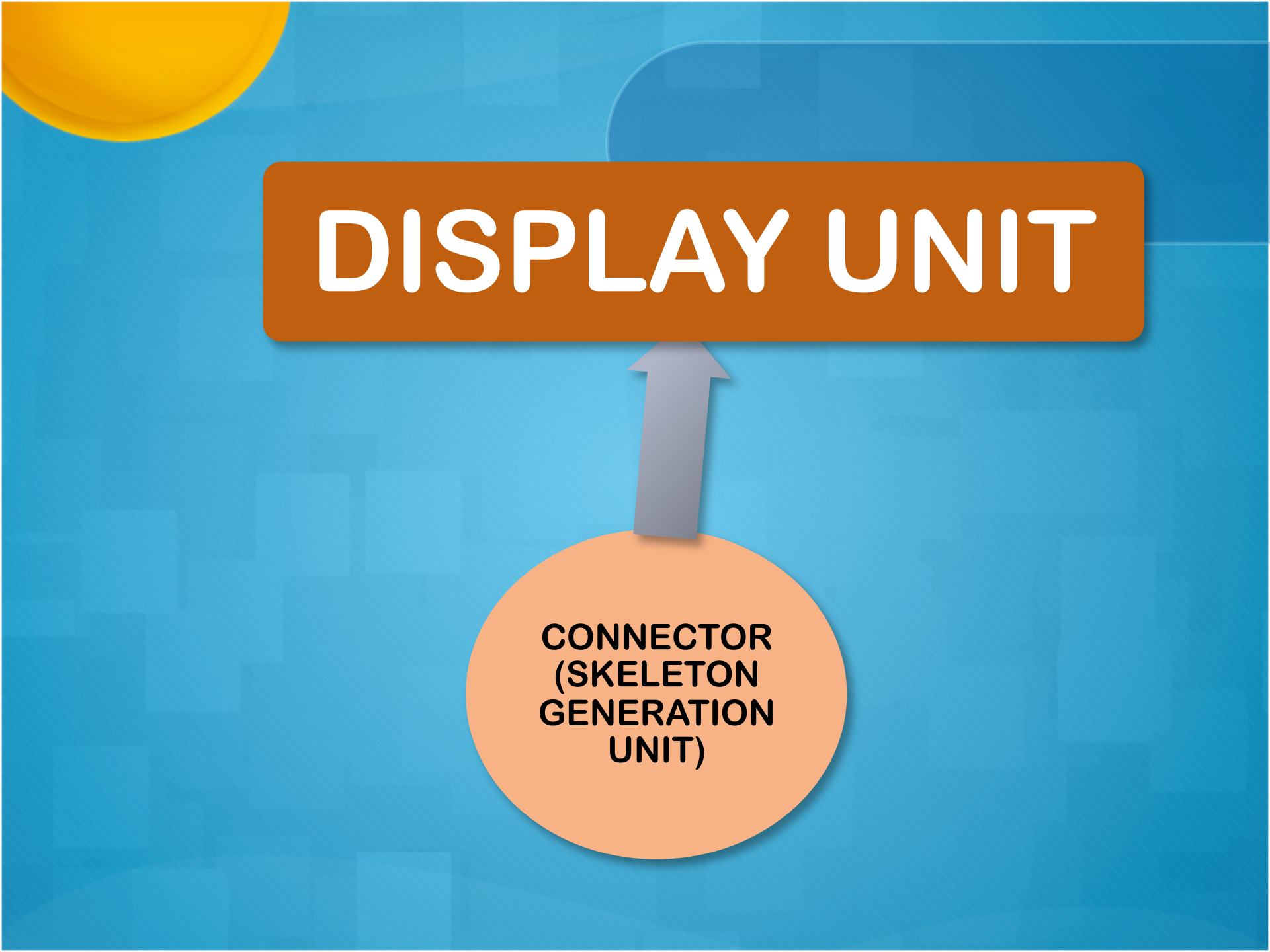


DOUBLE BUFFER

1. As the name reads, uses two buffers, switching between each other every frame.

2. Uses the ZBT memory to store the status of the pixels on the display for each point generated by the line drawer module.

3. It enumerates a different status for different colors.


DISPLAY UNIT
CONNECTOR (SKELETON GENERATION UNIT)

CONNECTOR

1. Uses the Line Drawer Module to connect the required points.

2. Uses the Points decider module to figure out what points to connect

3. Uses the Double buffer module to store everything.

4. Outputs the pixels to VGA_out.

CHALLENGES

1. The biggest challenge obviously will be to detect the color markers as all of them are of the same color.

2. To implement Anti-aliasing of a line.

3. To enumerate the right points to join using artificial intelligence.

MILESTONES

Till date, everything thing other than marker detection has been completed and tested.

(Thanks to Gim)

11/15 Figure out the best color for Detection

11/16 Figure out the range of H,S,V for the color to be detected.

11/17,18 Make code for marker detection.

11/21 Test Code and make changes if required.

11/23 Put all the modules together.

Plan for Implementation of Extensions.

REFERENCES

http://en.wikipedia.org/wiki/Bresenham%27s_line_algorithm

Video Tracking: Theory and Practice
By Andrea Cavallaro, Emilio Maggio

THANK YOU FOR YOUR ATTENTION ☺