Video Surveillance for Road Traffic Monitoring

5th Workshop on Road Traffic Monitoring

Marc Grau
Ignasi Mas
Hugo Prol
Jordi Puyoles

Introduction

Why Road Traffic Monitoring?

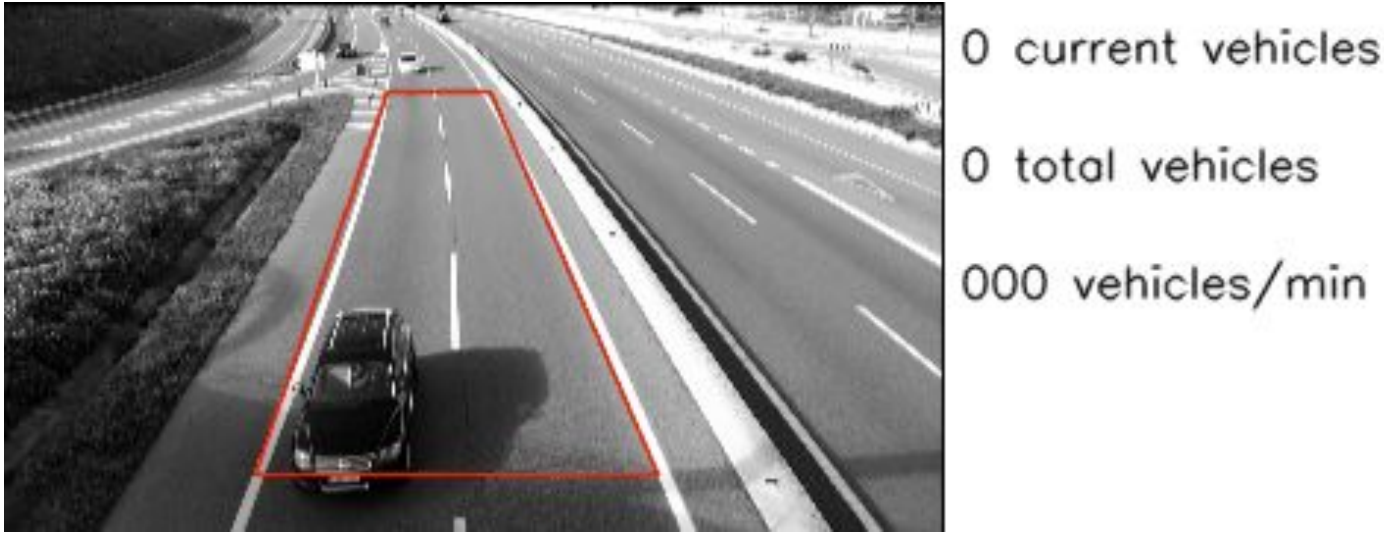
Introduction

Why Road Traffic Monitoring?

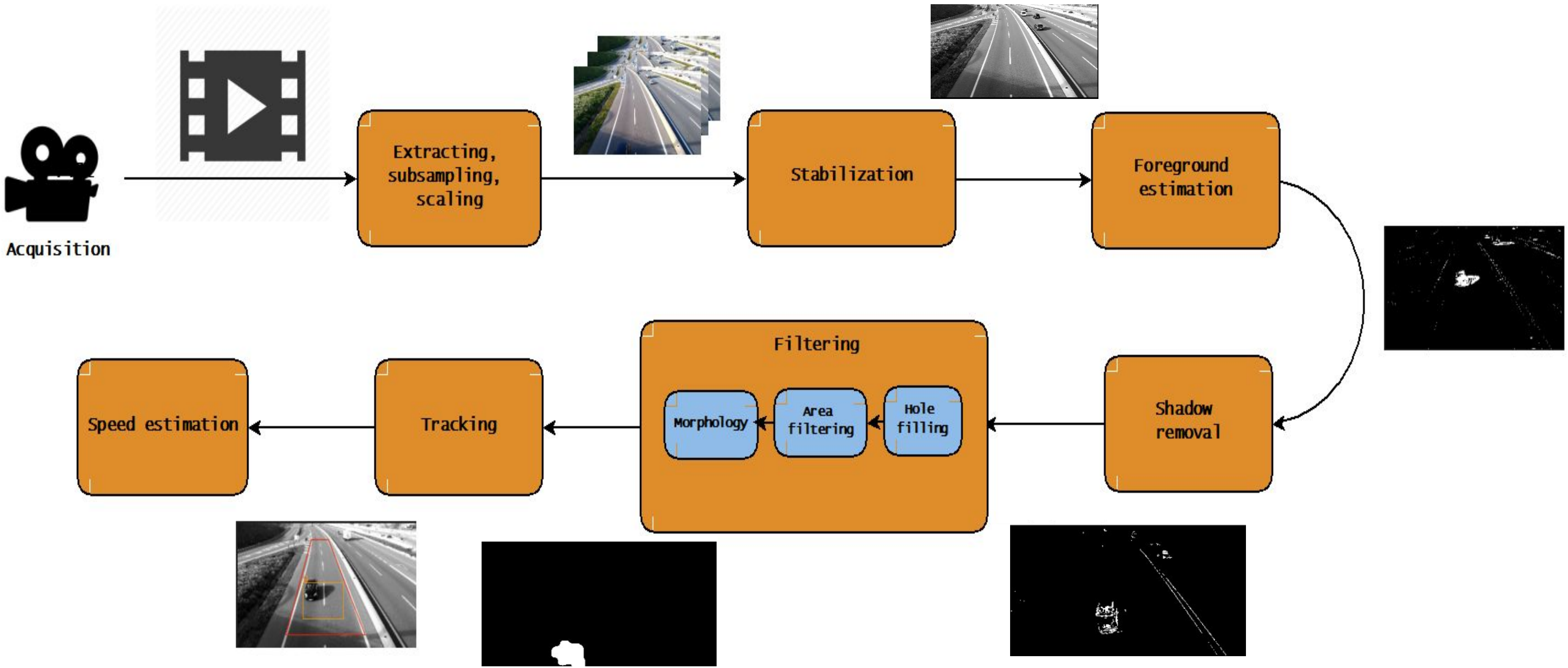
src: [Traffic Technology Today]

Introduction

- Increase road safety in zones with accident risk
- Control road density to avoid jams



Pipeline


Acquisition
Extracting, subsampling, scaling
Stabilization
Foreground estimation
Shadow removal
Filtering
Hole filling
Area filtering
Morphology
Tracking
Speed estimation

