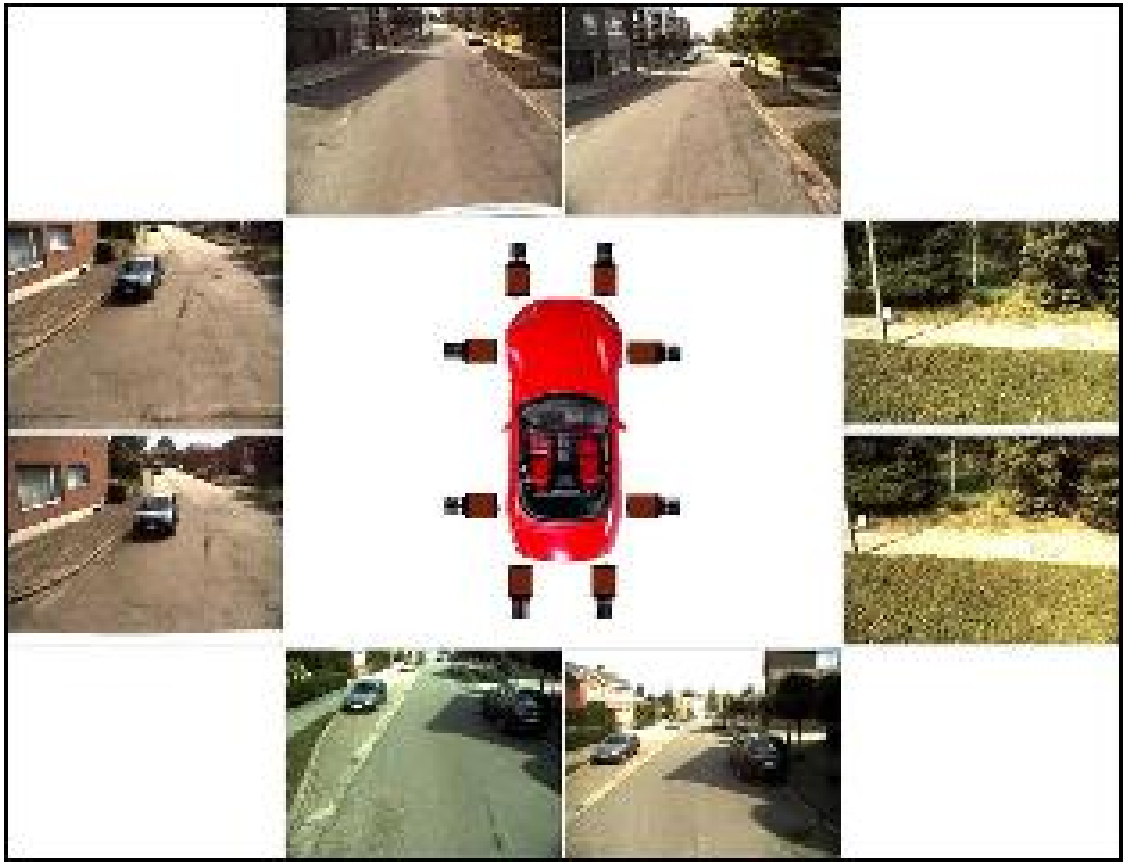


Introduction
Single-view
Segmentation
Detection and Recognition
Multi-view
2D optimization
3D optimization

Multiview Traffic Signs Detection / Recognition

February 10, 2009

Problem definition



- **Input:** Large set of views and corresponding camera locations.
- **Output:** List of detected traffic signs.

