Detecting Drops On Lens in Wireless Multimedia Sensor Network Nodes

Cesare Alippi, Giacomo Boracchi, Manuel Roveri

Politecnico di Milano

7 November 2009

ROSE 2009, Lecco

Outline

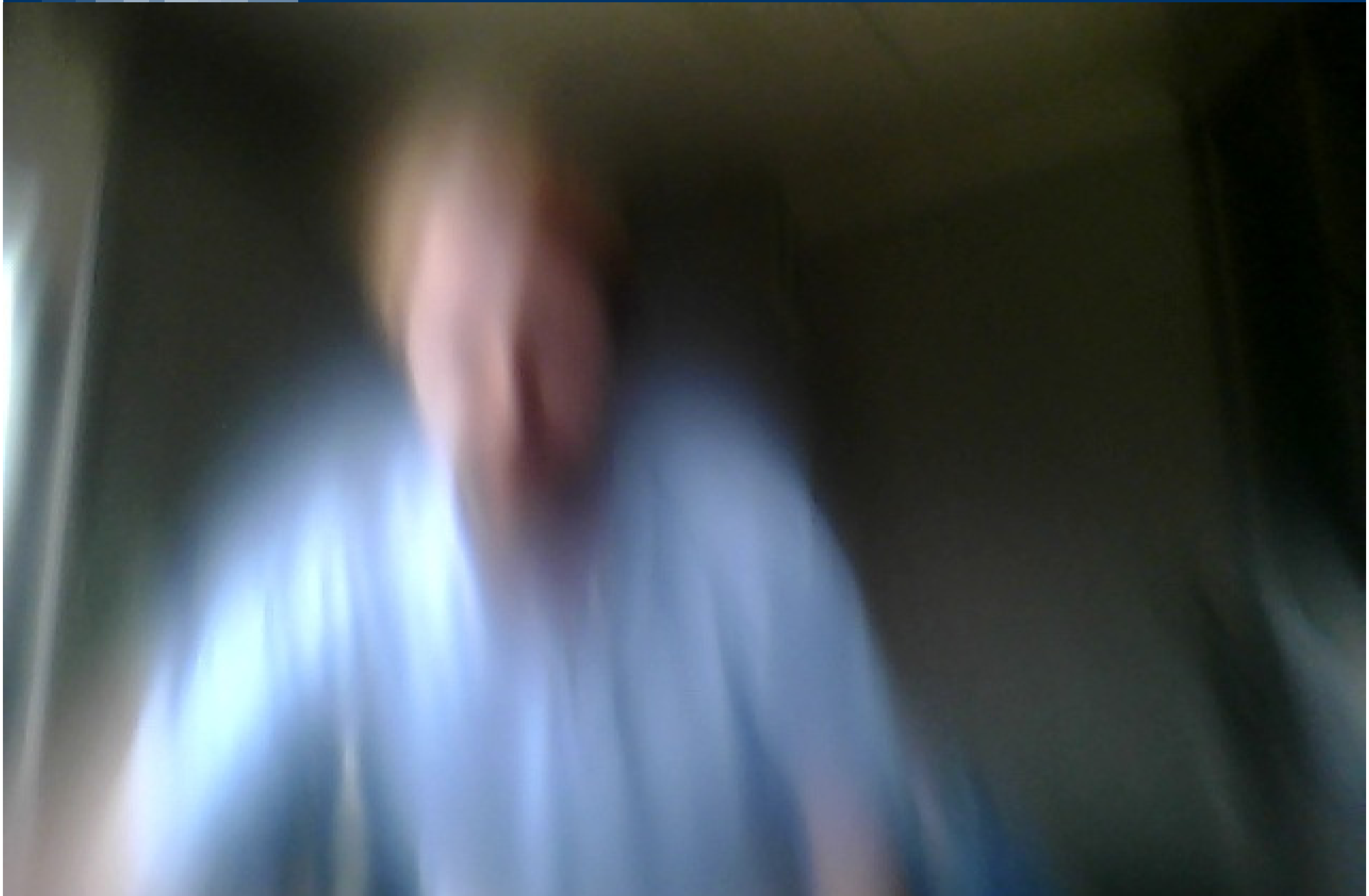
- The issue
- Our approach
- The observation model
- The blur measure
- The change detection test
- Experiments
- Concluding remarks

The issue

- We consider **Wireless Multimedia Sensor Networks (WMSN)** used for **monitoring outdoor environment**.
- The nodes (or the network) should then be able to determine when there is some **structural information loss** in the the **image acquisition system**
- In particular we consider the degradation induced by **drops** on the **camera lens**, as this may result because of rain, humidity, waves…

What’s up when a rain drop falls on camera lens

What's up when a rain drop falls on camera lens



What's up when a rain drop falls on camera lens

The issue

- We want to provide the network with the capabilities to **determine when** there is a drop on camera lens.

The issue

- We want to provide the network with the capabilities to **determine when** there is a drop on camera lens.

$$\{z_i\}_{i=1,\dots,N}$$

The issue

- We want to provide the network with the capabilities to **determine when** there is a drop on camera lens.

$$\{z_i\}_{i=1,\ldots,N}$$

The issue

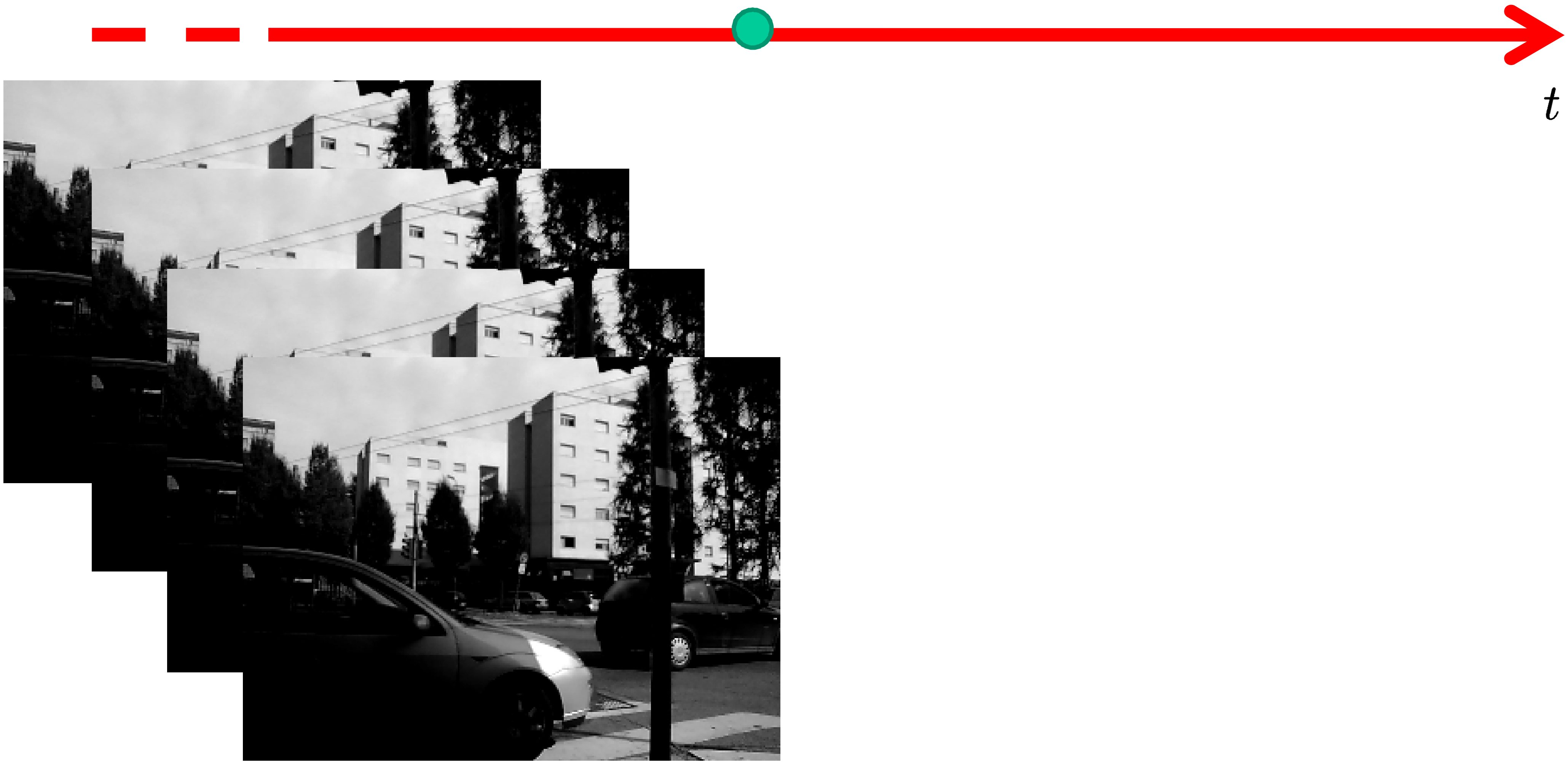
- We want to provide the network with the capabilities to **determine when** there is a drop on camera lens.

$$\{z_i\}_{i=1,\ldots,N}$$

The issue

- We want to provide the network with the capabilities to **determine when** there is a drop on camera lens.

$\{z_i\}_{i=1,\dots,N}$



The issue

- We want to provide the network with the capabilities to **determine when** there is a drop on camera lens.

$$\{z_i\}_{i=1,\ldots,N}$$

t

The issue

- We want to provide the network with the capabilities to **determine when** there is a drop on camera lens.

$$\{z_i\}_{i=1,\ldots,N}$$

The issue

- We want to provide the network with the capabilities to **determine when** there is a drop on camera lens.

$$\{z_i\}_{i=1,\ldots,N}$$

The issue

- We want to provide the network with the capabilities to **determine when** there is a drop on camera lens.

$$\{z_i\}_{i=1,\dots,N}$$

The issue

- We want to provide the network with the capabilities to **determine when** there is a drop on camera lens.

$\{z_i\}_{i=1,\dots,N}$ $\qquad \bar{t}$

The issue

WSN Constraints:

- Images have to be **processed locally**
  - In order to avoid **sending huge amount of data** on the network
  - Thus processing must have **low computational complexity**

- **Do not assume stationarity** in the observed scene,
  - frames could have been acquired at very different instants as the image acquisition is not continuous
  - we have **no a priori information** about the scene

Our Approach

- Drops on camera lens are modeled as a **blur operator**.