Datasets

Highway

- Train: Frames 1050 to 1200
- Test: Frames 1200 to 1350

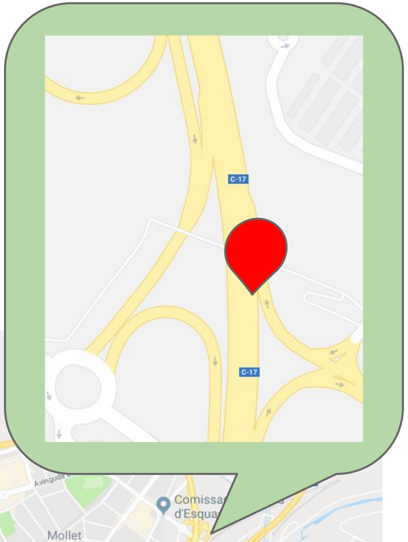
Traffic

- Train: Frames 950 to 1000
- Test: Frames 1000 to 1050

Datasets

Location:

· C-17 (Mollet del Vallès, Barcelona)
· Coordinates: 41.544427, 2.224059



Sequence 2 (Train: 3500 -3660, Test: 3660 - 4400)

Sequence 3 (Train: 4800 4900, Test: 4900 - 5664)

Acquisition Specs:

- FPS: 30
- 1920 x 1080 px

Extraction Specs:

- Frame Sampling: ½
- Pixel Sampling: 1/6

Analysis:

- Hard Shadows
- Low Traffic
- Two lanes
- Road Exit
- Unstabilized

Video stabilization

Issue: Original video suffers from some jitter!

Video stabilization

Issue: Original video suffers from some jitter!

Effect: Noise on road resulting in FP in foreground estimation

Video stabilization

Issue: Original video suffers from some jitter!

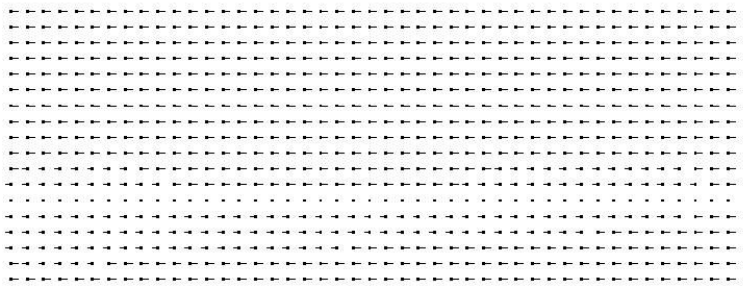
Solution: Video stabilization with optical flow through block matching

Video stabilization

Issue: Original video suffers from some jitter!

Solution: Video stabilization with optical flow through block matching

- **Forward** compensation

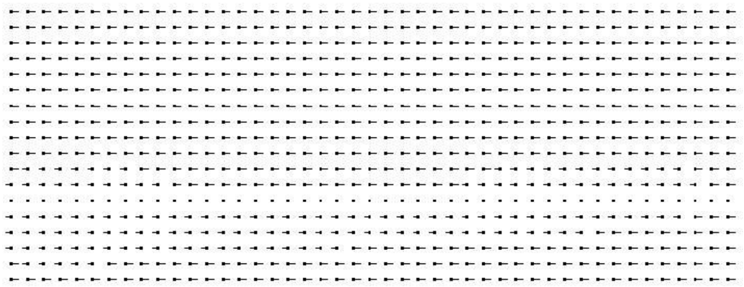


Video stabilization

Issue: Original video suffers from some jitter!

Solution: Video stabilization with optical flow through block matching

- **Forward** compensation
- Blocks of 16 px



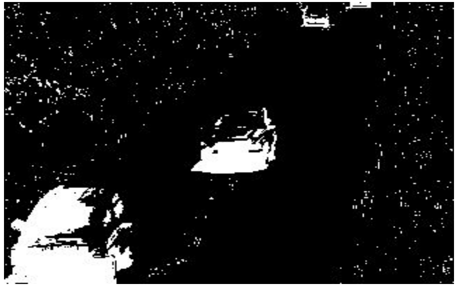
Video stabilization

Issue: Original video suffers from some jitter!

Solution: Video stabilization with optical flow through block matching

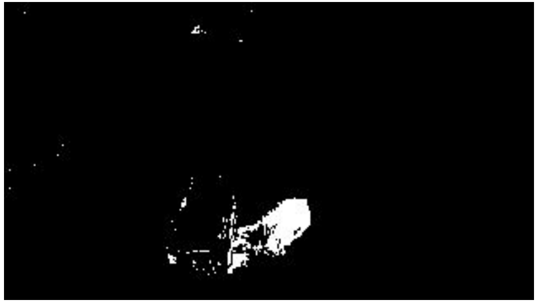
Foreground subtraction

- Adaptive modelling: validated model from previous project stages



Highway: non-adaptive

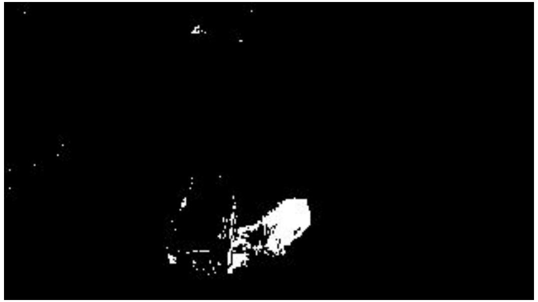
Highway: adaptive



Sensible difference with non-adaptive model (left). Example:highway

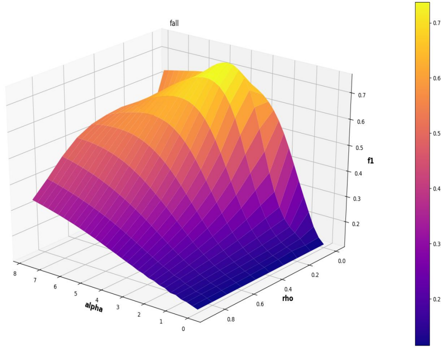
Foreground subtraction

- Adaptive modelling: validated model from previous project stages
- Parameters:
    - α
    - ρ

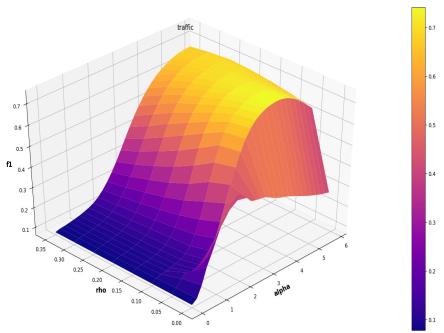


Foreground subtraction

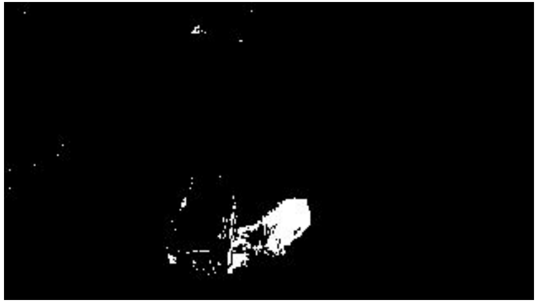
- Adaptive modelling: validated model from previous project stages
- Parameters:
  - α
  - ρ
- Parameter selection:
  - Start from parameters selected on previous stages with other (training) datasets: highway, fall traffic. They were selected by grid search
  - Manually adjust them guided by visual qualitative results



