

Validation

Jason Banich

March 13, 2012

The success of this software can be measured by its ability to recognize crosswalks and guide the users across them with better accuracy than the user on their own. There are a couple requirements that any algorithm must follow in order to be considered a success:

1. Successfully recognize crosswalks
2. Successfully guide user across the crosswalk (guiding them out of the crosswalk is an automatic failure)

In order to measure these metrics, a multitude of experiments will be run with blind and non-blind users in order to determine the success.

Some other metrics that may end up being useful are as follows:

1. CPU usage/Frames per second
2. Jitter
3. Robustness
4. Frame by frame analysis of where it is leading the user

The CPU usage/FPS measurement would be handy in order to determine if the method is too complex for the application. For example, if we're getting one frame every ten seconds, we can determine that that algorithm is too complex and we need to either simplify it, optimize it, or scrap it.

Jitter would be measured based on the amount of 'uneasiness' in the guidance. If the application is guiding the user left and right every other frame, and not allowing them to walk straight at all, they'll never reach the other end of the street.

Robustness refers to the amount of different scenarios that the application can perform in, such as low light, bright light, nighttime, etc. The more situations that the app can perform under, the better.

If we analyze frame by frame, we can see where the program is leading the user. If the program is leading the user outside of the crosswalk frequently, we may want to claim that it is defective.