- We combine
  - a **low-complexity blur measure**
  - a **sophisticated change detection test** on these measures

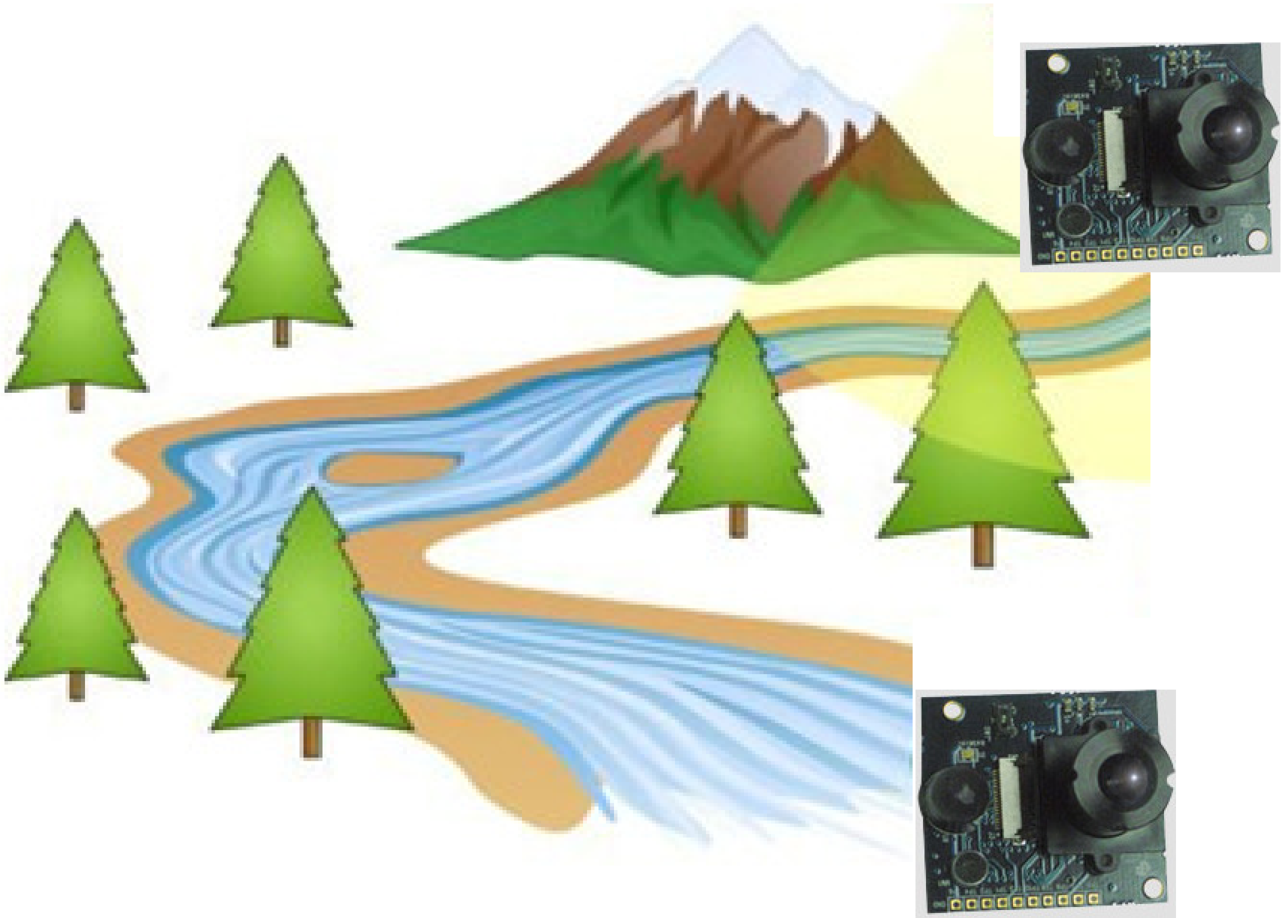
- The **blur measure** can be computed **directly on each sensor node**

- The blur measures are **scalar** that can be sent on the network

- The **test** can be reasonably **executed on cluster head**

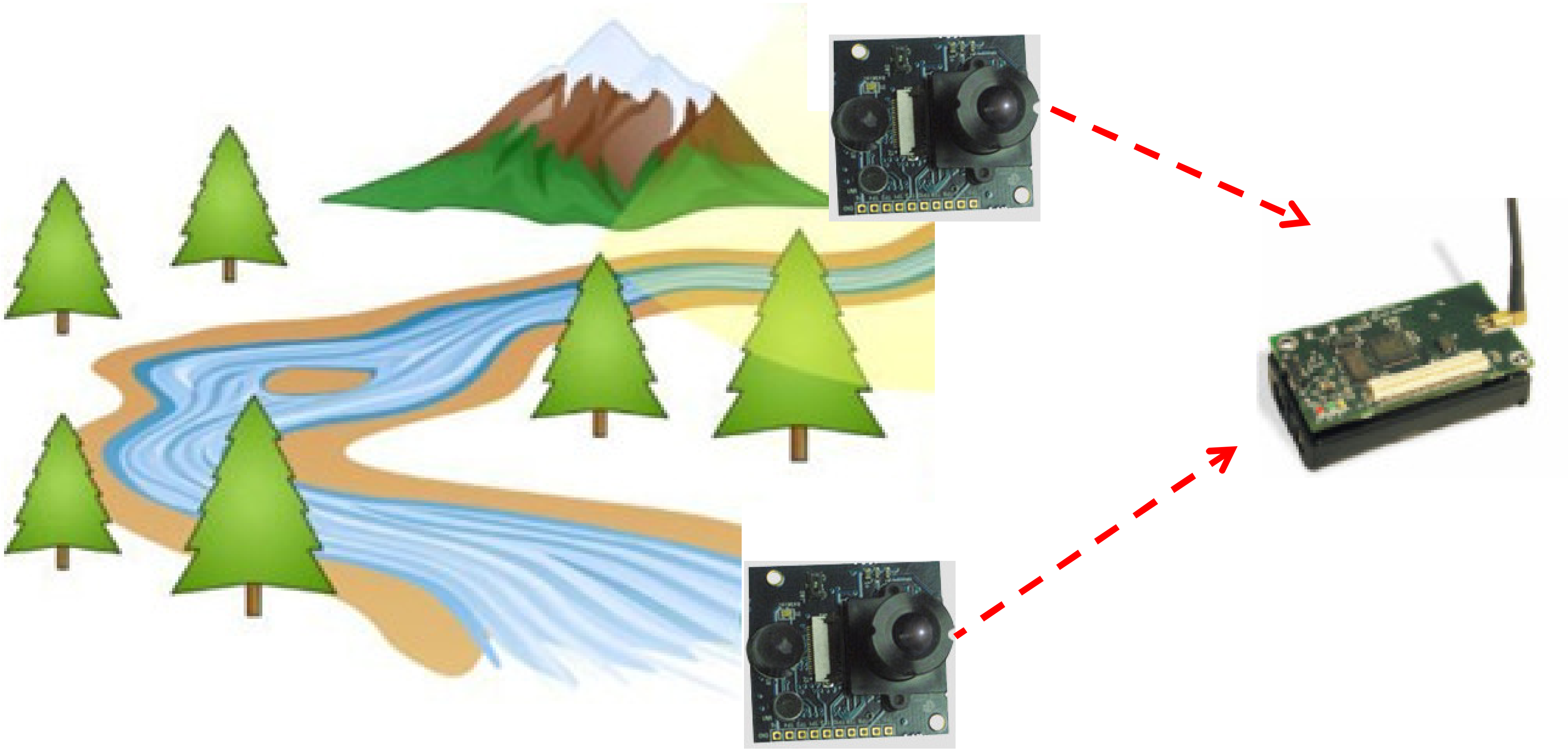
Our Approach

- Each node periodically compute blur measures



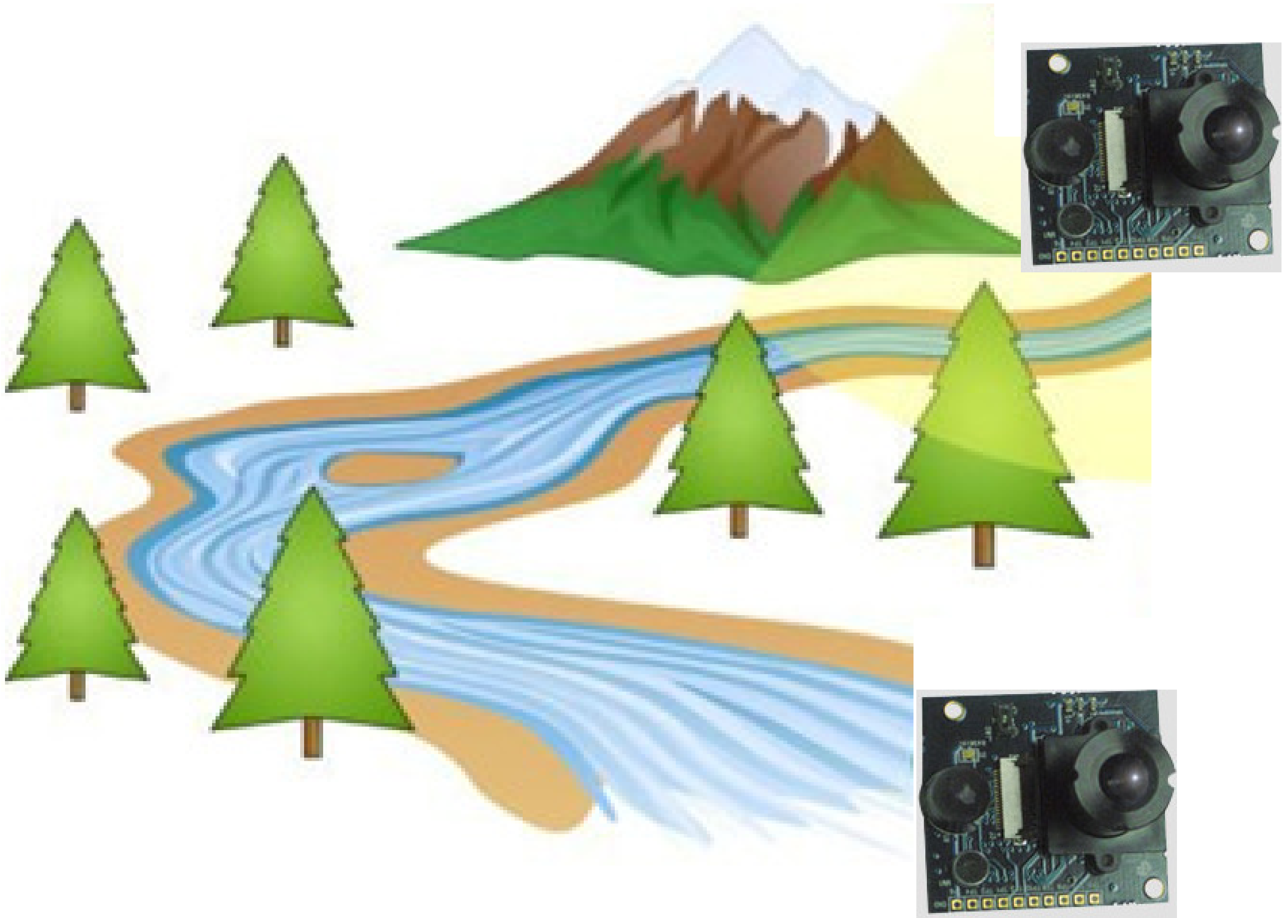
Our Approach

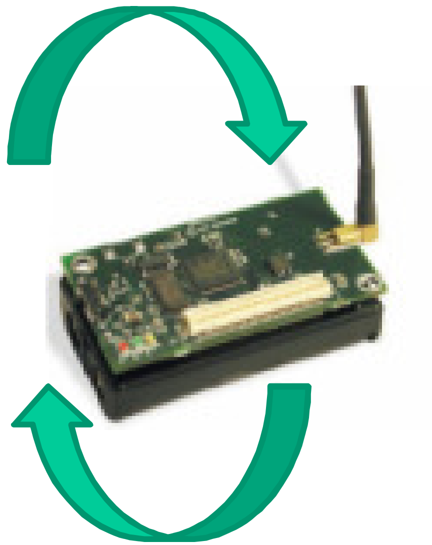
- Each node periodically compute blur measures and sends them to the remote station