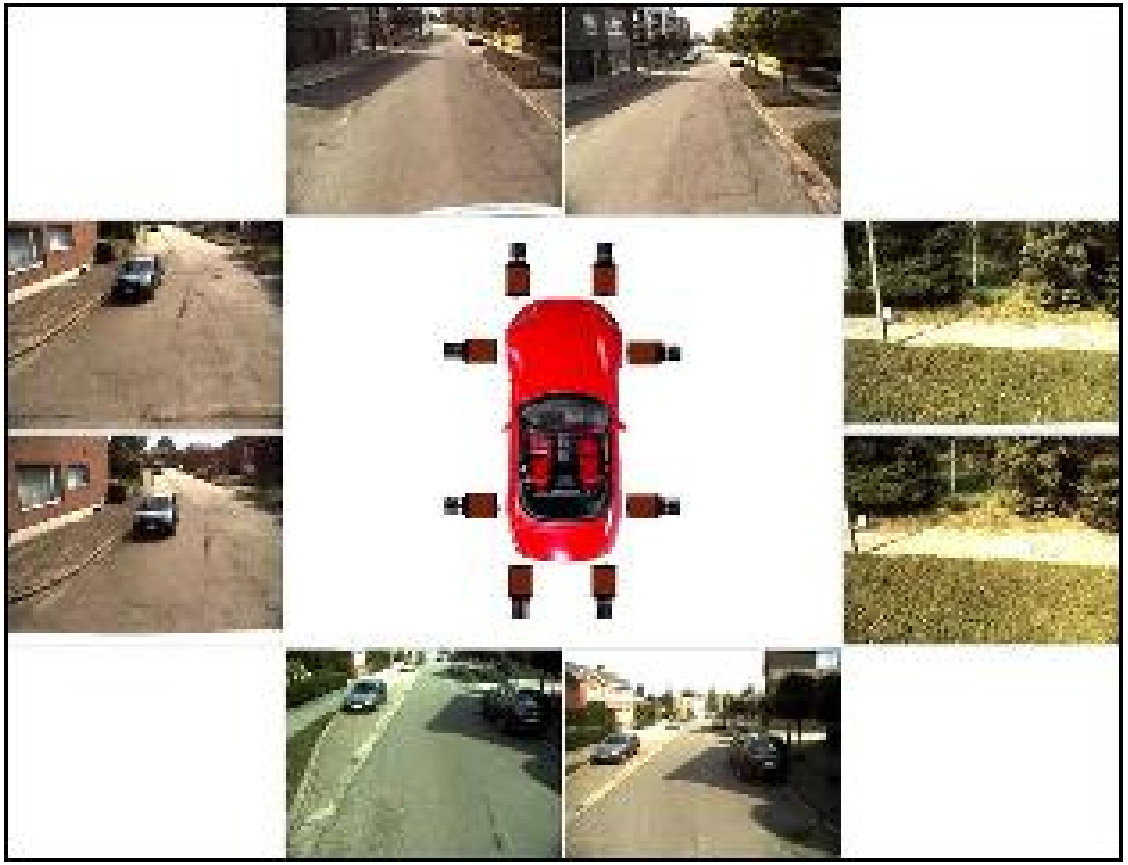
Problem definition

Multiview Traffic Signs Detection / Recognition

Introduction

Single-view
Segmentation
Detection and Recognition

Multi-view
2D optimization
3D optimization



- **Input:** Large set of views and corresponding camera locations.
- **Output:** List of detected traffic signs.



Outline

Multiview Traffic Signs Detection / Recognition

Introduction

Single-view
Segmentation
Detection and Recognition

Multi-view
2D optimization
3D optimization

Single-view

- **Segmentation** - extremely fast bounding box selection process with $FN \longrightarrow 0$.
  - Traffic signs are designed to be well distinguishable from background $\Rightarrow$ have distinctive colors and shapes.
- **Detection** - Adaboost classifier of bounding boxes.
- **Recognition** - SVM.

Multi-view

- **Global optimization** - combination of the single view detections satisfying geometrical constraints.



Outline

Multiview Traffic Signs Detection / Recognition

Introduction

Single-view
Segmentation
Detection and Recognition

Multi-view
2D optimization
3D optimization

Single-view

- **Segmentation** - extremely fast bounding box selection process with $FN \longrightarrow 0$.
  - Traffic signs are designed to be well distinguishable from background $\Rightarrow$ have distinctive colors and shapes.
- **Detection** - Adaboost classifier of bounding boxes.
- **Recognition** - SVM.

Multi-view

- **Global optimization** - combination of the single view detections satisfying geometrical constraints.



Outline

Introduction
Single-view
Segmentation
Detection and Recognition
Multi-view
2D optimization
3D optimization

Single-view

- **Segmentation** - extremely fast bounding box selection process with $FN \longrightarrow 0$.
  - Traffic signs are designed to be well distinguishable from background $\Rightarrow$ have distinctive colors and shapes.
- **Detection** - Adaboost classifier of bounding boxes.
- **Recognition** - SVM.

Multi-view

- **Global optimization** - combination of the single view detections satisfying geometrical constraints.



