WP4 - 2D train scanning

Francesca Odone
***DISI,* Università degli Studi di Genova**
odone@disi.unige.it

http://slipguru.disi.unige.it/

VIT Vision for Innovative Transport

CAPACITIES

The WP4 team

- **DISI**
  - Paolo Albini
  - Francesca Odone
  - Alessandro Verri
  - Luca Zini
- **IMAVIS**
  - Augusto Destrero
  - Alberto Lovato
- **DUNDEE**
  - Spela Ivekovic
  - Manuel Trucco

Outline of the presentation

- User requirements on 2D train scanning
- Technical issues and results:
  - Reconstruction of the train profile
  - U-Code reading
- Demo of the batch pipeline
- Current work beyond the VIT project

VIT Vision for Innovative Transport

CAPACITIES

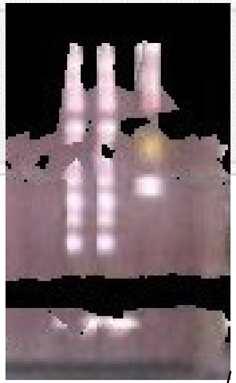
Main user requirements

- A feasibility study on
  - Reconstruction of the train profile (sequence of empty and filled wagons)
  - Generate a recording of the sequence of containers and their size
  - Reconstruction of the train can be done either whilst the train is coming to a stop (initial speed of 60 KM/h) or when it is stanting at the station

The feasibility study

- State of the art analysis
- Design of the architecture and tech specs
- Development and testing of the main modules
  - Laboratory tests: Months 9-12
  - Tests on Vado Ligure data: Months 13-18
- The final prototype is a *batch sw module* implementing the best choices with respect to current hw layout:
  - It processes a previously recorded video