Our Approach

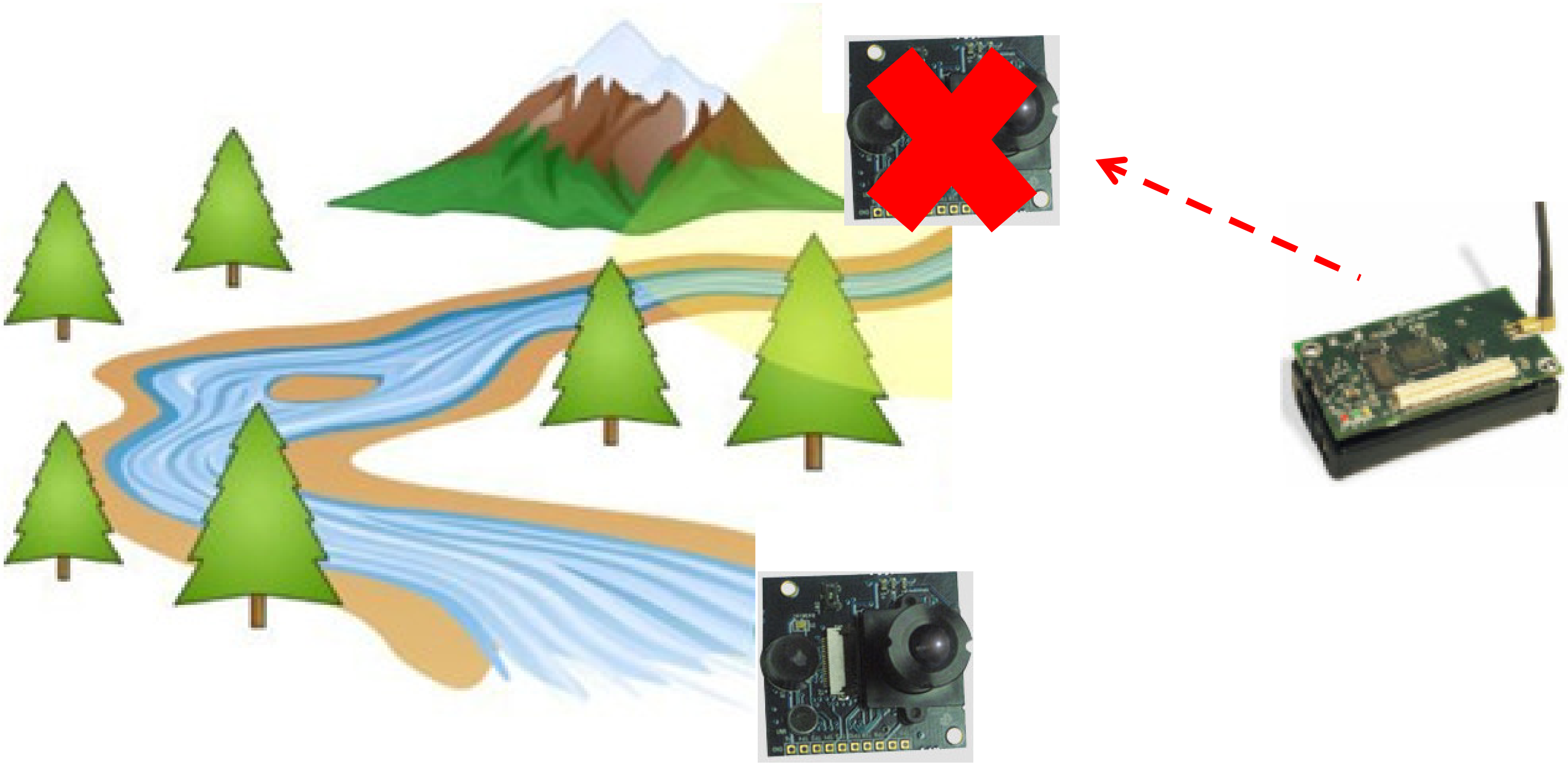
- The remote station run the test and determine if one node is acquiring corrupted data





Our Approach

- and then the netwotk adopts some strategy to compensate the node



The Observation Model

- For the sake of simplicity the observed image z is modeled as the result of a **degradation process** $\mathcal{D}$ that acts on the original (and unknown) image y :

$$z(x) = \mathcal{D}(y)(x) = \mathcal{B}(y)(x) + \eta(x), \;\; x \in \mathcal{X}$$

$\mathcal{B}(y)(x) = \int_{\mathcal{X}} y(x)h(x,s)ds$ is the blur operator

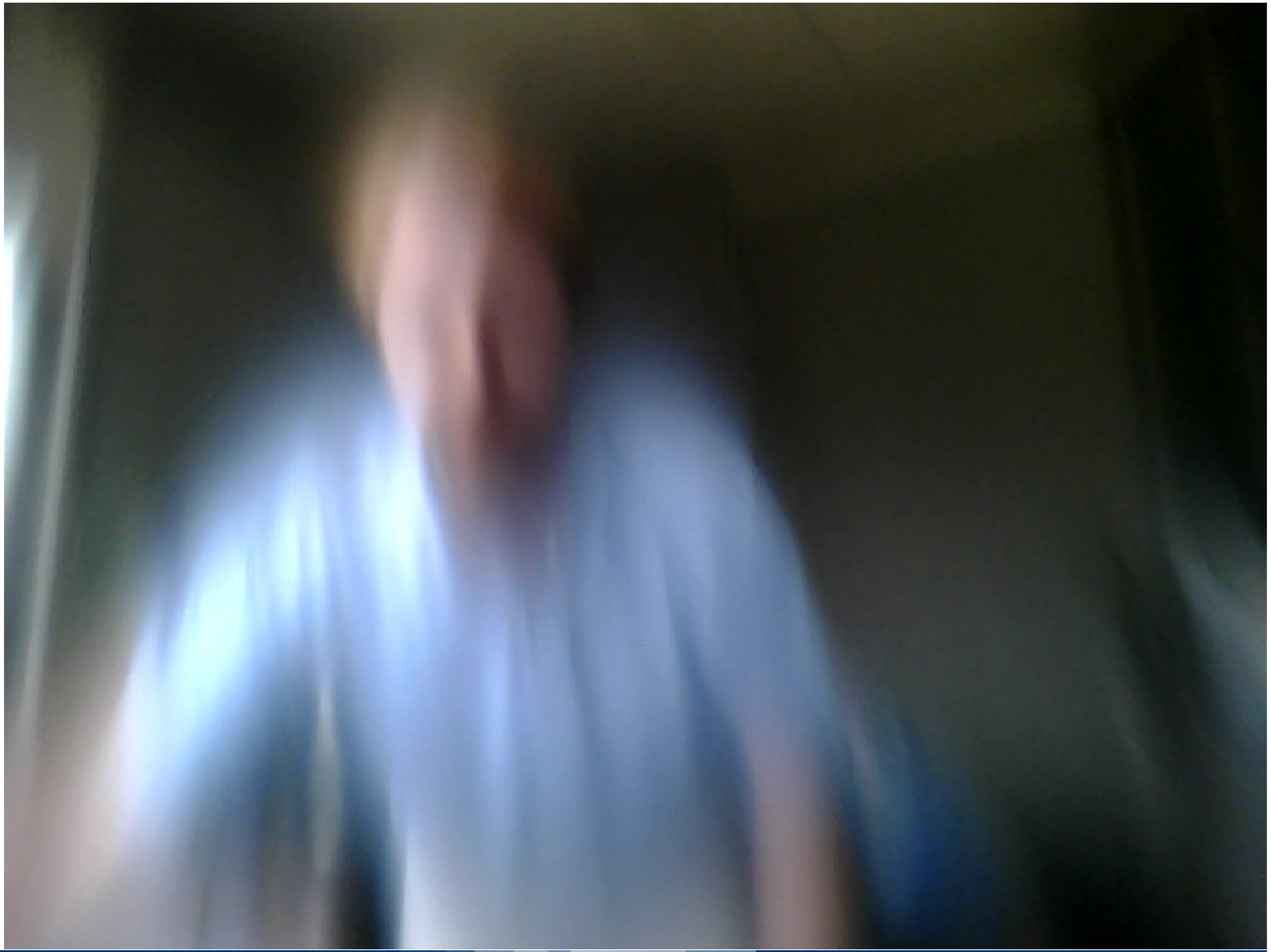
$h(x,\cdot)$ is the Point Spread Function at pixel x

η is the noise term

$\mathcal{X}$ is the image domain

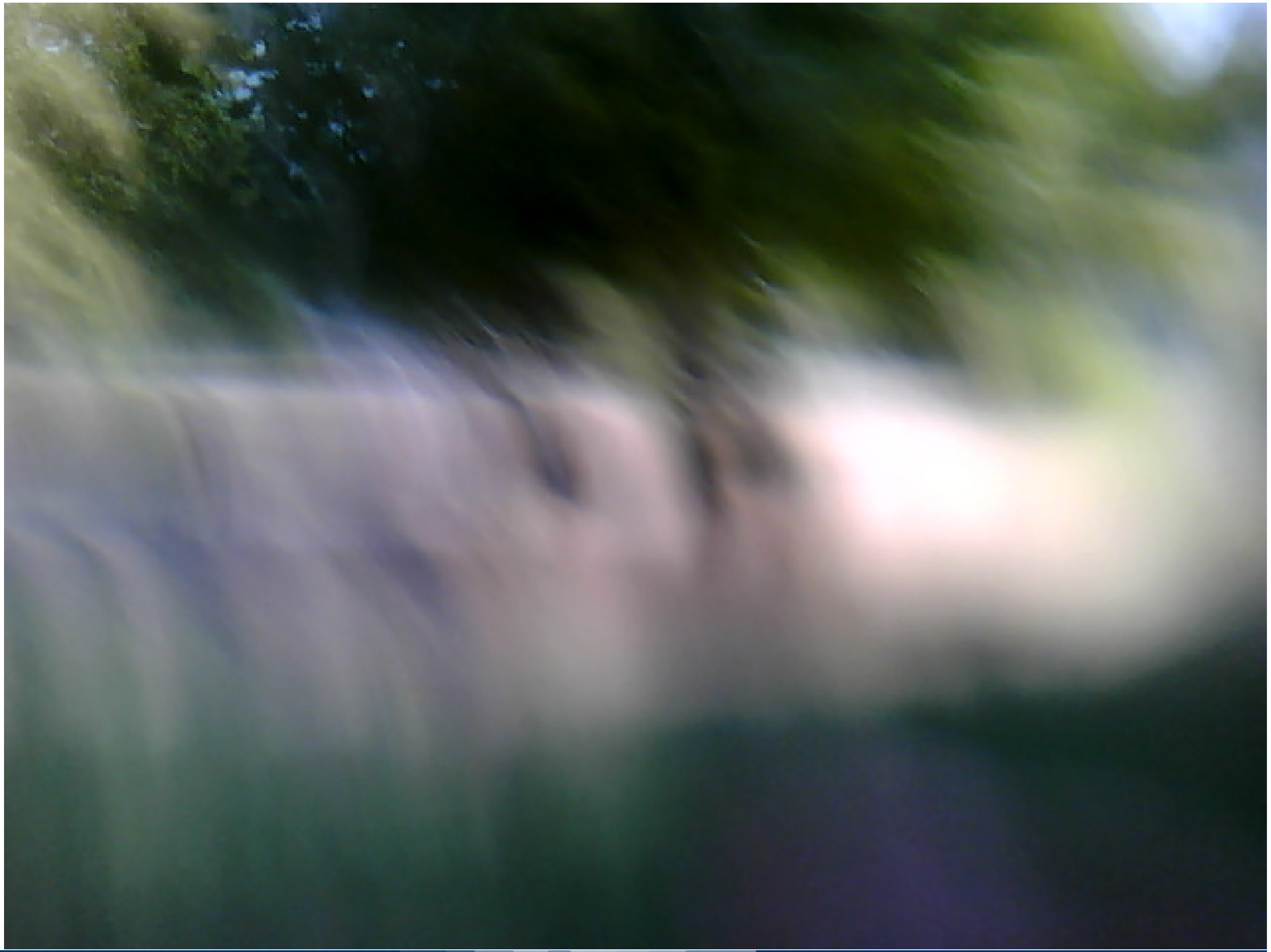
The Observation Model

- Space invariant blur $h(x, s) = g(x - s) \quad x \in \mathcal{X}$



The Observation Model

- Space variant blur $h(x,s) = g(x,s) \quad x \in \mathcal{X}$



The Observation Model

- Space invariant blur $$h(x,s) = \begin{cases} \delta(x-s), & x \in \mathcal{X}_0 \\ g(x,s), & x \in \mathcal{X}_1 \end{cases}, \mathcal{X}_0 \cup \mathcal{X}_1 = \mathcal{X}$$

The Observation Model

- We assume that we have a sequence of images

$$z_i(x) = \mathcal{B}_i\left(y_i\right)(x) + \eta(x),\ \ i = 1, \ldots, N$$

and **possibly** the original images y_{i-1} and y_i are different, as they have been acquired at different time instants.

- Since estimating such a blur is a very ill-pose, we simply measure the “amount of blur” in the resulting image.

- The blur operator may also change within the image sequence.