α

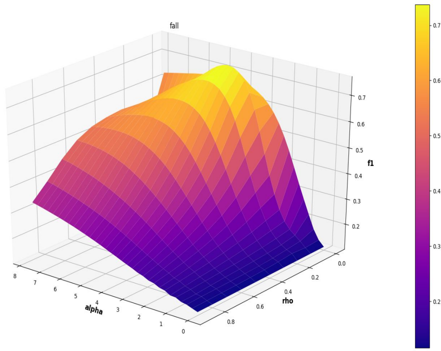


ρ

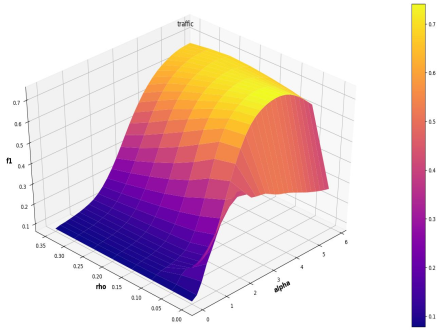


Foreground subtraction

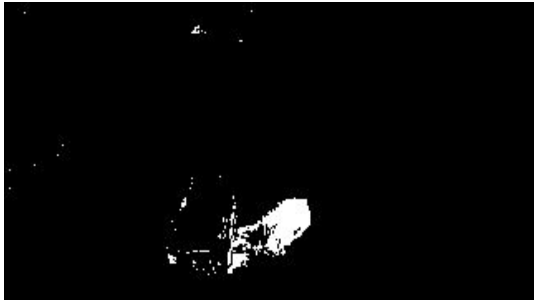
- Adaptive modelling: validated model from previous project stages
- Parameters:
    - α
    - ρ
- Parameter selection:
    - Start from parameters selected on previous stages with other (training) datasets: highway, fall traffic. They were selected by grid search
    - Manually adjust them guided by visual qualitative results



α



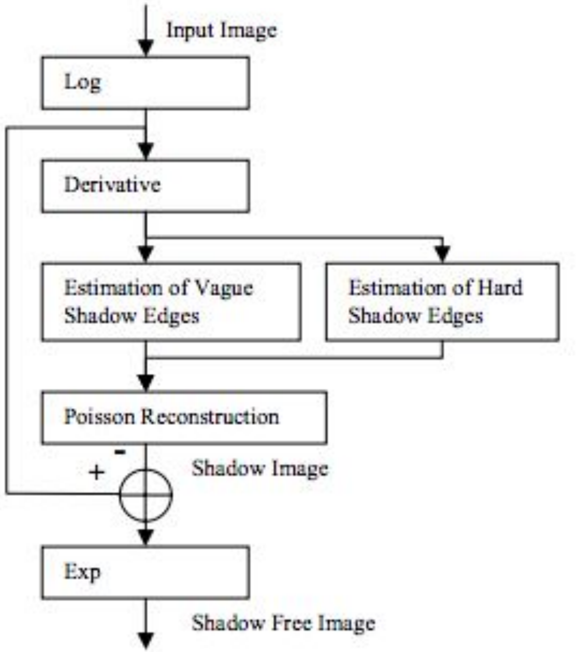
ρ



We still have much noise to filter!!

Shadow Removal

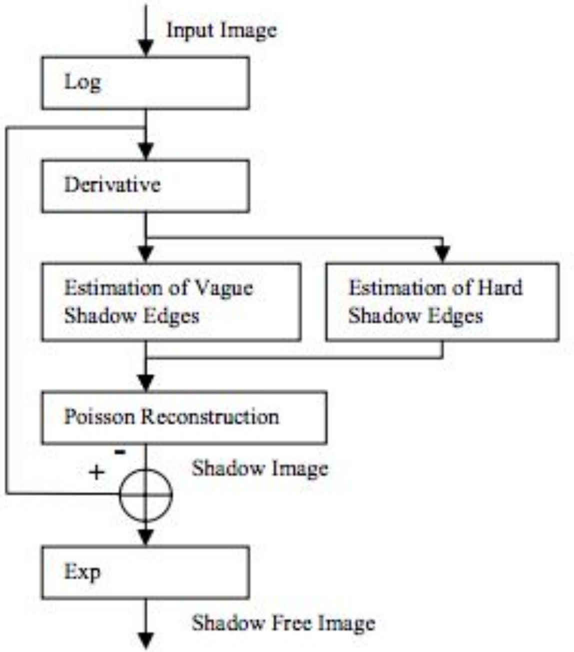
Based on Xu & Others
'Shadow Removal from a Single Image'



- Classification is applied to the derivatives of the input image to separate the vague shadows
- Exploit color invariance to distinguish hard shadow edges from material edges
- Illumination image obtained via solving the standard Poisson Equation

Shadow Removal

Based on Xu & Others
‘Shadow Removal from a Single Image’

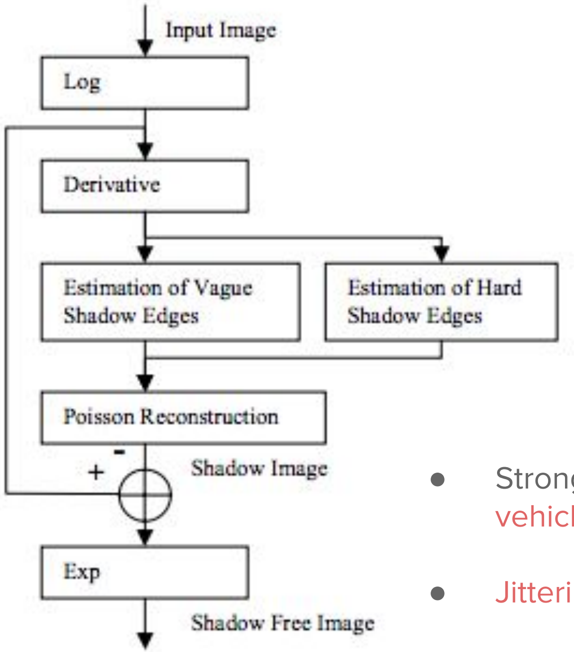


Original
Prediction

- Hard Shadow with very large size
- Cars passing together can fail as a single blob.
- Additional Noise due to jittering