Outline

Introduction
Single-view
Segmentation
Detection and Recognition
Multi-view
2D optimization
3D optimization

Single-view

- **Segmentation** - extremely fast bounding box selection process with $FN \longrightarrow 0$.
  - Traffic signs are designed to be well distinguishable from background $\Rightarrow$ have distinctive colors and shapes.
- **Detection** - Adaboost classifier of bounding boxes.
- **Recognition** - SVM.

Multi-view

- **Global optimization** - combination of the single view detections satisfying geometrical constraints.

Color-based segmentation (thresholding)

- Estimation of connected components of a thresholded image ($T = [0.5, 0.2, -0.4, 1.0]^\top$)

Original image



Thresholded image $I(T)$



Connected components



Segmented bound. boxes



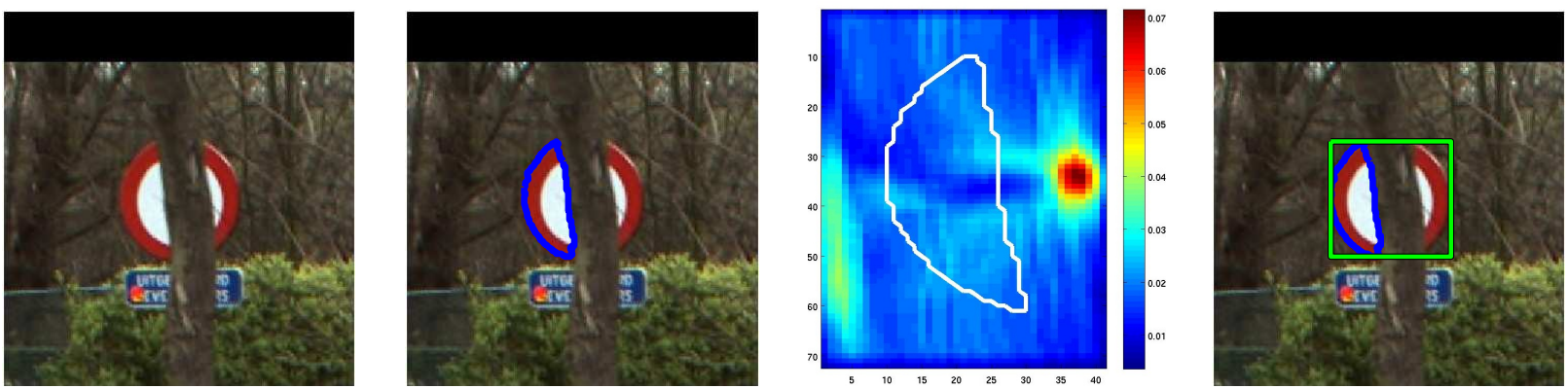
Shape-based segmentation

- Searching for specific shapes (rectangles, circles, triangles).

\+ Not all the traffic signs are locally threshold separable.

\- More time consuming, many responses for small shapes (every small region is approximatelly some basic shape).

Original image | **Segmented region** | **Hough accumulator** | **Refined bound. box**



Learning segmentation

- There are thousands of possible setting of such methods e.g. different projection from color space.
- Learning is searching for a reasonable subset of these methods/settings.
- **Optimal trade-off among FN, FP and the number of methods.**

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} FP(\mathcal{T}) + K_1 \cdot FN(\mathcal{T}) + K_2 \cdot \mathrm{card}(\mathcal{T}))$$

- Boolean Linear Programming selects ≈ 50 methods out of 10000 in 2 hours.
- **Segmentation results** for example:

$$FN_{BB} = 1.4\%, FP = 4307/\ 2\mathrm{Mpxl\ image}, (FN_{TS} = 0.5\%)$$

Learning segmentation

- There are thousands of possible setting of such methods e.g. different projection from color space.
- Learning is searching for a reasonable subset of these methods/settings.
- **Optimal trade-off among FN, FP and the number of methods.**

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} FP(\mathcal{T}) + K_1 \cdot FN(\mathcal{T}) + K_2 \cdot \mathrm{card}(\mathcal{T}))$$

- Boolean Linear Programming selects ≈ 50 methods out of 10000 in 2 hours.
- **Segmentation results** for example:

$$FN_{BB} = 1.4\%, FP = 4307/\ 2\mathrm{Mpxl\ image}, (FN_{TS} = 0.5\%)$$

Learning segmentation

- There are thousands of possible setting of such methods e.g. different projection from color space.
- Learning is searching for a reasonable subset of these methods/settings.
- **Optimal trade-off among FN, FP and the number of methods.**