The Blur Measure

- We use a **blur-measure** taken from auto-focus algorithms

$$m_i = \int_{\mathcal{X}} ||\nabla z_i(x)||_1 dx$$

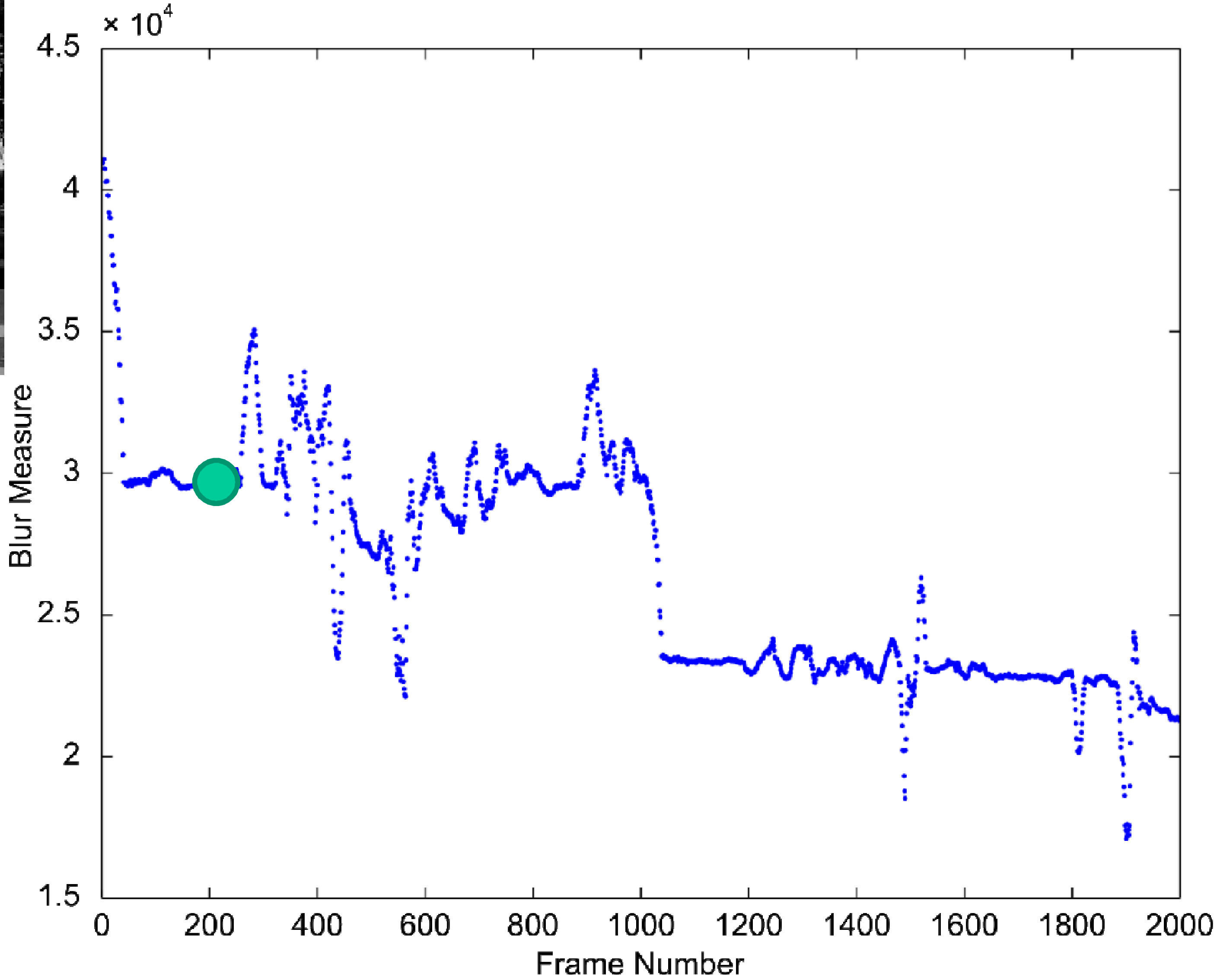
where $\| \bullet \|_1$ is the ℓ^1 norm.

The observations are assumed to have 0 mean

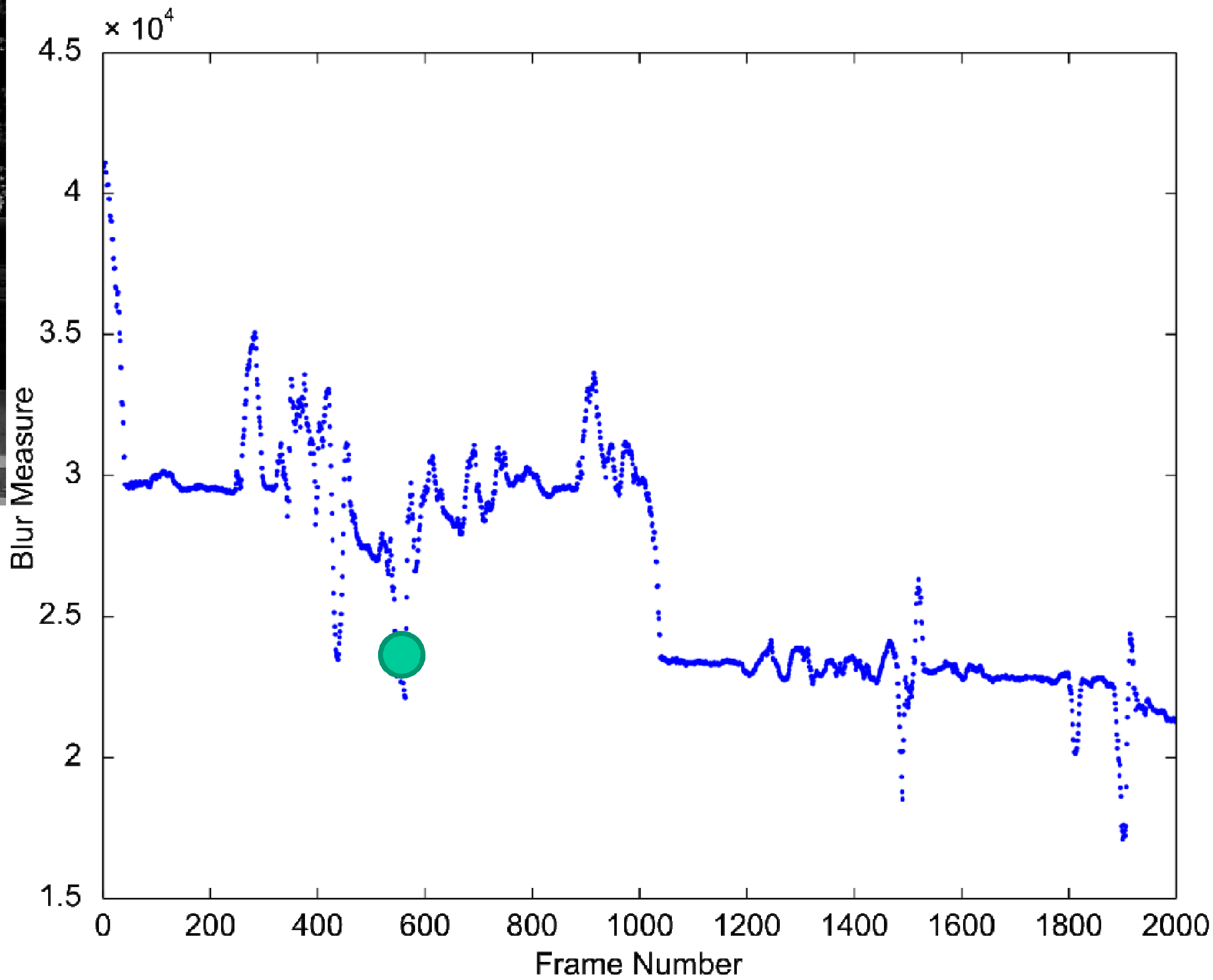
- The underlying mechanism of this measure reflects the intuitive idea that the blur suppresses the high frequency components of an image.

- The blur measure is computed on each observed image **separately: no comparison** is performed among z_i and z_{i-1}, as these may be acquired in very different time instants.