Shadow Removal

Based on Xu & Others
'Shadow Removal from a Single Image'



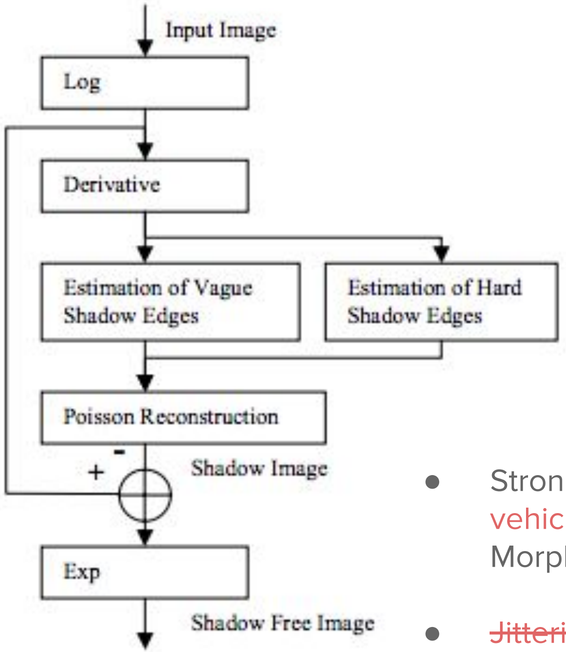
Original Prediction

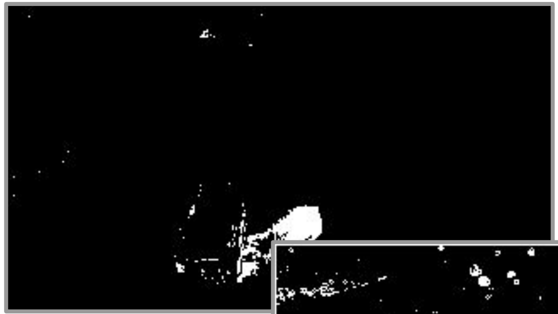
Shadow free image obtained from the original sequence

- Strong shadow removal works properly but vehicle areas highly decrease
- Jittering noise persists

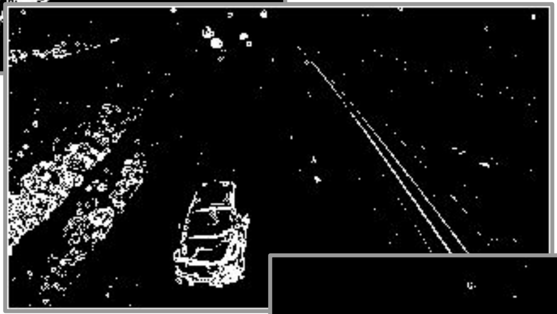
Shadow Removal

Based on Xu & Others
'Shadow Removal from a Single Image'





Original Prediction

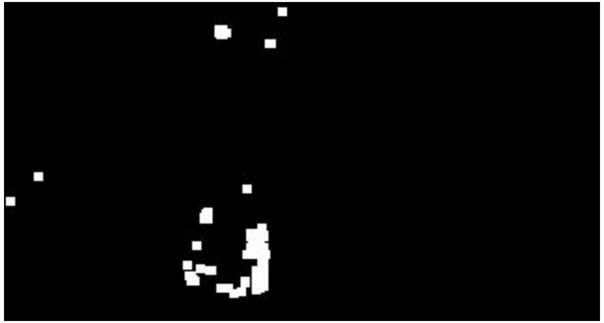


Shadow free image obtained from the original sequence

Shadow free prediction

- Strong shadow removal works properly but vehicle areas highly decrease → Further use of Morphology to recover object shape
- ~~Jittering noise persists~~ → Masking with original prediction removed most noise

Filtering

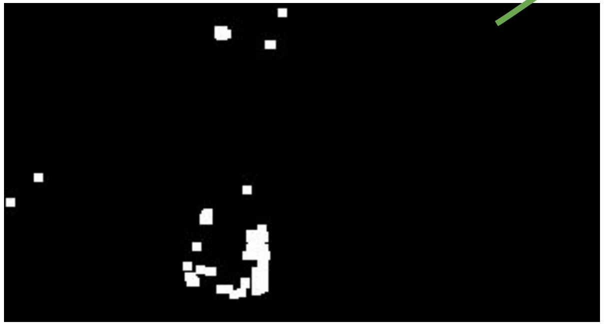


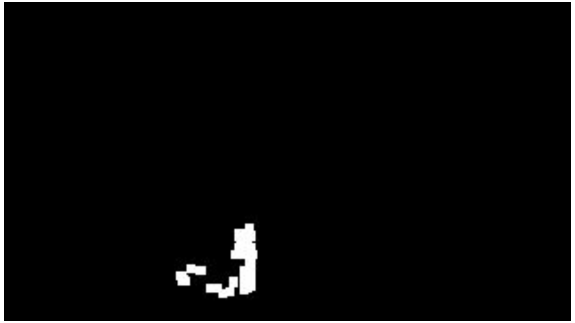
Dilation (5,5)

Dilation (5 , 5)

- Goal:
  - Increase vehicles shape for further hole filling and area filtering
- Observations:
  - Small noise increased 5 pixels
  - Dense areas (vehicles) are merged

Filtering


Dilation (5,5)


Imfilling (8 conn) & Clean (4 conn)

Hole filling (4 connectivity)

- Goal:
  - Remove loss of pixels in objects to have the real area of the objects
- Observations:
  - 8-connectivity

Area filtering (4 connectivity)

- Goal:
  - Remove objects smaller than cars (typically noise)
- Observations:
  - Size: smaller than observed from cars

Filtering

Dilation (5,5)

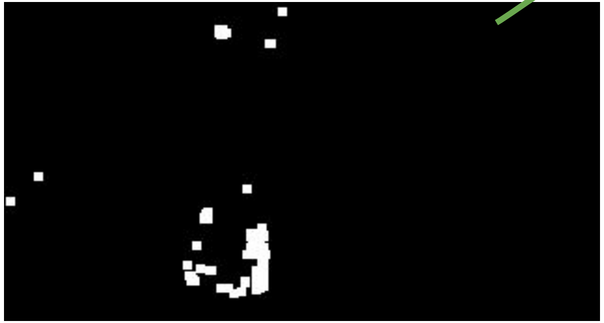
Imfilling (8 conn) & Clean (4 conn)

Closing (5,5)