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} FP(\mathcal{T}) + K_1 \cdot FN(\mathcal{T}) + K_2 \cdot \mathrm{card}(\mathcal{T}))$$

- Boolean Linear Programming selects ≈ 50 methods out of 10000 in 2 hours.
- **Segmentation results** for example:

$$FN_{BB} = 1.4\%, FP = 4307/\ 2\mathrm{Mpxl\ image}, (FN_{TS} = 0.5\%)$$

Learning segmentation

- There are thousands of possible setting of such methods e.g. different projection from color space.
- Learning is searching for a reasonable subset of these methods/settings.
- **Optimal trade-off among FN, FP and the number of methods.**

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} FP(\mathcal{T}) + K_1 \cdot FN(\mathcal{T}) + K_2 \cdot \mathrm{card}(\mathcal{T}))$$

- Boolean Linear Programming selects ≈ 50 methods out of 10000 in 2 hours.
- **Segmentation results** for example:

$$FN_{BB} = 1.4\%, FP = 4307/\ 2\mathrm{Mpxl\ image}, (FN_{TS} = 0.5\%)$$

Learning segmentation

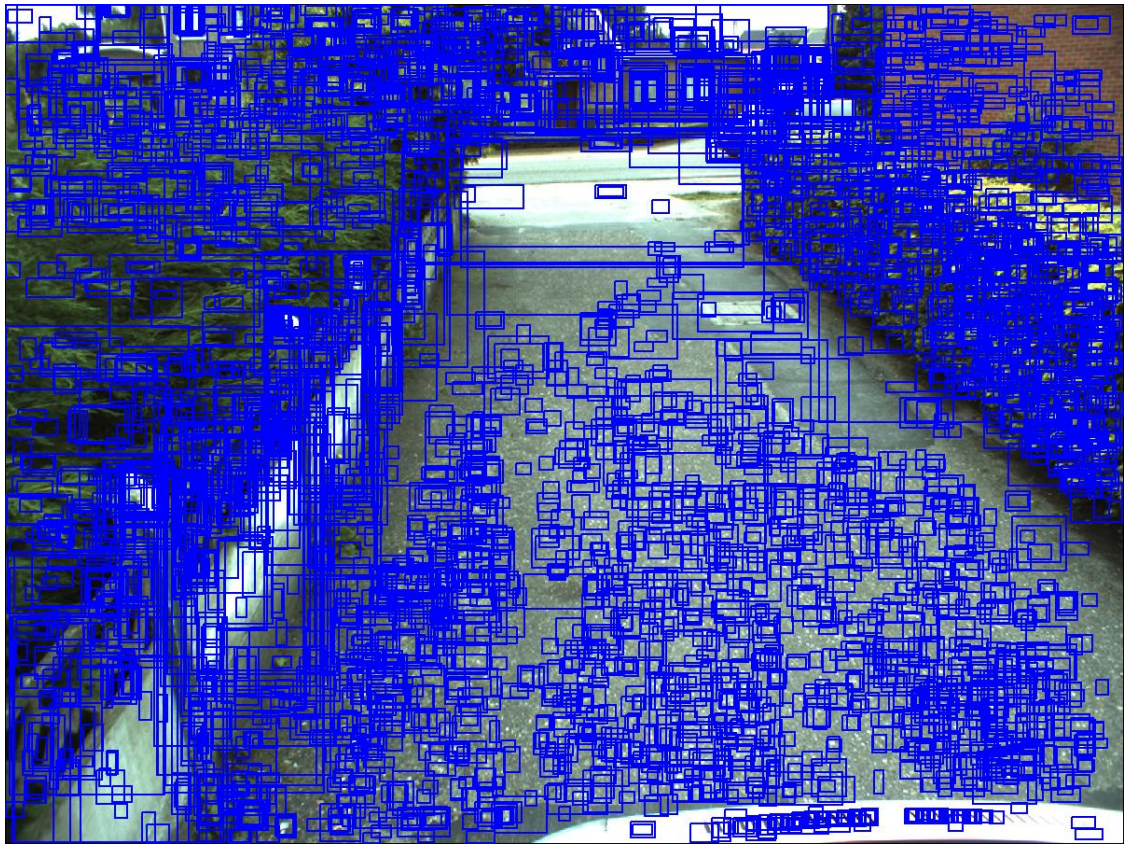
- There are thousands of possible setting of such methods e.g. different projection from color space.
- Learning is searching for a reasonable subset of these methods/settings.
- **Optimal trade-off among FN, FP and the number of methods.**