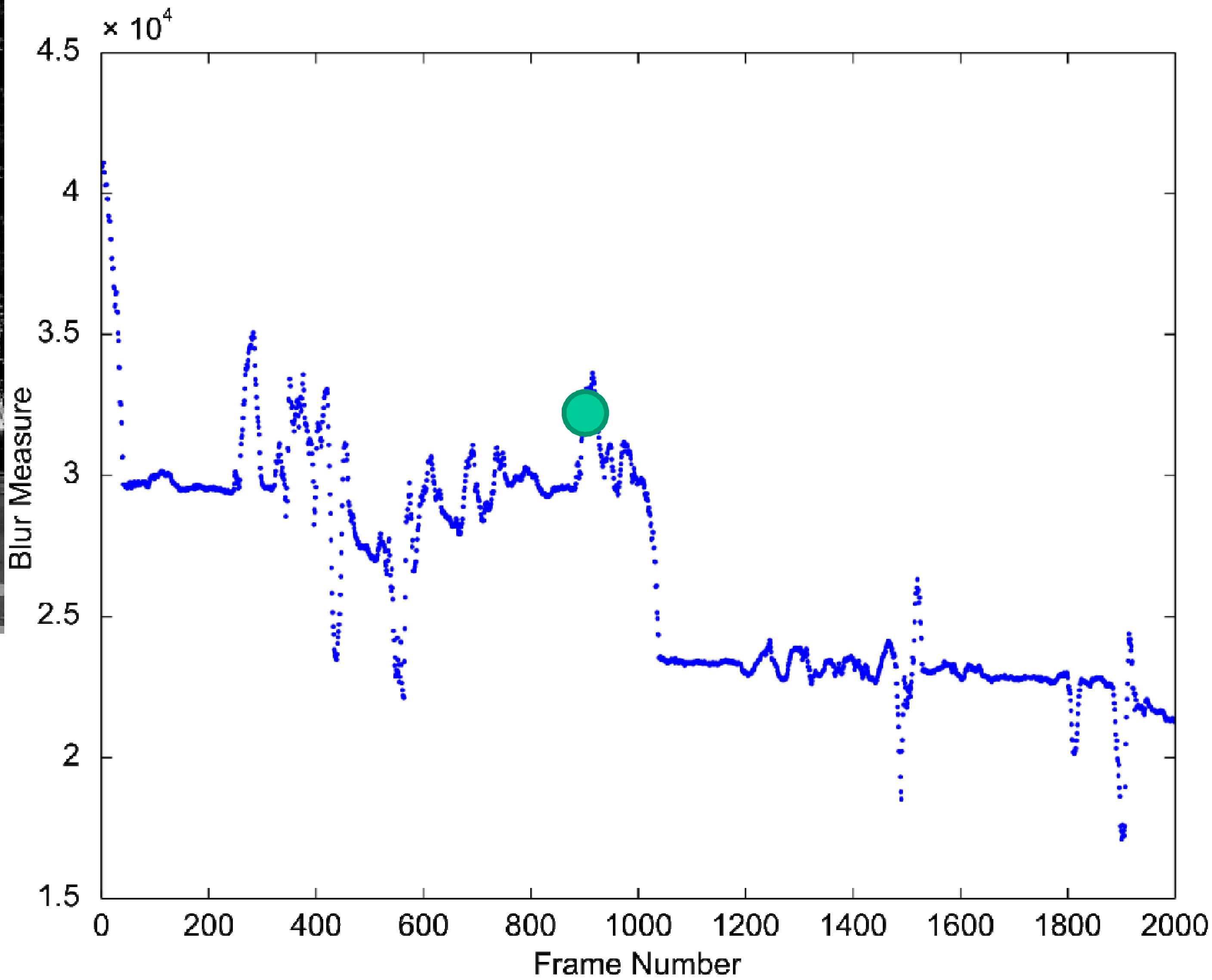
The Blur Measures



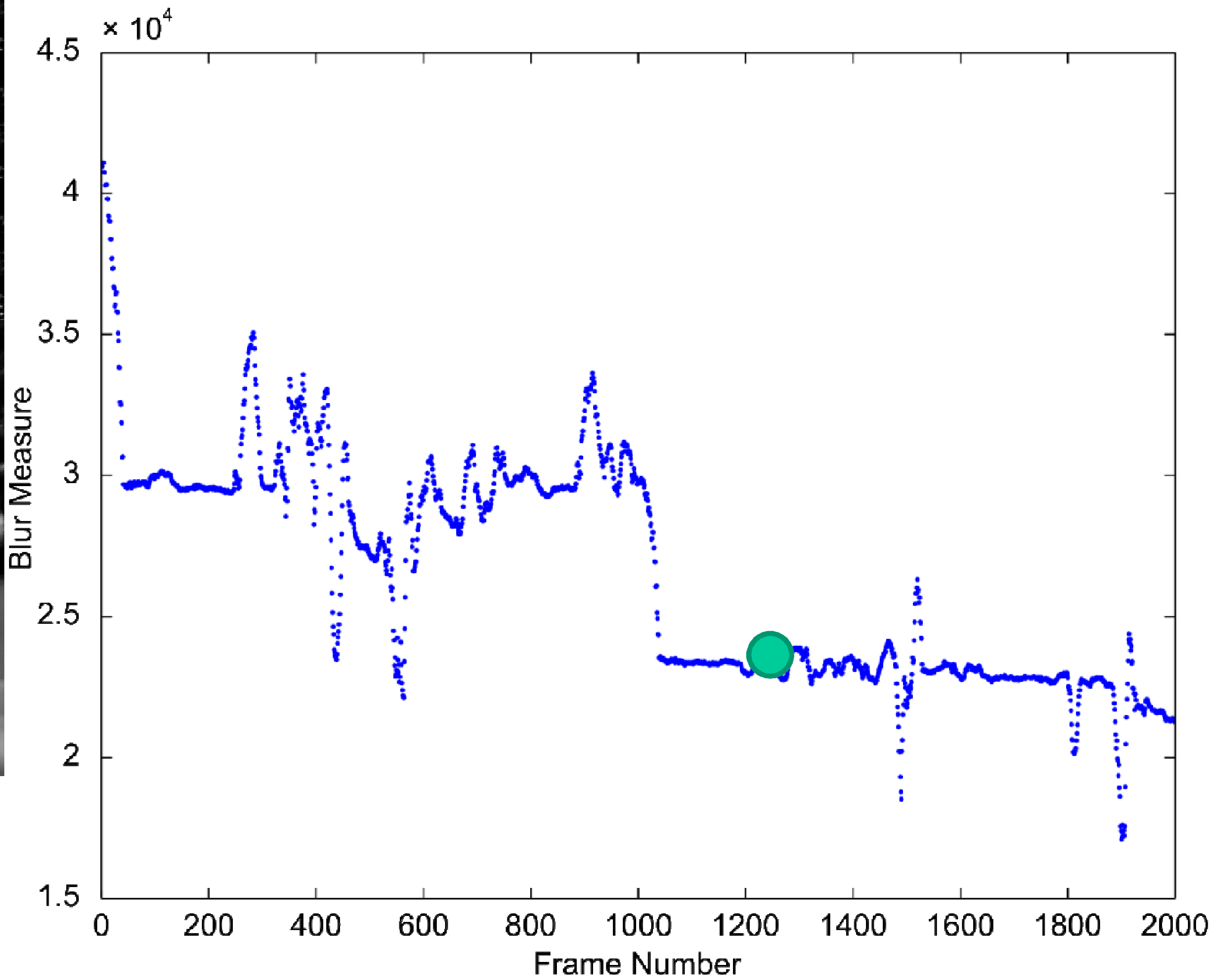
The Blur Measures



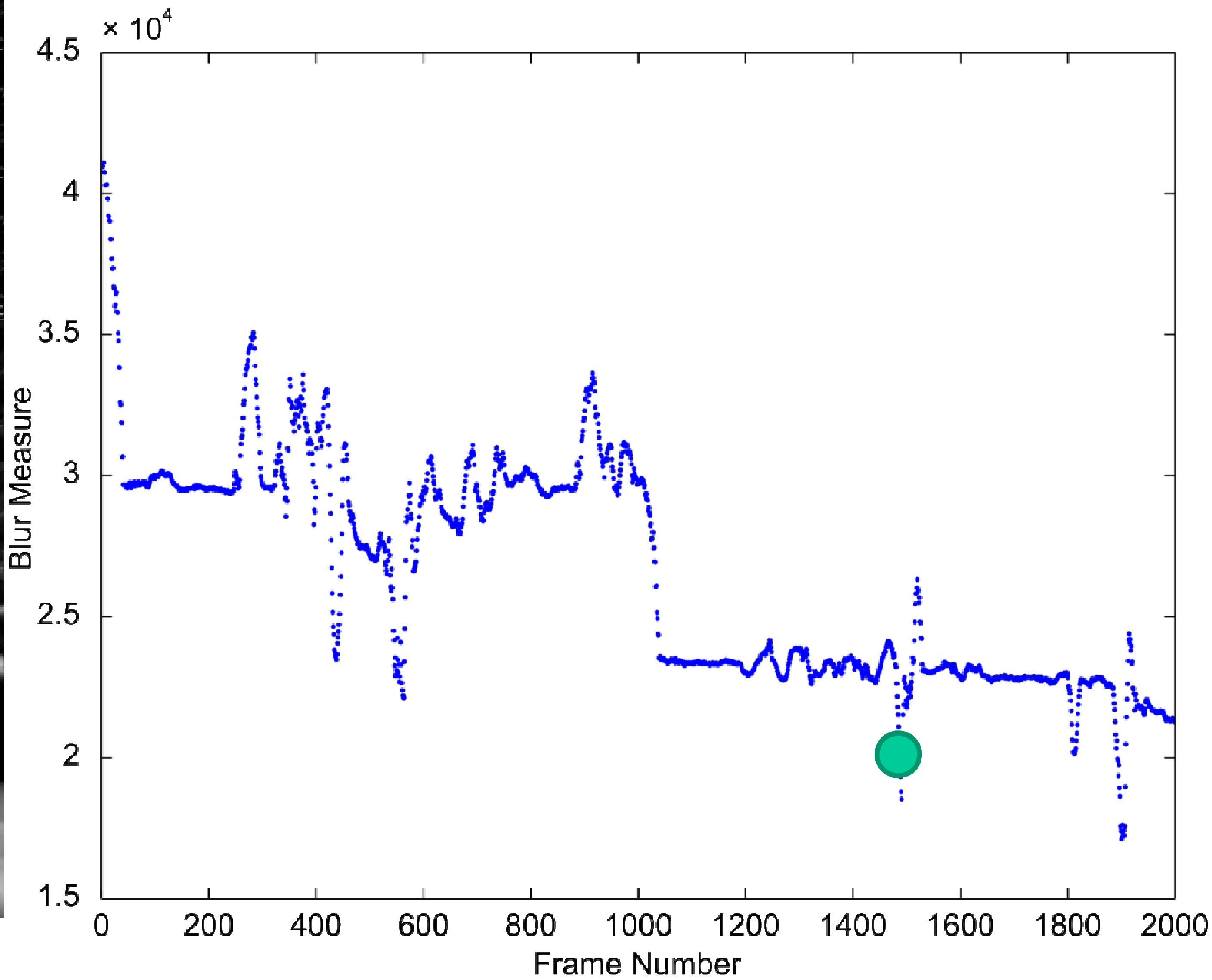
The Blur Measures



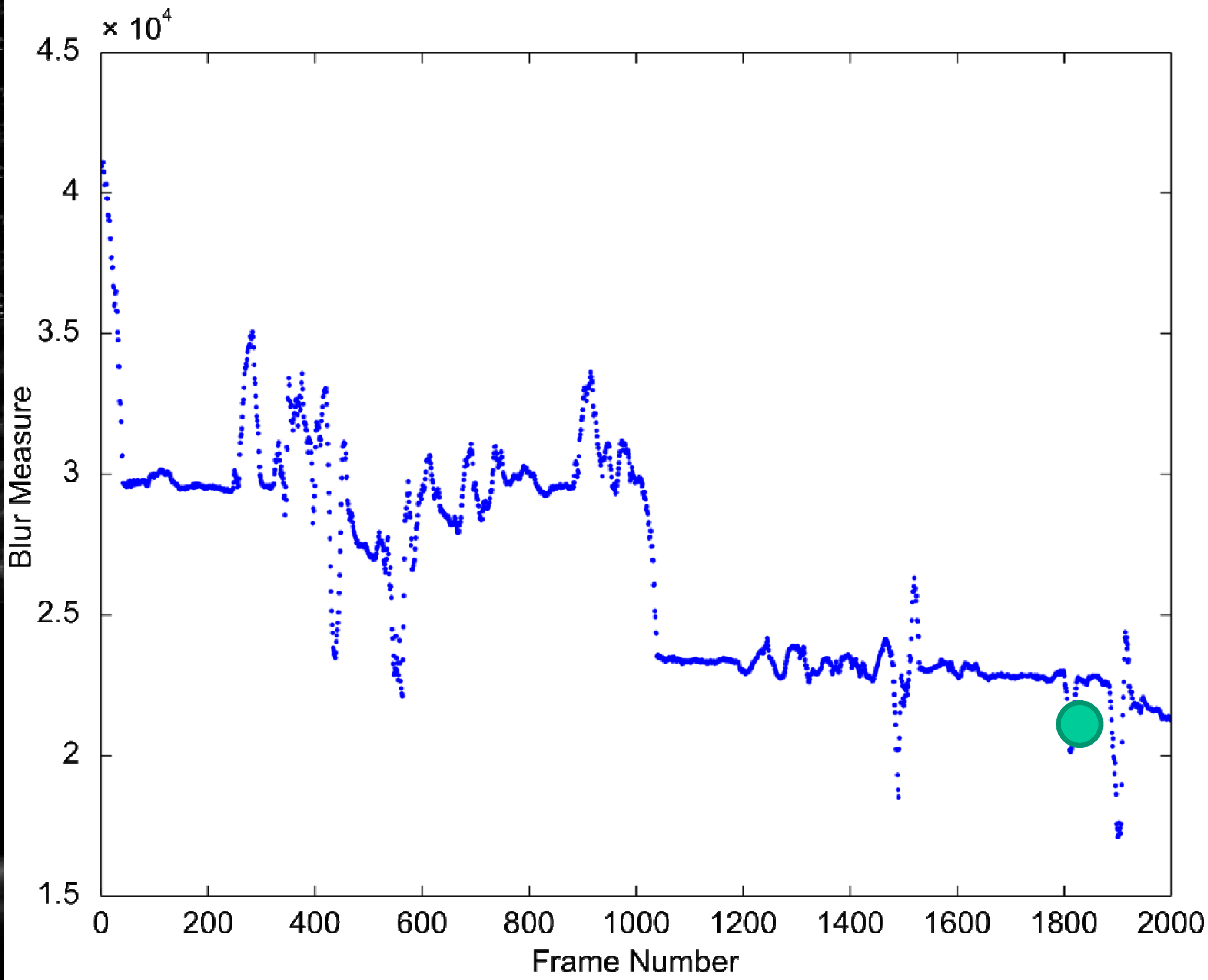
The Blur Measures



The Blur Measures



The Blur Measures



Change Detection Test

- A statistical technique to monitor the state of a process over time.
- We use **CI-CUSUM test on blur measures** m_i to detect changes in the statistical behavior of the degratation process $\mathcal{D}$

Change Detection Test

- A statistical technique to monitor the state of a process over time.
- We use **CI-CUSUM test on blur measures** m_i to detect changes in the statistical behavior of the degratation process $\mathcal{D}$
  - Stationarity means the acquisition system has **no structural loss** due to blur: i.e. **no drop.**

Change Detection Test

- A statistical technique to monitor the state of a process over time.
- We use **CI-CUSUM test on blur measures** m_i to detect changes in the statistical behavior of the degratation process $\mathcal{D}$
  - Stationarity means the acquisition system has **no structural loss** due to blur: i.e. **no drop.**
  - The arrival of **a drop on camera lens** changes the statistical behavior of the blur measures, and thus it is detected as a **non-stationarity** in the test