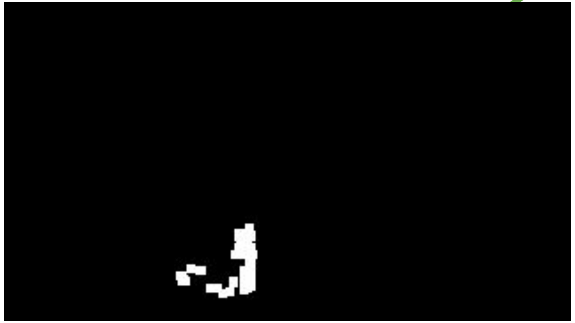
Closing (5 , 5)

- Goal:
  - Increase area inside the vehicles (merge broken parts)
- Observations:
  - Elliptical structuring element
  - Slight improvement

Filtering



Dilation (5,5)



Imfilling (8 conn) & Clean (4 conn)

Closing (5,5)

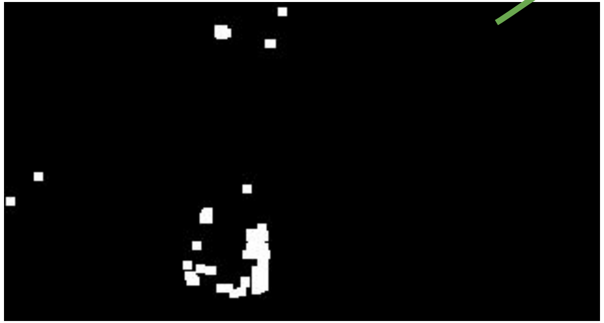
Opening (14 , 14)

- Goal:
  - Improve vehicle shape and remove unwanted objects

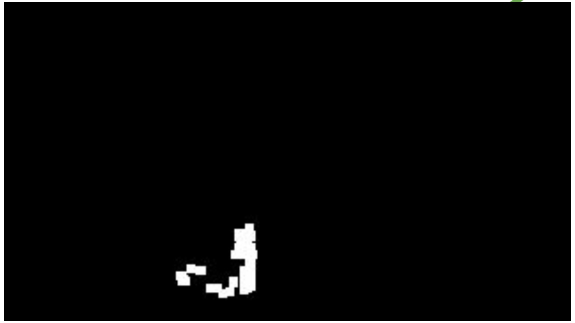
- Observations:
  - Uniform shapes easier to track
  - Adjacent objects not merged
  - Far vehicles are lost

Opening (5,5)

Filtering



Dilation (5,5)



Imfilling (8 conn) & Clean (4 conn)

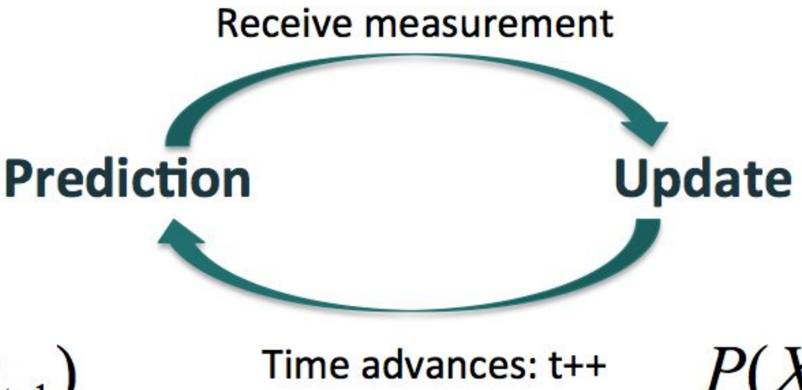
Closing (5,5)

Vehicle shapes ready for tracking!

Dilation (7,7)

Opening (5,5)

Tracking: Kalman filters



$$P(X_t \mid z_0, \ldots, z_{t-1})$$

$$\mu_t^-, \sigma_t^-$$

Mean and std. dev. of predicted state:

$$P(X_t \mid z_0, \ldots, z_t)$$

$$\mu_t^+, \sigma_t^+$$

Mean and std. dev. of predicted state:

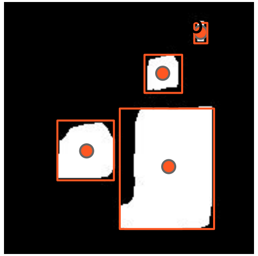
Slide credit: Kristen Grauman

Tracking: Kalman filters

Steps for t=K

- Extract blob centroids for t=K

frame t=K

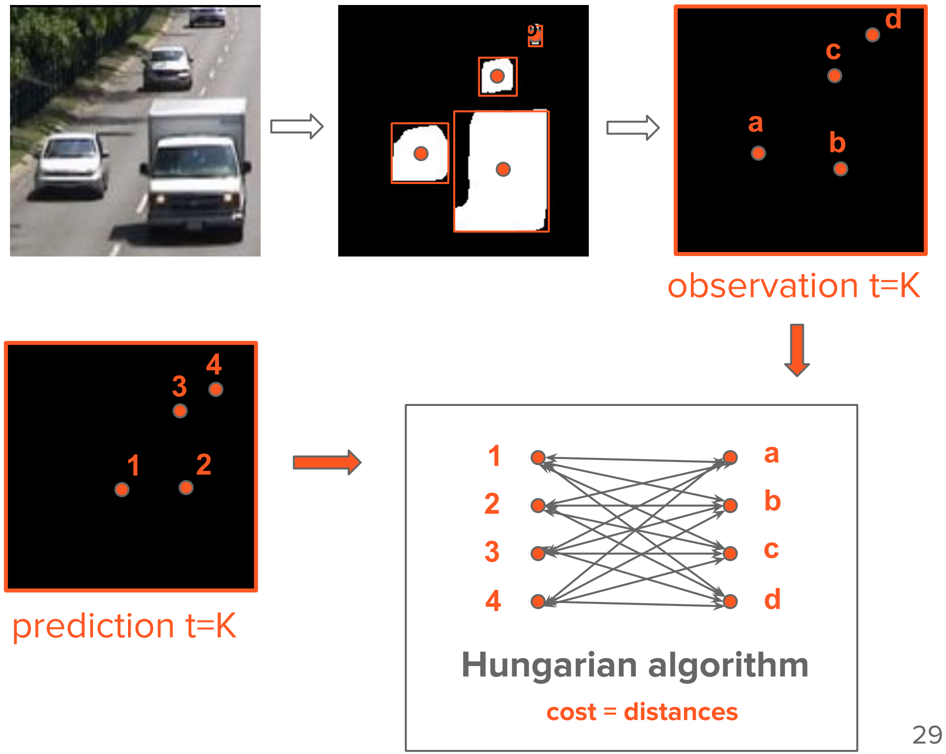


observation t=K

Tracking: Kalman filters

Steps for t=K

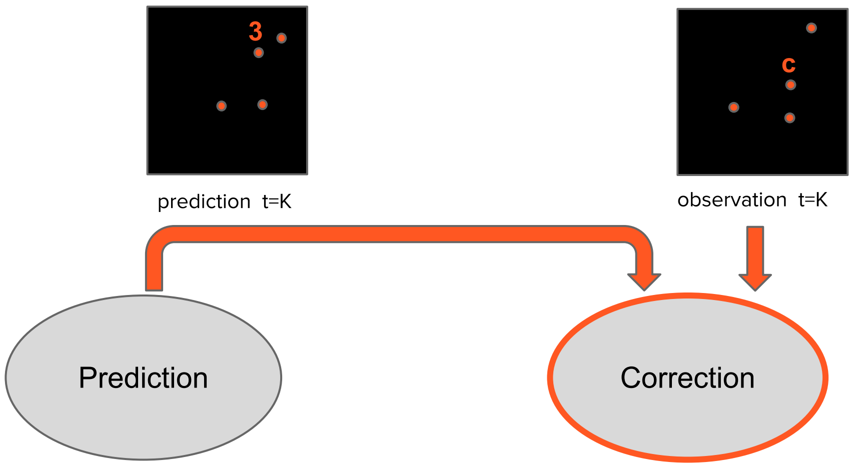
- Extract blob centroids for t=K
- **Assign blob centroids to predicted positions**



Tracking: Kalman filters

Steps for t=K

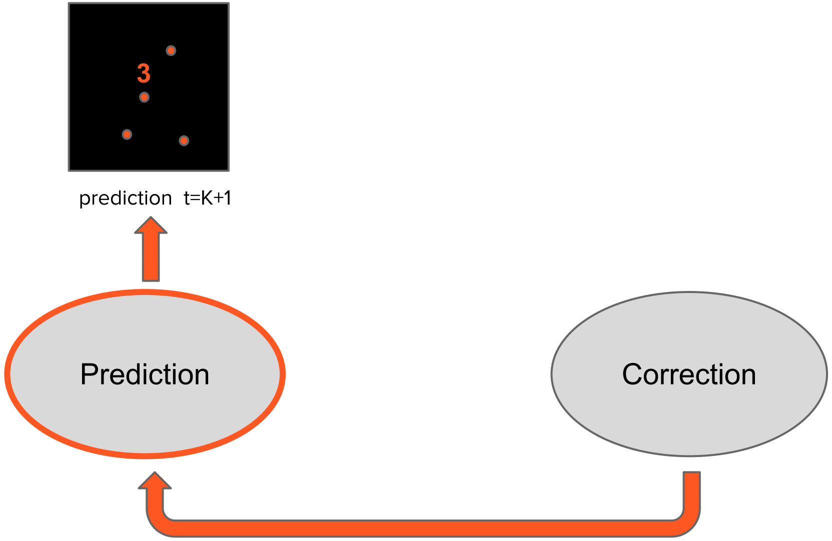
- Extract blob centroids for t=K
- Assign blob centroids to predicted positions
- **Correct predictions with blob centroids**



Tracking: Kalman filters

Steps for t=K

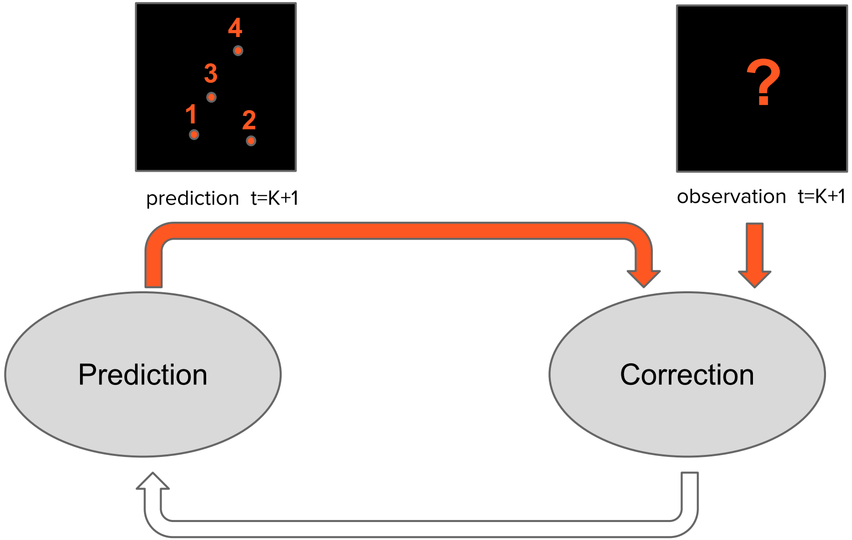
- Extract blob centroids for t=K
- Assign blob centroids to predicted positions
- Correct predictions with blob centroids
- **Predict new vehicle position (t=K+1)**



Tracking: Kalman filters

Steps for t=K

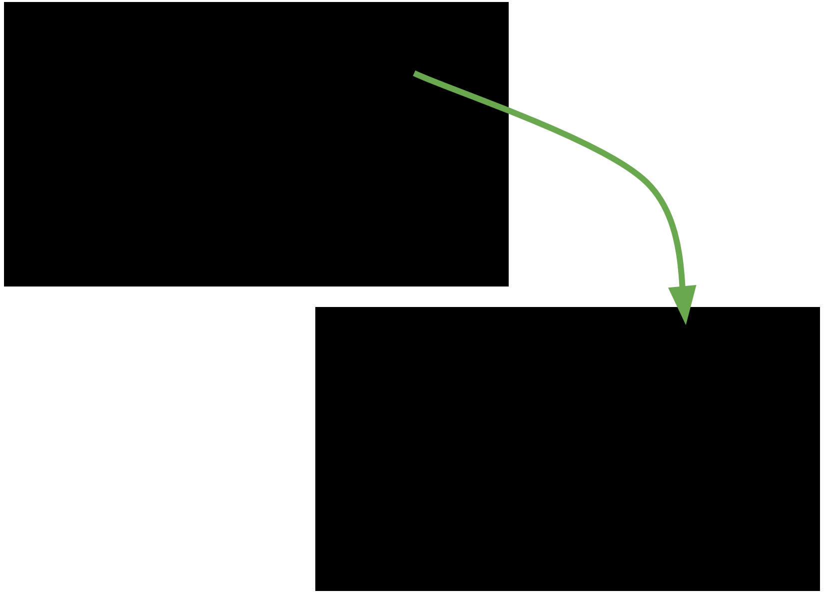
- Extract blob centroids for t=K
- Assign blob centroids to predicted positions
- Correct predictions with blob centroids
- Predict new vehicle position (t=K+1)
- **Iterate from the beginning for t=K+1**