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} FP(\mathcal{T}) + K_1 \cdot FN(\mathcal{T}) + K_2 \cdot \mathrm{card}(\mathcal{T}))$$

- Boolean Linear Programming selects ≈ 50 methods out of 10000 in 2 hours.
- **Segmentation results** for example:

$$FN_{BB} = 1.4\%, FP = 4307/\ 2\mathrm{Mpxl\ image}, (FN_{TS} = 0.5\%)$$

How does the output of **segmentation** look like?





Detection

- **Detection:** suppresion of bounding boxes which does not look like a traffic sing.
  - Haar features computed on HSI channels selected (greedy construction by Adaboost).
  - Classifier is separated cascades of Adaboosts (5 different cascades for different shapes).
- **Detection (+segmentation) results** for example:

$$FN_{BB} = 3.9\%, FP = 30.6/\ 2\text{Mpxl image}, (FN_{TS} = 1.9\%)$$

$$FN_{BB} = 4.8\%, FP = 9.1/\ 2\text{Mpxl image}, (FN_{TS} = 2.6\%)$$

$$FN_{BB} = 7.5\%, FP = 1.1/\ 2\text{Mpxl image}, (FN_{TS} = 4.0\%)$$

Detection

- **Detection:** suppresion of bounding boxes which does not look like a traffic sing.
  - Haar features computed on HSI channels selected (greedy construction by Adaboost).
  - Classifier is separated cascades of Adaboosts (5 different cascades for different shapes).
- **Detection (+segmentation) results** for example:

$$FN_{BB} = 3.9\%, FP = 30.6/\ 2\text{Mpxl image}, (FN_{TS} = 1.9\%)$$

$$FN_{BB} = 4.8\%, FP = 9.1/\ 2\text{Mpxl image}, (FN_{TS} = 2.6\%)$$

$$FN_{BB} = 7.5\%, FP = 1.1/\ 2\text{Mpxl image}, (FN_{TS} = 4.0\%)$$

How does the output of **detection** look like?





2D optimization - introduction

- Single view detection and recognition is just preprocessing, the final decission is the subject of the global optimization over multiple views.
- The idea based on Minimum Description Length, i.e. explaining detected bounding boxes by the lowest number of real world traffic signs.
- If detections satisfy some *geometrical constraints*, than all of these detections are explainable by one real world traffic sign.



2D optimization - introduction

- Single view detection and recognition is just preprocessing, the final decission is the subject of the global optimization over multiple views.
- The idea based on Minimum Description Length, i.e. explaining detected bounding boxes by the lowest number of real world traffic signs.
- If detections satisfy some *geometrical constraints*, than all of these detections are explainable by one real world traffic sign.



2D optimization - introduction

- Single view detection and recognition is just preprocessing, the final decission is the subject of the global optimization over multiple views.
- The idea based on Minimum Description Length, i.e. explaining detected bounding boxes by the lowest number of real world traffic signs.
- If detections satisfy some *geometrical constraints*, than all of these detections are explainable by one real world traffic sign.

Geometrical constraints.



Geometrical constraints.



Geometrical constraints.

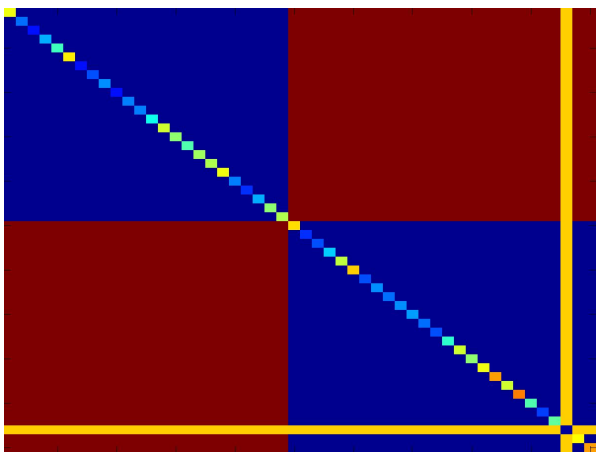


Geometrical constraints.