Technical specifications



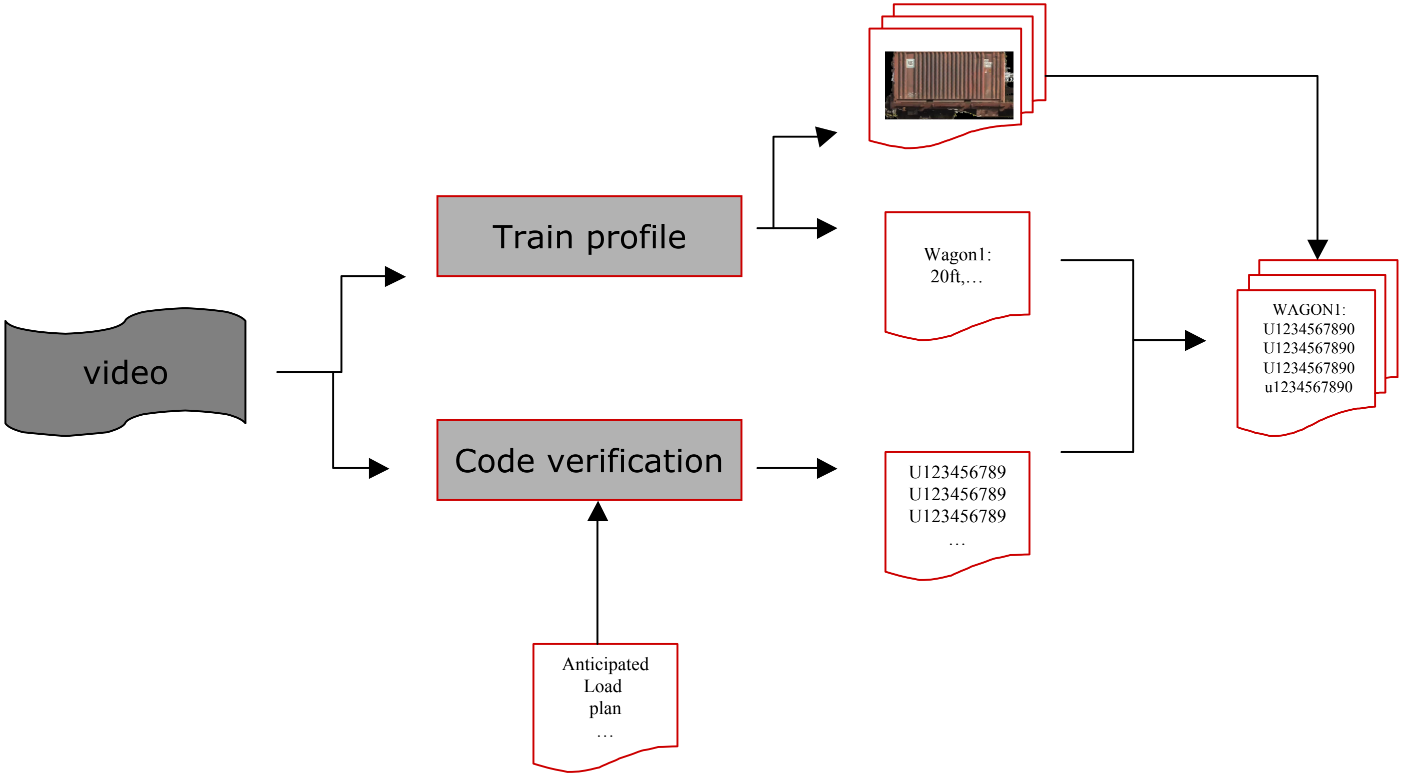
- The video-camera choice

  - Video-surveillance quality
  - Mega-pixel quality

Technical specifications

- One camera VS N cameras
  - Limit the amount of intervention on the plant
- Thus, the final prototype is based on the use of a single mega-pixel video-camera
- To comply to real-time computation (after an engineering phase)

Functional dependencies



Train profile

- A profile is built while the train is entering or leaving the station
  1. A panoramic image of the whole train is built
     - This allows us to automatically discard parts of the plant (turrets,…)
  2. Rectangle detection and gap detection is applied to the panorama for
     - Localization of the wagons
     - Identification of empty slots


SEZ 1
LC01
LC1

How to build a panorama

- Simple models to be able to cope with real-time processing
  - Background subtraction with a codebook model
  - Feature (corner) selection and tracking with a prior on the train motion direction (horizontal)
  - Image stitching

How to detect rectangles

- Line detection with classical computer vision methods:
  - Canny edge detection,
  - Hough transform
- Filter out “not horizontal” lines
- Use a prior on containers size to group 4-plets of lines that could be containers edges
- Discard the ones intersecting background zones

How to detect rectangles

How to detect rectangles

The procedure also allows us to associate information on the containers length

How to detect gaps

- Gaps are located by computing the integral of pixels belonging to the foreground with respect to the ground plane

A train at a glance

Experiments months 16-18

Results obtained with the camera in the final configuration.

Video sequences acquired in Vado Ligure; they include various weather and illumination conditions.

Camera Type	Containers error % (average)	Gaps error % (average)
MEGA-far	22%	8.4%
MEGA-close (month 17)	20.5%	3%
MEGA-close (month 18)	3.9%	0%

VIT Vision for Innovative Transport

CAPACITIES

Ownership code identification

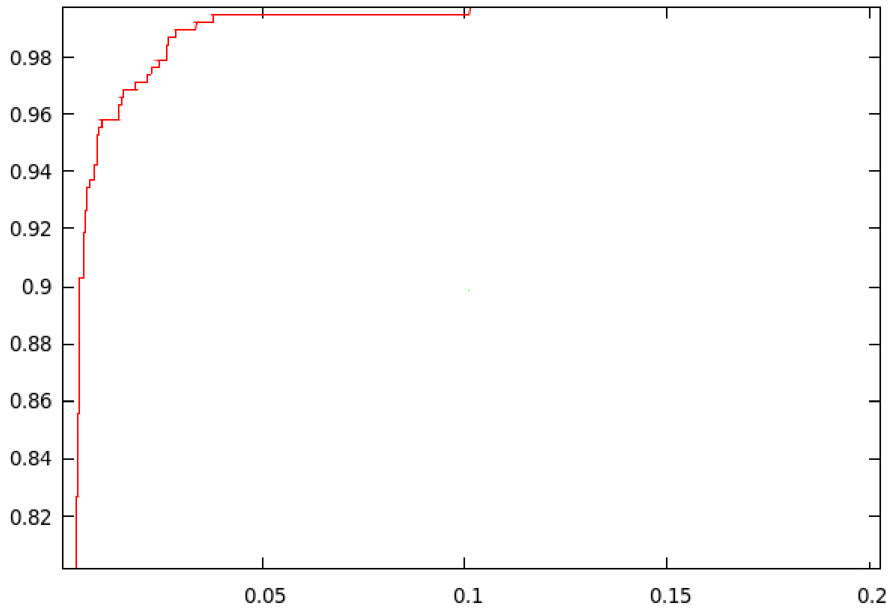
- For each video frame:
  - Character detection
  - Code verification
- For groups of adjacent video frames:
  - Output coherence

