

ACTION TRACKING SYSTEM

1. Overview

The Action Tracking System (ATS) is a project using FPGA to achieve real time motion tracking with position feedbacks from an NTSC camera. The system uses one camera to obtain a straight view of the field, where the target human will be. Using the 2-D information from the camera, an object recognition module will perform threshold filtering to track the positions of various joints of the human in the view of the camera.

The joints being tracked are used to reproduce a Stick figure of the human on the screen. The Action Tracking system, also involves the usage of artificial intelligence to enumerate the right connections between the joints being tracked. The ATS includes the following main modules.

Shubhang's Part

1. `video_decoder`: This module takes signals from a NTSC camera and decodes that information into YCrCb that is 30 bits long.
2. `ntsc_to_ZBT`: stores pixel information as RGB into the on-board ZBT memory.
3. `YCrCb_to_RGB`: converts 30 bit YCrCb to 24 bit RGB.
4. `threshold_filter`: filters the incoming pixels with color thresholds in order to find the position of the human joints.

Raghavendra's Part

5. `distance_calculator`: Takes 2 points in 2-D space as the inputs and gives the distance between them.
6. `line_drawing_tool`: Takes 2 points in 2-D space as the inputs and connects them with a line of desired colour.
7. `Dots_Connector`: with the position information of the human joints and artificial intelligence, the `dots_connector` module decides the dots

to be joined and instantiates the `line_drawing_tool` module appropriately.

2. Description of Modules

This section explains the functions of each module in more detail, including the descriptions of inputs and outputs, test strategy, and potential challenges.

1. `video_decoder`: This module interfaces with the NTSC camera, decodes the input signal and converts the pixel information into a 30 bit value, with 10 bits each for Y, Cr and Cb. Since NTSC video standard implements interlacing mechanism, this module also outputs `f` as a one bit value to indicate whether it is in odd or even field. One bit values, `v` and `h` are outputted to show the position of that pixel. It outputs a `data_valid` bit to tell other modules that the data is ready for use.
2. `ntsc_to_ZBT`: This module is responsible for generating corresponding address in ZBT memory to store the current filtered incoming pixel information. Since the camera uses a different clock frequency, this module is the entry point to synchronize data from the camera with the rest of the system. A sample module is already written, but slight changes have to be made in order for it to store colored pixels.
3. `YCrCb_to_RGB`: This module converts 30bit YCrCb pixel input into a 24bit RGB string. Such module is already written, but we note that this module has a 5 clock cycle delay between inputs and outputs. Therefore, the address into the ZBT memory should be delayed appropriately in order to store in the correct location.
4. `threshold_filter`: This module reads directly from the incoming video data; an YCrCb to RGB converter is embedded in this

module in order to store RGB information in memory. Then this module filters each pixel using a simple color threshold. On the Human body there will be markers for the torso, hands and legs, we find the center of mass of each marker so that we can find the configuration of the human body.

5. `distance_calculator`: This module does not have any external dependencies and works merely as a tool for the `Dots_connector` module. It takes 2 points in the 2-D space as the inputs and gives the distance between them. It also gives a `data_valid` pulse which denotes the occurrence of valid output data.
6. `line_drawing_tool`: This module also does not have any external dependencies and again works as a tool for the `Dots_connector` module. It takes 2 points in the 2-D space as inputs and connects them using the Bresenham's line algorithm.
7. `Dots_Connector`: This module is the heart of the system. It takes the position of the joints and figures out the appropriate joints to be connected by using artificial intelligence. It also instantiates the `distance_calculator` and `line_drawing_tool` modules appropriately.

3. ATS Block Diagram