Change Detection Test

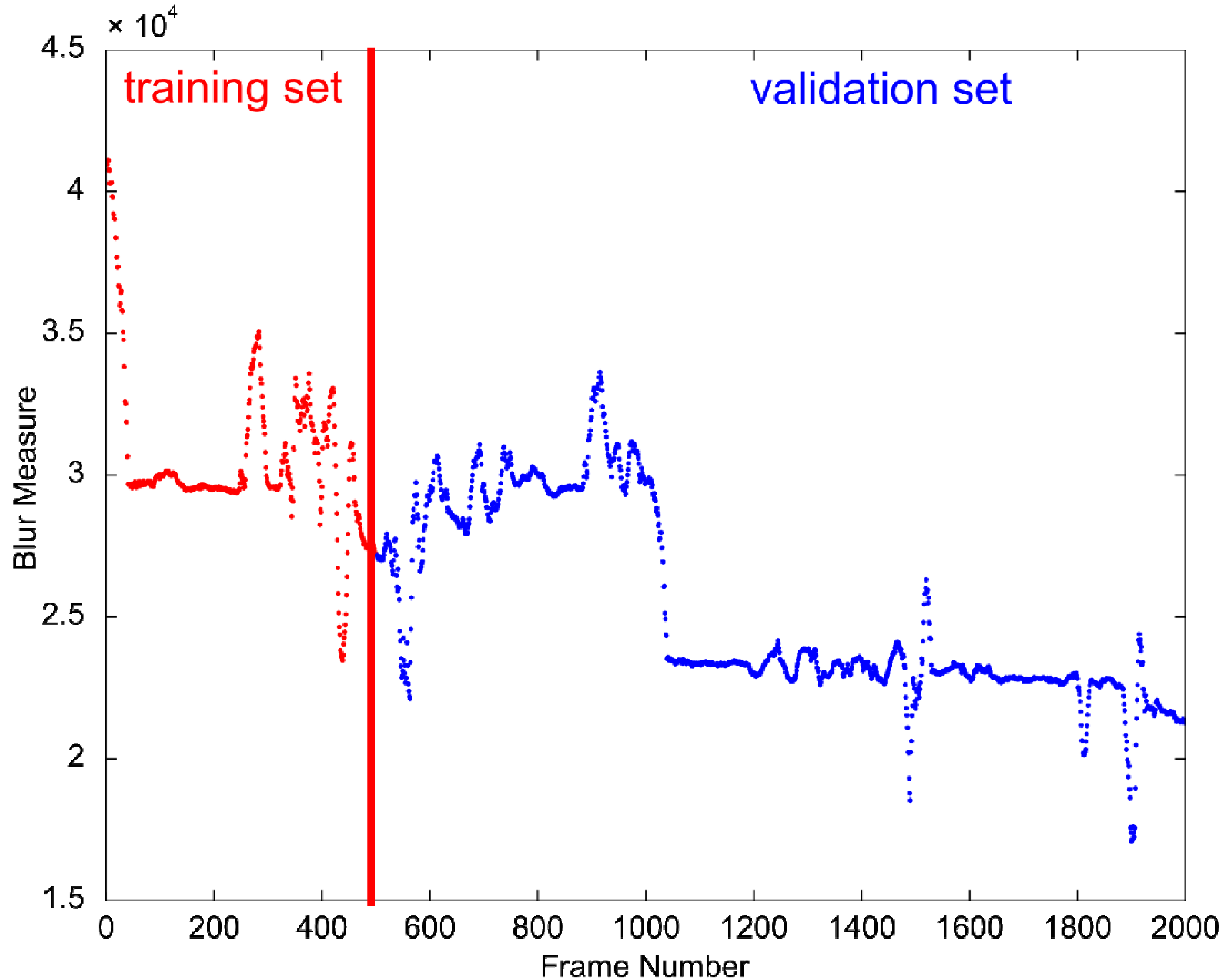
- A statistical technique to monitor the state of a process over time.

- We use **CI-CUSUM test on blur measures** m_i to detect changes in the statistical behavior of the degratation process $\mathcal{D}$
  - Stationarity means the acquisition system has **no structural loss** due to blur: i.e. **no drop.**
  - The arrival of **a drop on camera lens** changes the statistical behavior of the blur measures, and thus it is detected as a **non-stationarity** in the test

- CI-CUSUM **is general** and is automatically configured from a **training set of** m_i **computed from images in the stationary state.**

The Training Set

- The training set is composed by 500 drop-free images



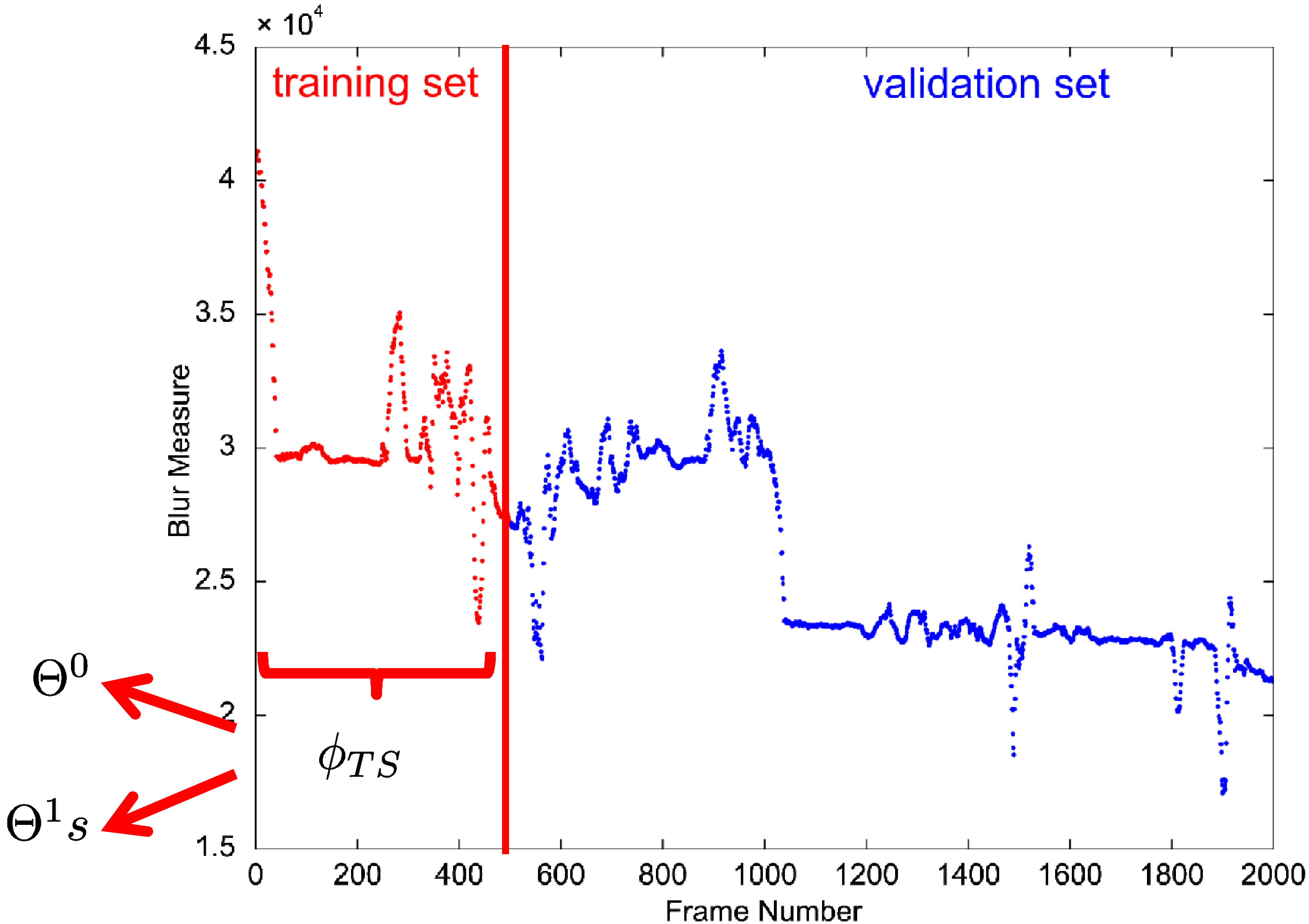
CI-CUSUM Test: training

- The CI-CUSUM test estimates some figures of merit ϕ for m_i in absence of drops, and define the null hypothesis, Θ^0 as “being in the no-drop state”.

- The alternative hypotheses Θ^1 are defined as “not being in Θ^0 ”, and thus address any type of changes w.r.t. the initial stationary state.

The Training Set

- Definition of the stationary and the alternative hypothesis



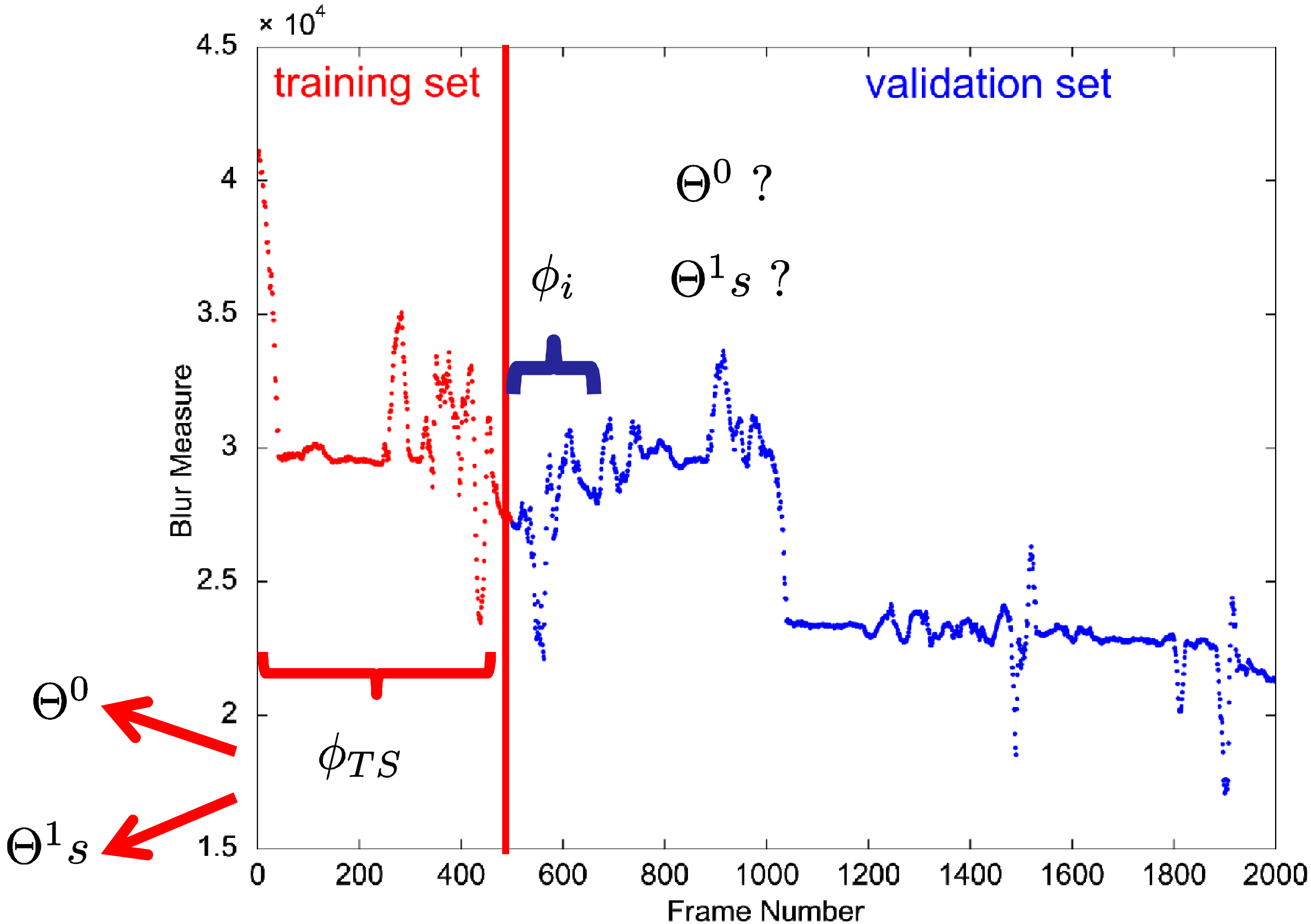
CI-CUSUM Test: change detection

- The test computes the figures of merit ϕ by gruoping observations in the validation set