Tracking: Kalman filters

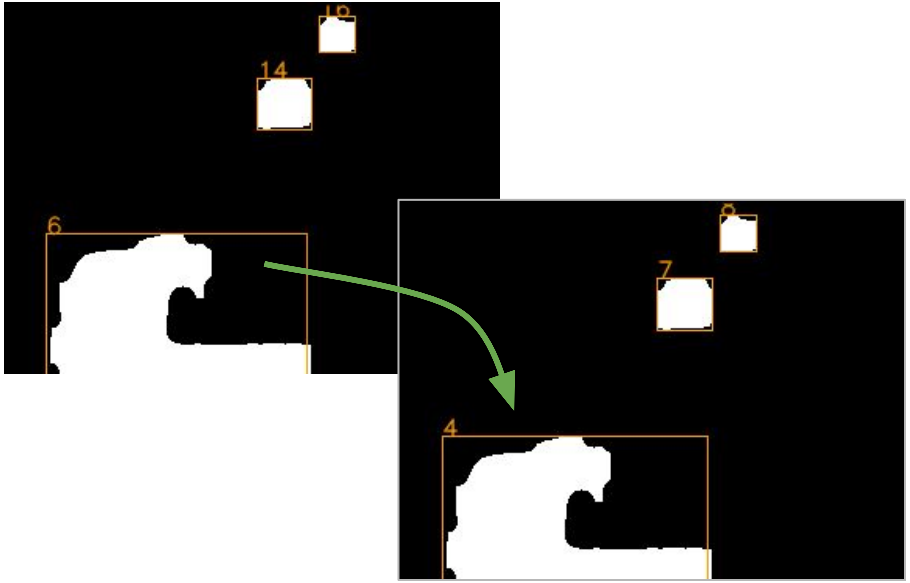
Refinements!

mismatches in Hungarian algorithm



max distance constraint

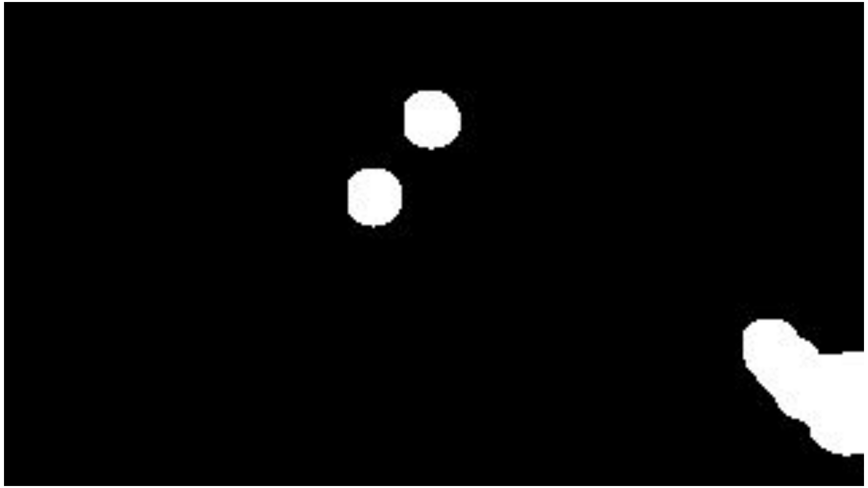
false positive due to noise



hide tracks during N frames

Tracking: Kalman filters

Results



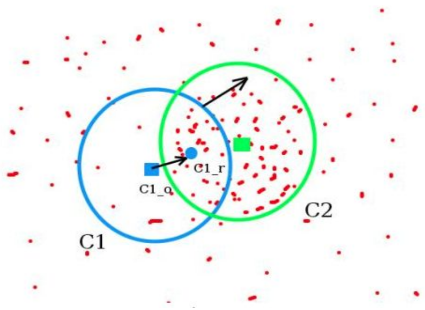
- good tracking in both directions
- kalman prediction in 10 and 13

- no blobs generated for trucks
- false positive 14

Tracking: Mean-Shift Filters

Mean shift based on [OpenCV with Python Blueprints](), Chapter 5.

Given an initial window (blue circle), the algorithm proceeds to move the window towards the area of maximum pixel density (green circle).



src: OpenCV Meanshift

Traffic dataset

Highway dataset

- Fast detection of vehicles
- Satisfactory tracking at nearby objects
- Bounding Box filtered by area to reduce false positives
- Highly variable bounding box size (difficult to compute speed)
- Multiple objects tracked in the same region ➔ Need to posterior filtering

Speed estimation

From kinematics, we know that, assuming that there is **no acceleration**, $v = \frac{\Delta x}{t}$

Therefore, we must estimate the distance traveled for each car in a known period of time in order to get an approximation of the real velocity.

First of all, we reduce the effect of projective distortion using homographies:

Rectified traffic

Rectified highway

Speed estimation

- Pixel to meter ratio is computed **manually** in the rectified image
- Final conversion is done with the following expression:

$$\frac{km}{h} = 3.6\frac{s}{h}\frac{km}{m}\frac{\Delta pixels}{nº frames}\frac{frames}{s}\frac{m}{pixels}$$

Where:

- frame/s corresponds to the camera frame rate
- meters/pix is the ratio manually measured

- Under this approach, we are assuming that **cars are part of the road**. Since this is not true, an **error** in the computation **is introduced** when projecting the centroid of the detection to the warped space.
- **Distance** between consecutive detections is projected from normal space to warped space
- **Variability** on the **size of the bounding boxes** is also a source of error, since influences the displacement of the centroid

Speed estimation

- We’ve assumed a frame rate of **30fps** in both highway and traffic and that **road lines follow** the **US federal guidelines** in order to compute the pixel to meter ratio
- Final speed is computed using the median of the observed. Outliers are removed using a standard deviation threshold

