Possible Extensions to CCR Vision

by: Josh Domina

Use a Second Webcam

A second webcam could be used to allow for greater visibility of the track. The virtual track structure would need to be modified to indicate which camera defined track points came from. Also a mechanism for identifying camera overlap would be needed. Alternatively the camera data could be combined into one large image that the program would use. The second webcam would need to be configured in the same manner as the first webcam.

Use a Better Webcam

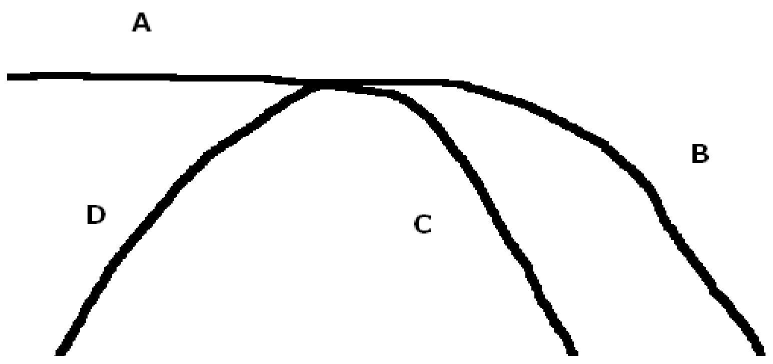
Continue to use a single webcam, but get one with a greater field of view and a faster frame rate. Ideally this camera could see the entire track. It must be able to get the entire track in view while keeping objects on the track large enough to be detected. Large aperture values are preferred.

Modify ParseTrack() to handle switches

As it stands the ParseTrack() function does not handle switches. Modify it so that it will follow switches to find motion. A boolean flag will need to be added to the Train user defined type to indicate whether or not

Further abstraction of the Track (from Dr. Pankratz)

Further abstract the track by breaking it in to sections separated by switches. This additional layer of abstraction can be represented by its own data structure that links to track points in the existing structure. When detecting motion all sections connecting to the current section will be examined. If there is a switch it will be examined at the same time as the non-switch section. If motion overlaps two sections it will be analyzed as a special case. This algorithm will operate recursively.



Implement Signal/Messaging System

Modify CCR Vision so that it responds to messages or signals. When asked for the locations of trains CCR Vision should send the number of trains found and the beginning and ending track points and whether or not they used a switch. The recipient program could use the virtual track to plot the trains. Response time would need to be quick, preferably less than 0.25 seconds. Recipient programs can generate the visual output by using the track file and the data for located trains.

Remove the GUI and have CCR vision run as a Daemon

As an extension of the messaging idea, make it so that the CCR Vision software runs as a daemon with no GUI. This may require porting the algorithms to a language other than Visual Basic. This daemon should function much like the messaging system above.

Make the data web accessible

Write a web program that can read in the track file. This program will use the message/signal aspect of CCR Vision to request train locations. It will then display this information on the web, either graphically by generating the virtual track image, analytically by displaying the information received in the message, or a combination thereof.