A Neural Network Based Head Tracking System

CSL2010 Project

Deepanshu
B20ME025

Problem Statement

With the expansion of multimedia computers and peripheral equipment devices, video conferencing finally appears to ready to enter the mainstream. But there arises a problem that the video conferencing system uses a fixed camera which can make the meeting non-interactive and boring as we are not using the whole space and our body movements and the user is also tied to a fixed location.

Solution

- In this project, a neural network head tracking system is presented.
- It learns the characteristics of a person’s head in real time and automatically tracks it around the room.
- The camera movements in this video conferencing system closely resemble the movements of human eye.

Hardware Implementation

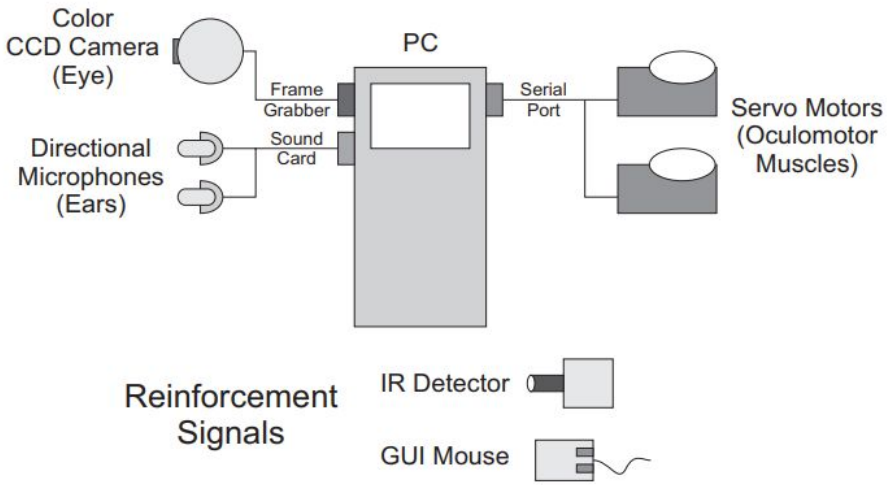


Figure shows a schematic of the tracking system named “Marvin”.

- Marvin's eye consists of a small CCD camera with a 65 degree field of view.
- Two RC servo motors are used to rapidly pan and tilt over a wide range of viewing angles.
- The system also includes two microphones or ears that give Marvin the ability to locate auditory cues.
- Integrating auditory information with visual inputs allows the system to find salient objects better than with either sound or video alone.

Neural Network Architecture

- The video stream is first digitized into a series of raw 120X160 RGB images. Each RGB color image is then converted into its YUV representation, and a difference (D) image is also computed as the absolute value of the difference from the preceding frame.
- The Y component represents the luminance or grayscale information.
- The U and V channels contain the chromatic or color information.
- Motion information in the video stream is captured by the D image where moving objects appear highlighted.