- For each group, the test computes the log-likelihoods between the **figures of merit of the current state** with those of the **initial stationary state**, and compare it with an **automatically defined thresholds**

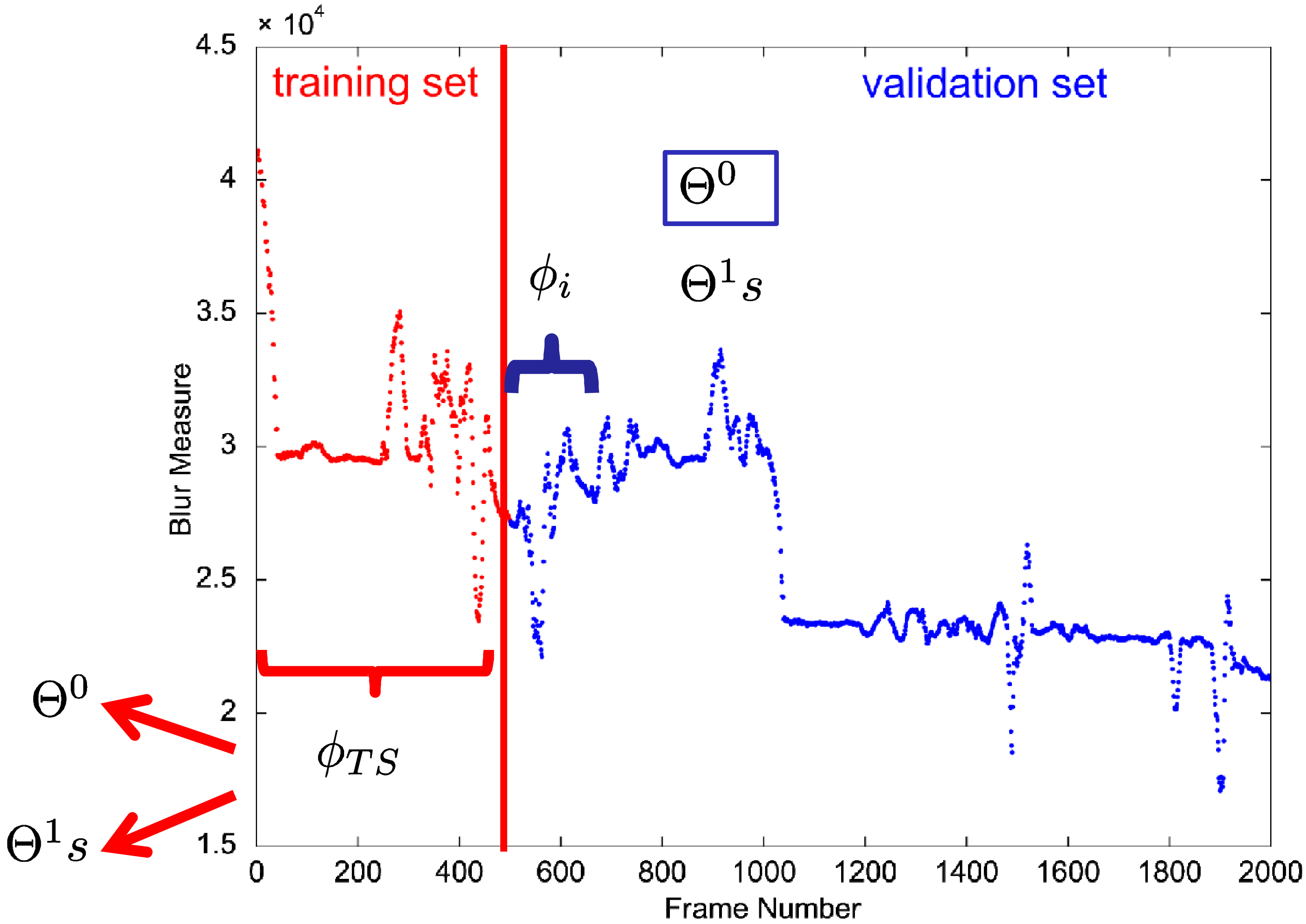
The Training Set

- Change Detection



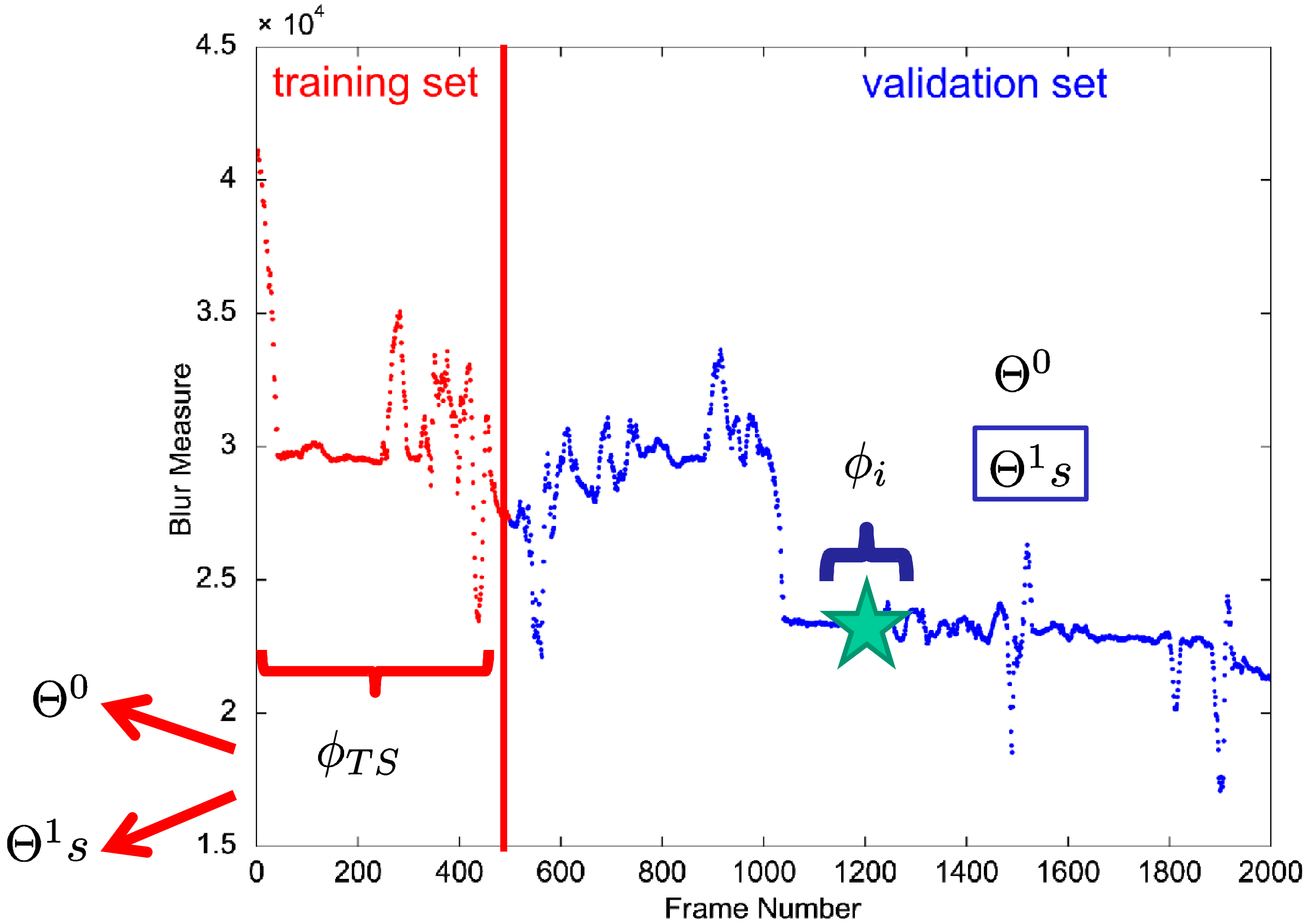
The Training Set

- Change Detection



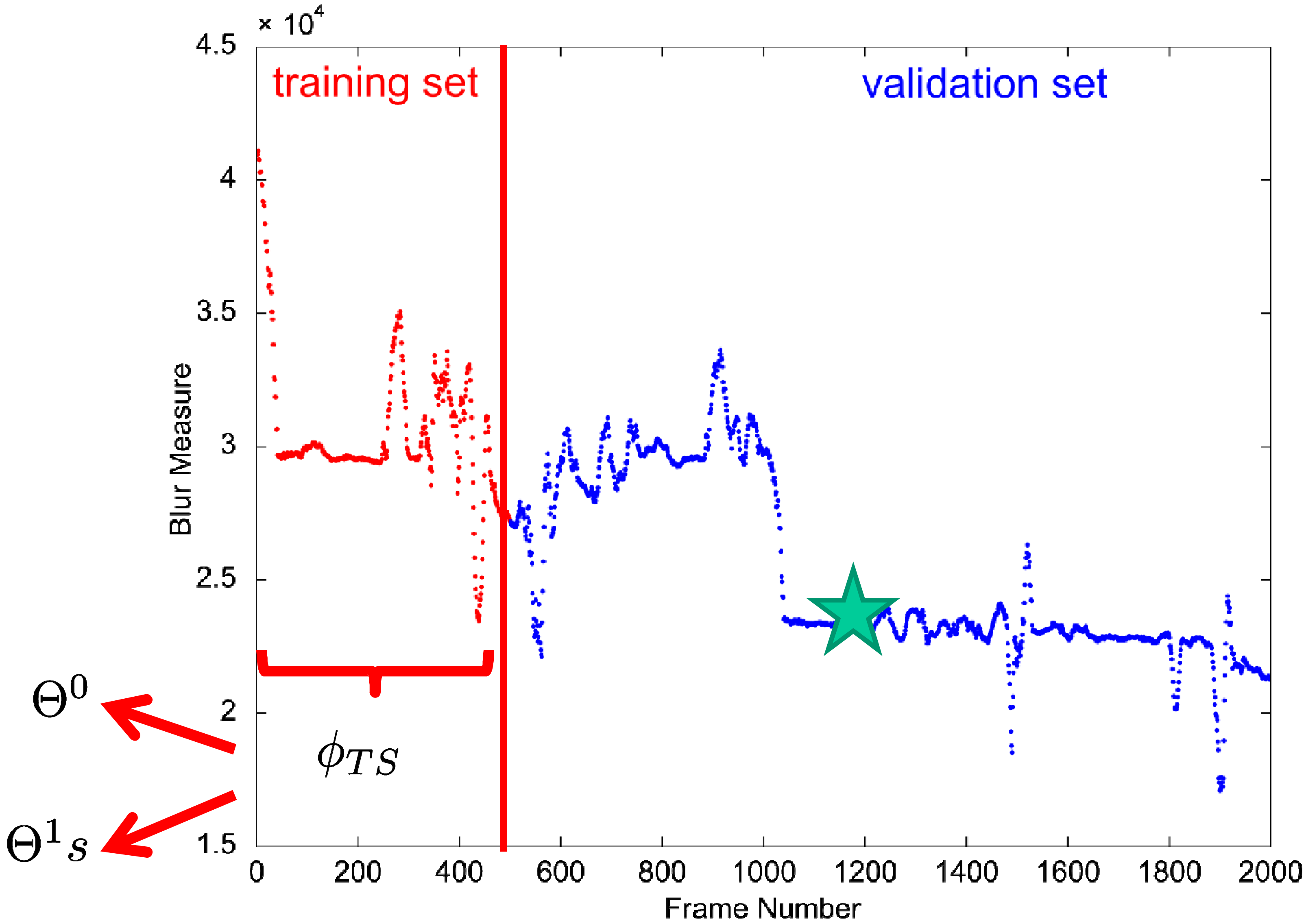
The Training Set

- Change Detection



The Training Set

- Change Detected at frame 1160

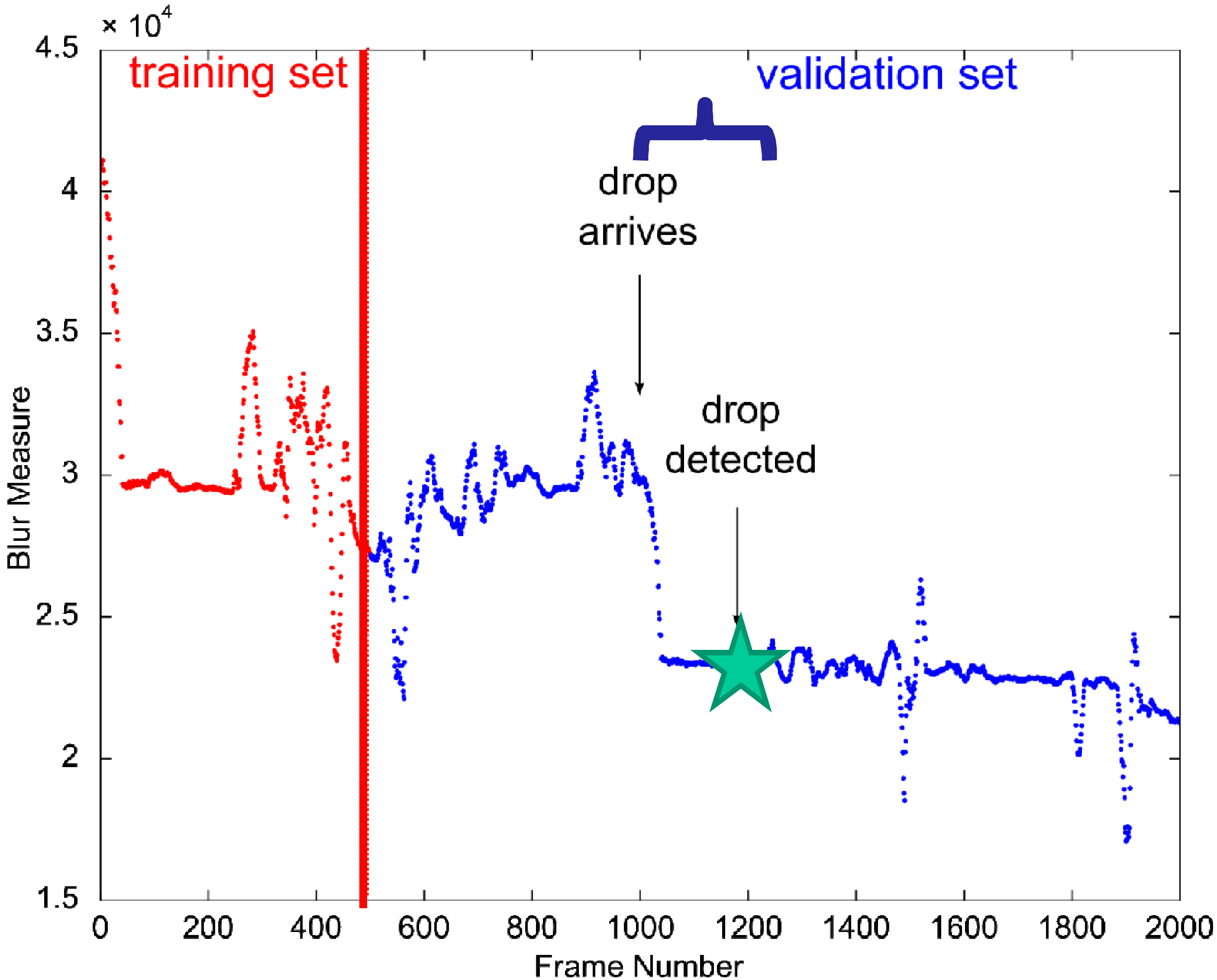


Experiments

- We adopted the following figures of merit
  - ***DL*** Detection Latency: the number of images acquired before identifying a change in the blurring process.
  - ***FP*** False Positive, the number detected changes not supported by a real change in the blurring process.