Problem formulation



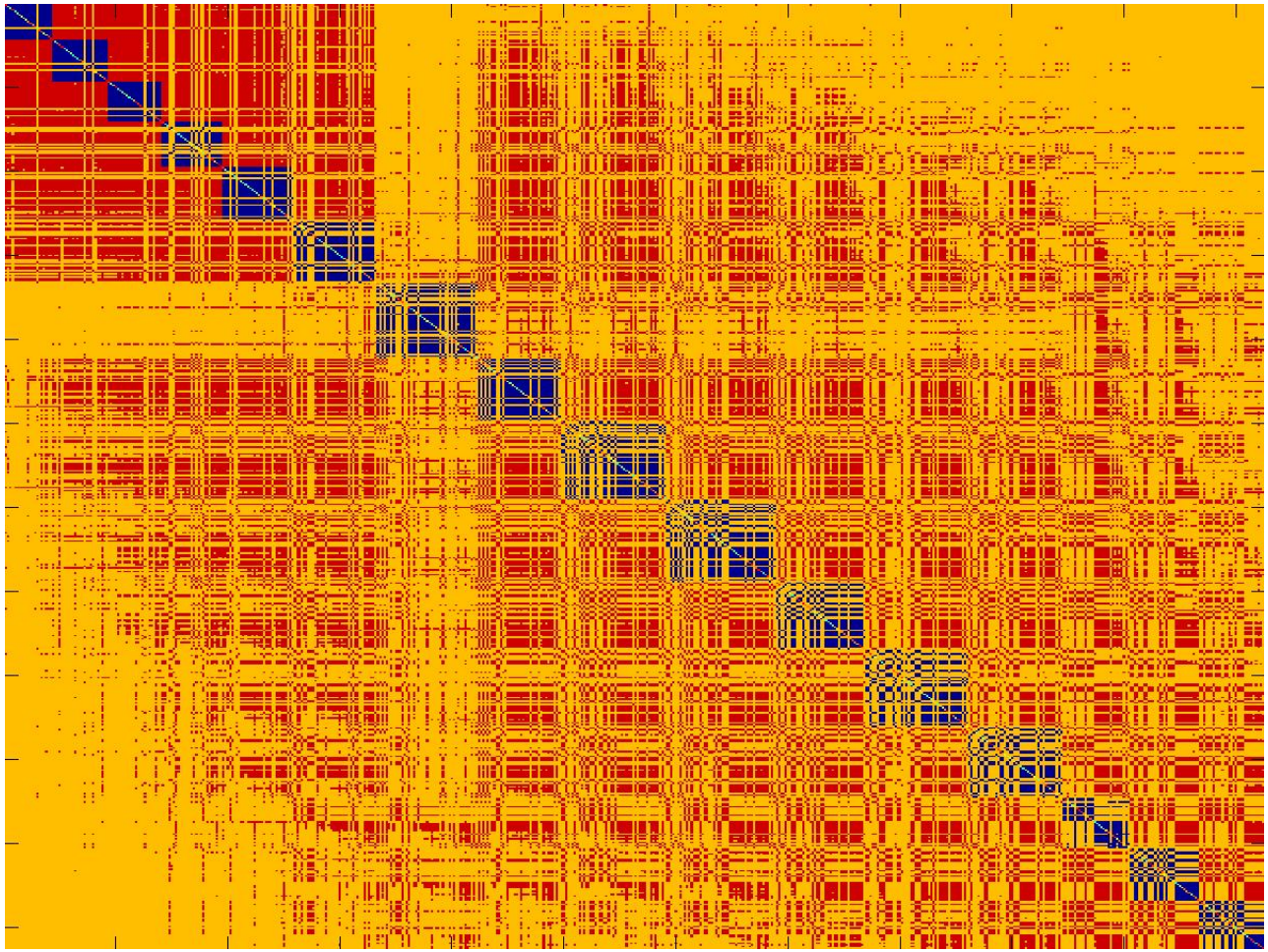


$$\max_{\mathbf{x} \in \{0,1\}} \mathbf{x}^\top \cdot \cdot \mathbf{x}$$

Example with 16 views



Example with 16 views



How does the **2D optimization** looks like?





3D optimization

- So far we just searched for geometrically consistent not too much mutually occluding clusters of detected bounding boxes.
- Now, we model traffic signs in 3D requiring consistency with all possible views.

3D optimization



3D optimization + recognition by voting