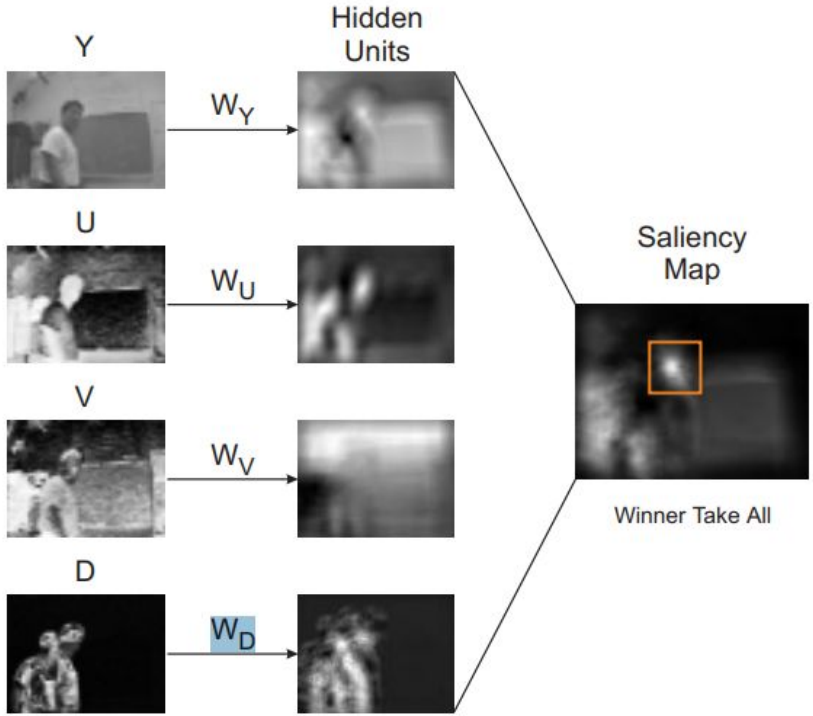
Figure 2: Preprocessing of the video stream. Luminance, chromatic and motion information are separately represented in the Y, U, V, D channels at multiple resolutions.

Neural Network Architecture

Marvin locates the salient objects at the different resolutions. The YUVD input images are filtered with separate 16x16 kernels, denoted by W_Y, W_U,W_V, and W_D respectively. Results are Y^s,U^s,V^s,D^s:

$$\bar{A}^s(i,j) = W_A \circ A^s = \sum_{i',j'} W_A(i',j')\, A^s(i+i', j+j') \qquad (1)$$

Where s denotes the scale resolution of inputs. A is any Y,U,V,or D channels. These filtered images represent a single layer of hidden units in the neural network. The saliency map X_s :



$$X^s(i,j) = c_Y\, g[\bar{Y}^s(i,j)] + c_U\, g[\bar{U}^s(i,j)] + c_V\, g[\bar{V}^s(i,j)] + c_D\, g[\bar{D}^s(i,j)] + c_0. \qquad (2)$$

Neural Network Architecture

g(x) = tanh(x)

The scalar variables c_Y,c_u,c_V and c_D represent the relative importance of different luminance, chromatic, and motion channels in the saliency map.

The final output of the neural network is then determined in a competitive manner by finding the location (i_m,j_m) and scale s_m of the best possible match:

g[X^s(i,j)] is the relative probability that the head exist at (i,j) at resolution s.

$$g[X_m] = g[X^{s_m}(i_m, j_m)] = \max_{i,j,s} g[X^s(i,j)]. \qquad (3)$$

Training and Results

The neural network is updated when the maximally salient location on neural network does not correspond to the desired object's true location. A cost function proportional to the sum squared error terms at the maximal location and new desired location is used for training.

$$e_m^2 = |g_m - g[X^{s_m}(i_m, j_m)|^2, \quad (4)$$

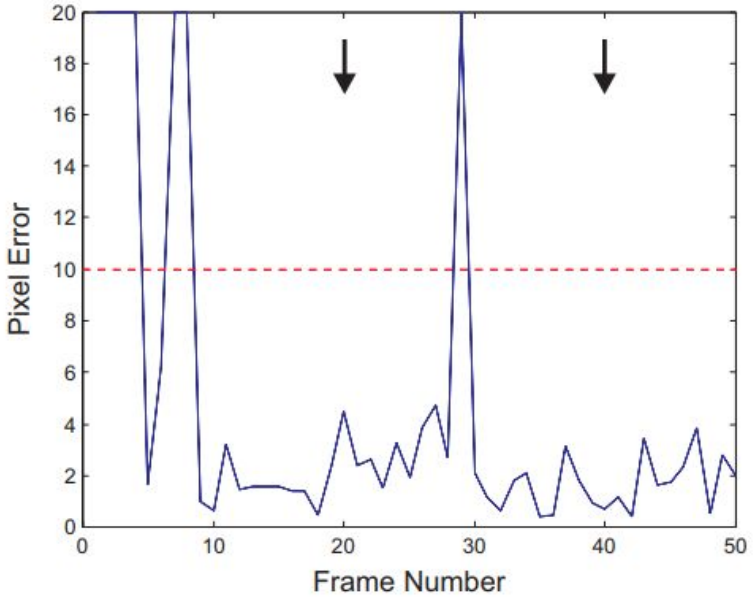
$$e_n^2 = \min_s |g_n - g[X^s(i_n, j_n)|^2. \quad (5)$$

