TRAFFIC LIGHTS DETECTION IN ADVERSE CONDITIONS USING COLOR, SYMMETRY AND SPATIOTEMPORAL INFORMATION

George Siogkas
Evangelos Skodras
Evangelos Dermatas

Electrical & Computer Engineering Dept.
University of Patras, Greece
GOAL

Develop a system for automated vision based detection of traffic lights, robust even under adverse conditions.
Why is this system important?

To warn drivers for red-light running dangers.

For autonomous vehicles driving in existing road infrastructure.
Why hasn’t it been solved yet?

Traffic sign recognition has taken all the glory.

Google Scholar Search (February 21\textsuperscript{st}, 2012):
• “Traffic sign recognition” approx. 1660 results.
• “Traffic light recognition” approx. 75 results
Why hasn’t it been solved yet?

Big cities pose an extremely hard problem because of multiple light sources.

Night driving is also a challenging scenario.

Bad weather makes detection even more difficult.

Combining all of the above, even in small scale, makes detection almost impossible.
What has been tried?

• Color thresholding in RGB or HSV
• Symmetry Detection with Hough or other novel methods.
• Structural Models of Traffic Lights.
• Tracking modules to help minimize false positive results.
• A mixture of the above.
Proposed System Overview

Frame Acquisition

RGB to CIE-\(L^*a^*b^*\) Conversion

Red-Green Color Difference Enhancement

Hole Filling Process

Image Pre-Processing

TL Candidate Detection

Local Maxima/Minima Localization

Radial Symmetry Detection

TL Candidate Verification

Spatiotemporal Persistency Check

TL Present?

Reject Candidate

NO

YES

Inform Driver

TL Present?
Image Pre-Processing

Step 1: RGB to L* a* b*

L*:

a*:

b*: 
Image Pre-Processing

Step 2: Red – Green Enhancement

L*:

Multiplying L* and a* to enhance only bright red and green regions (exclude trees, roofs, etc.)
Image Pre-Processing

Step 3: Yellow – Blue Enhancement

L*:

b*:

Multiplying L* and b* to enhance only bright yellow and blue regions (red lights include yellow and green lights include some blue)
Image Pre-Processing

Step 4: Blooming Effect Reduction

R-G:

Y-B:

Image filling of the two images followed by addition. This reduces the “blooming effect” significantly.
Step 4: Blooming Effect Reduction

R-G:

Y-B:

Image filling of the two images followed by addition. This reduces the “blooming effect” significantly.
Image Pre-Processing

Step 1: RGB to L*a*b*

L*: 

a*: 

b*:
Image Pre-Processing

Step 2: Red – Green Enhancement

Multiplying $L^*$ and $a^*$ to enhance only bright red and green regions (exclude trees, roofs, etc.)
Image Pre-Processing

Step 3: Yellow – Blue Enhancement

$L^*$:

$b^*$:

Multiplying $L^*$ and $b^*$ to enhance only bright yellow and blue regions (red lights include yellow and green lights include some blue)
Step 4: Blooming Effect Reduction

R-G:

Y-B:

Image filling of the two images followed by addition. This reduces the “blooming effect” significantly.
Image Pre-Processing

Step 4: Blooming Effect Reduction

R-G:

Y-B:

Image filling of the two images followed by addition. This reduces the “blooming effect” significantly.
Traffic Light Candidate Detection
Traffic Light Candidate Detection

Step 1: Fast Radial Transform
Traffic Light Candidate Detection

Step 1: Fast Radial Transform
Traffic Light Candidate Detection

Step 2: Maxima/Minima Localization
Traffic Light Candidate Detection
Traffic Light Candidate Detection

Step 1: Fast Radial Transform
Traffic Light Candidate Detection

Step 1: Fast Radial Transform
Traffic Light Candidate Detection

Step 2: Maxima/Minima Localization
Traffic Light Candidate Verification

FRAME #1:

Spatiotemporal persistency check: Traffic Lights candidates for 4 consecutive frames get verified.
Traffic Light Candidate Verification

FRAME #2:

Spatiotemporal persistency check: Traffic Lights candidates for 4 consecutive frames get verified.
Traffic Light Candidate Verification

FRAME #3:

Spatiotemporal persistency check: Traffic Lights candidates for 4 consecutive frames get verified.
Traffic Light Candidate Verification

FRAME #4:

Spatiotemporal persistency check: Traffic Lights candidates for 4 consecutive frames get verified.
Traffic Light Candidate Verification

FRAME #5:

Spatiotemporal persistency check: Traffic Lights candidates for 4 consecutive frames get verified.
Traffic Light Candidate Verification

FRAME #6:

Spatiotemporal persistency check: Traffic Lights candidates for 4 consecutive frames get verified.
Experimental Evaluation

**Quantitative**
- Normal weather conditions.
- City Driving.
- Daytime.
- Manually Annotated.
- 8-bit, 640x480 @ 25fps
- 11179 frames

**Qualitative**
- Rainy Weather
- Night Driving
- Different cameras.
- Different resolutions.
- Downloaded off YouTube, OR shot in Greek roads.

The quantitative data is coming from the Robotics Centre of Mines ParisTech and is publicly available at: [http://www.lara.prd.fr/benchmarks/trafficlightsrecognition](http://www.lara.prd.fr/benchmarks/trafficlightsrecognition)
Results in Normal Conditions
Results in Normal Conditions
Results in Rainy Conditions
Results in Night Driving Scenes
Method Limitations

False Positives

Night in Big Cities
Conclusions

• Proposed method shows good results in both normal and adverse conditions.
• Many false alarms, that could be reduced by some kind of structural model.
• Extremely difficult cases cannot be tackled.
• Future work on construction of an annotated database in adverse conditions.
• Also, the incorporation of a tracking module and a color consistency module.
Bibliography
Thank you for your attention.

Questions?