Results on **highway** (ID: appearance order)**:**

ID	Speed (km/h)
1	78.78
2	83.5
3	106.4

ID	Speed (km/h)
4	75.9
5	73.25
6	84.77

Tracking using kalman filters on Highway

Speed estimation

Results on **traffic** (ID: appearance order)**:**

ID	Speed (km/h)
1	61.38
2	190.7
3	64.50
4	81.79

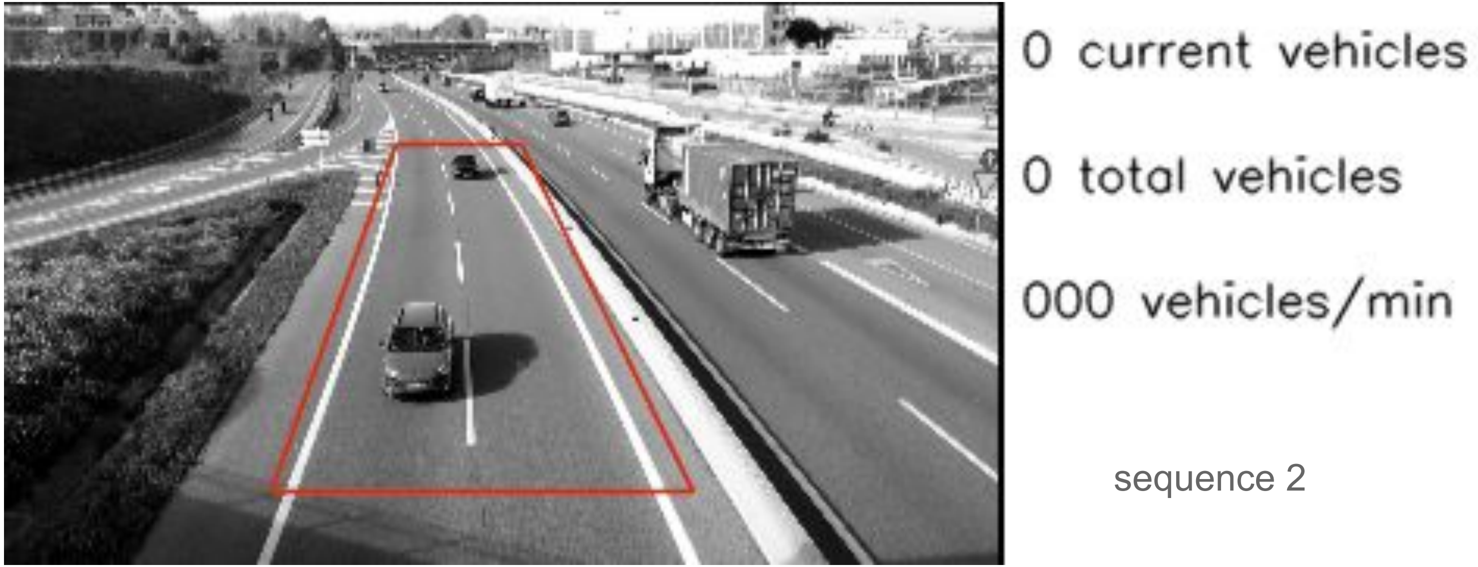
ID	Speed (km/h)
5	100.20
6	71.7
7	84.2
8	99.9

ID	Speed (km/h)
9	100.20
10	87.15
11	111.4
12	100.0

Tracking using kalman filters on stabilized Traffic

Note than speeds for Car 2 in traffic and Car 3 in highway are slightly different from the other values. This is due to the merging of the tracked bounding box with a near one

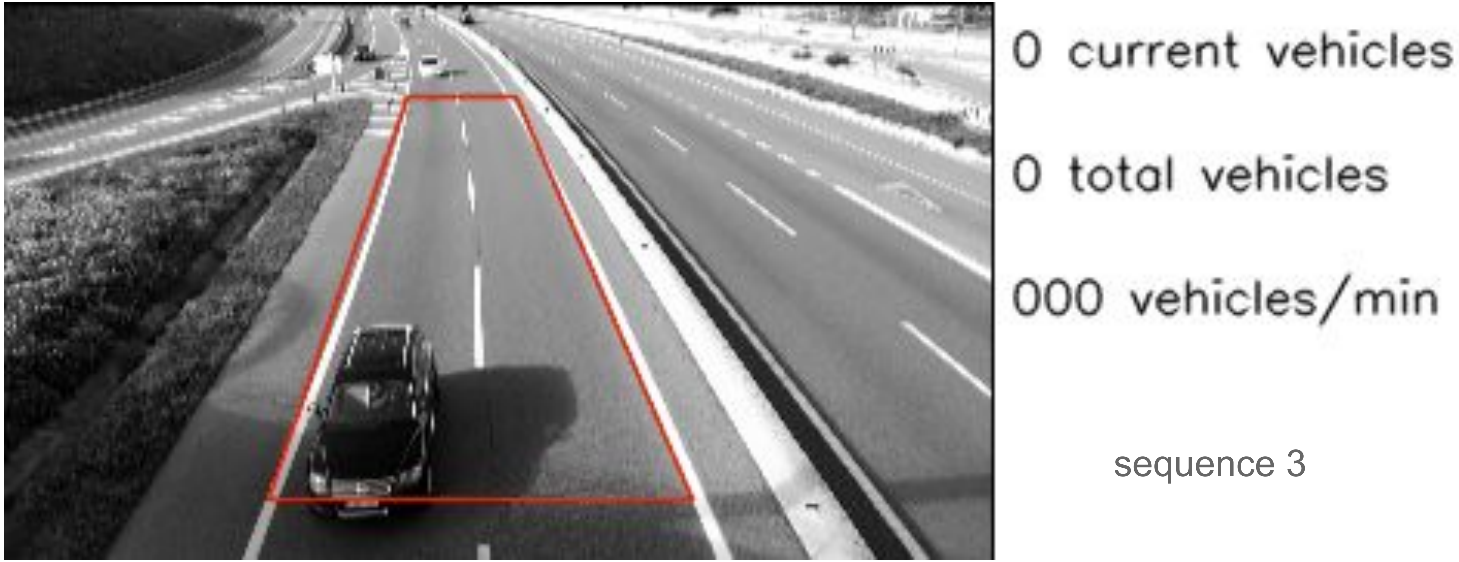
Application



sequence 2

- White cars result on later tracking due to filtering stage
- Delimited area to reduce false positives

Application



sequence 3

- Cars moving closer are more difficult to track.

Application

Possible improvements on the **implementation**:

- Add robustness against false positives by comparing the estimated motion of an object with other estimations
- Refine shadow removal:
  - Close cars could be considered as one due to a possible connection between their shadow
  - Due to hard shadows in the sequences, shadow removal eliminates a great number of non shadowed pixels. A combination of less aggressive methods could improve the results
- Filter strong displacements of the bounding box in order to minimize its effect on the speed computation
- Images should be taken from a much higher position in order to minimize the projectivity effect

Conclusions

- Video stabilization can be very profitable to avoid noise in further stages caused by jittering effects.

- Adaptive model for background subtraction: easy to implement, yet very powerful method.

- Vehicle tracking properly dealt with hard shadows (shadow detection)

- In general good performance for Kalman filters, although adjacent objects difficult to track separately (morphology). Try more complex models with Kalman.

Further Work

- Detection of driving infractions, as:
  - Identify vehicles passing others through the right lane.
  - Identify vehicles driving through the left lane when the right is empty.
- Collision avoidance
- Traffic jams prediction

Thank you!