In the following example, the constants g_m=0 and g_n=1 are used. The Gradients to Eqs. 4-5 are then backpropagated through the convolutional network.

$$\Delta c_A = \eta\, e_m g'(X_m) g[\bar{A}(i_m, j_m)] + \eta\, e_n g'(X_n) g[\bar{A}(i_n, j_n)], \quad (6)$$

$$\Delta W_A = \eta\, e_m g'(X_m) g'(\bar{A}_m) c_A A_m + \eta\, e_n g'(X_n) g'(\bar{A}_n) c_A A_n. \quad (7)$$

Results

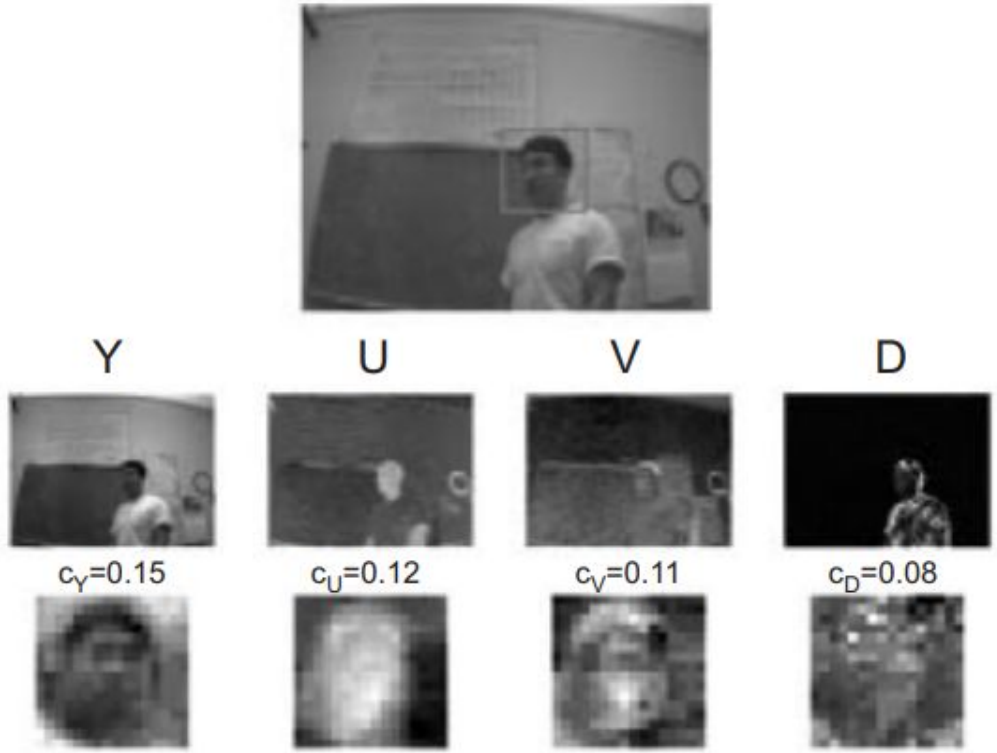


An example of how quickly Marvin is able to learn to track one of the authors as he moved around his office. The weights were first initialized to small random values, and Marvin was corrected in an online fashion using mouse inputs to look at the author's head.

Conclusion

Figure shows the inputs and weights of the network as the author moved around his office.

The relative weighting of different input channels shows that the luminance and color information are the most reliable for tracking the head.



Thank You