How to detect codes

- Text detection:
  - Segment the input frame into connected components (CC) with the Niblack algorithm
  - Discard the CCs too small or big
  - For each CC:
    - Represent it by means of an appropriate feature vector (area, perimeter, elongation, avg curvature, moments, …)
    - Classify the feature vector into text/non-text with a classification cascade *(learning from examples)*

Text detection: quantitative results

- Dataset for *lab* testing:
  - Training set acquired by the RTDs in various conditions
  - Test set from the SMEs



Code verification

- Code reading:
  - Multi-class classification:
    - RBF SVM classifier with a one-vs-all scheme
    - Model selection performed for each classifier with cross-validation
  - Geometry and vicinity are used to group character into strings
- Code verification:
  - We compare each code read with the expected code (Needleman-Wunsch comparison)
  - A tolerance to the number of correct chars is added with a remarkable improvement

Code verification: quantitative results

- Experiments on the choice of a tolerance

Tolerance	False negatives	False positives
0	2.91E-001	0.00E+000
1	4.17E-002	0.00E+000
2	3.59E-003	0.00E+000
3	2.15E-004	0.00E+000
4	5.00E-006	3.90E-005
5	0.00E+000	9.12E-002

- Tests at months 16-18:
  - manually annotated
  - False positives estimated simulating 10.000 random wrong codes

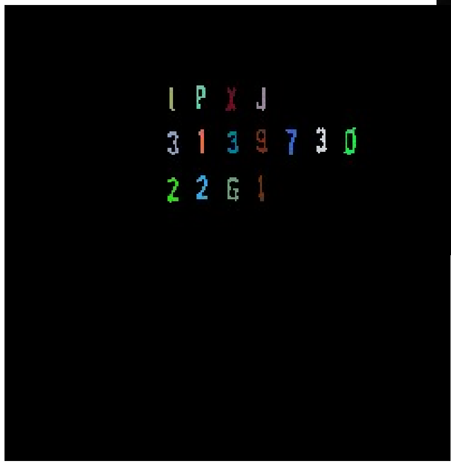
DATA (Vado L.)	FALSE POSITIVE	FALSE NEGATIVE
Month 16-17	0.042%	10.2%
Month 18	0.01%	4.3%

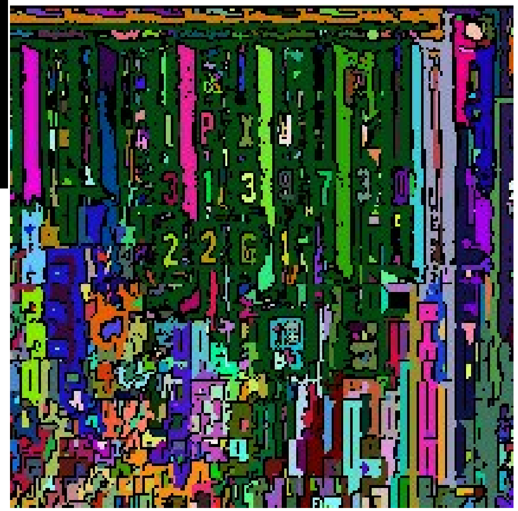
At the beginning of month 18 the
camera was
tuned and sharpened

The quality of the signal

BAD

GOOD





The full pipeline at work

....

Main user requirements

- A feasibility study on
  - ✓ Reconstruction of the train profile (sequence of empty and filled wagons)
  - ✓ Generate a recording of the sequence of containers and their size
  - ✓ Reconstruction of the train can be done either whilst the train is coming to a stop (initial speed of 60 KM/h) or when it is standing at the station

What now

- **At month 18** the software was working as a batch module on a video input
- The module was **already compatible** with the video-surveillance software suite developed in WP5

- Following the **SMEs positive comments** to the result of the feasibility study we are currently integrating it to the video-surveillance server
- **Ongoing** laboratory tests :
  - Feature extraction and tracking
  - Container code detection

VIT Vision for Innovative Transport

CAPACITIES