  - ***FN*** False Negative, the number missed changes in the blurring process

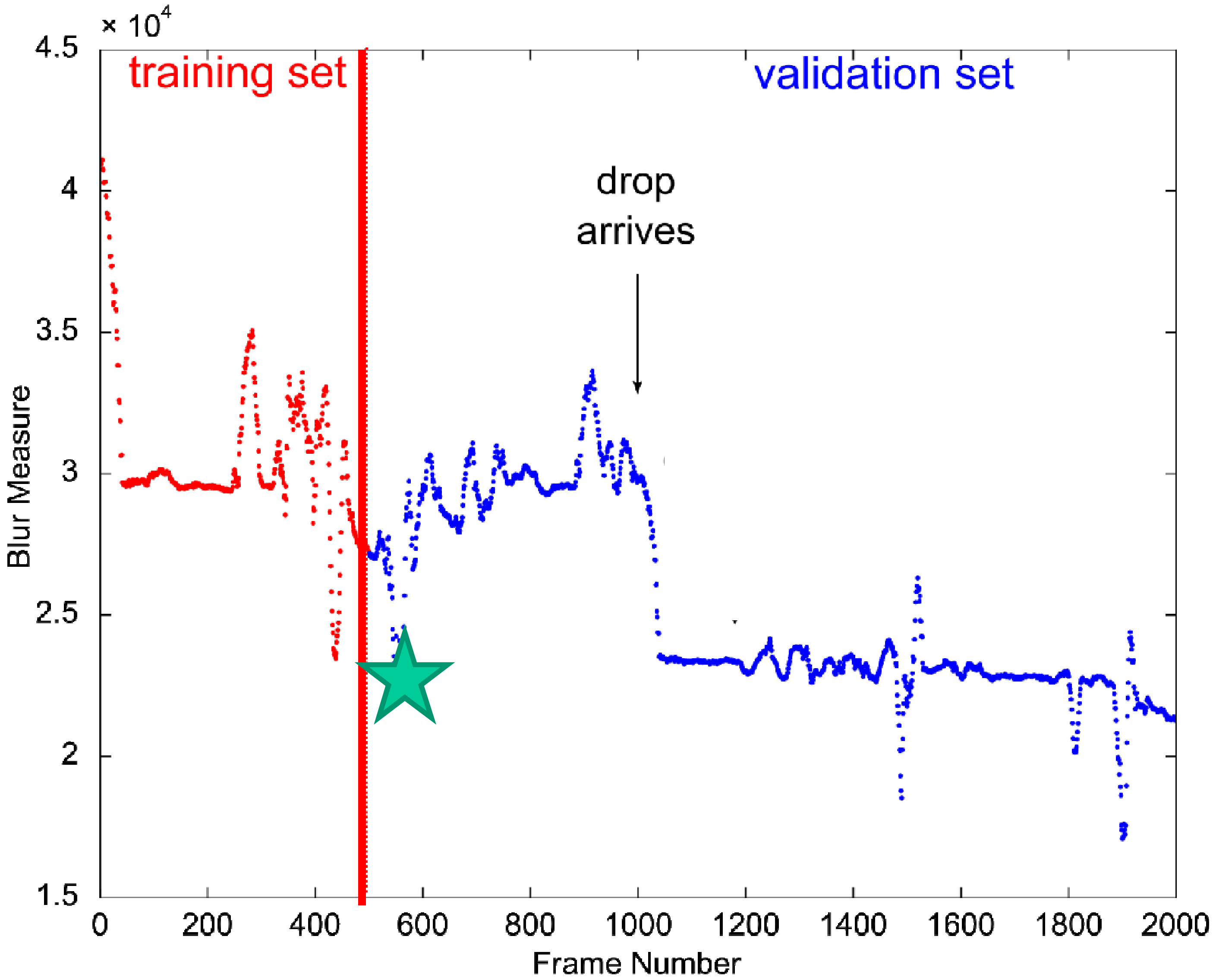
Experiments

- Detection Latency



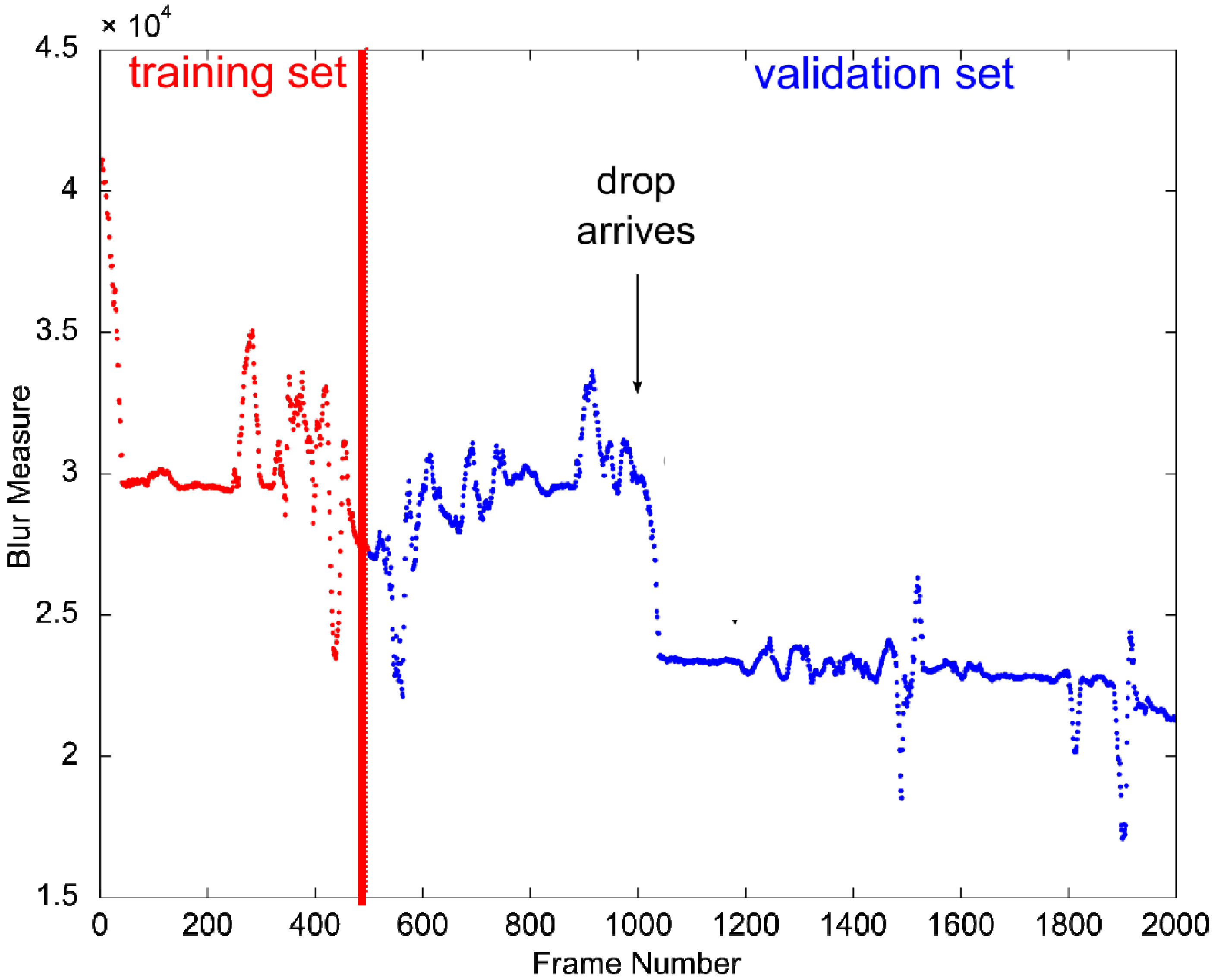
Experiments

- False Positive



Experiments

- False Negative



Experiments on Sythetically Blurred Images

- We generated sequences from 75 grayscale images in a random order

Experiments on Sythetically Blurred Images

- We generated sequences from 75 grayscale images in a random order

Experiments on Sythetically Blurred Images

- We generated sequences from 75 grayscale images in a random order
- For simplicity, the blur has been generated with a 2D convolution with a Gaussian kernel h having standard deviation ν

$$\mathcal{B}(y) = (y \circledast h)$$

- Noise has been generated from a Normal distribution

$$\eta \sim N(0, \sigma^2)$$

- We considered different amount of blur and noise

$$\nu = 1, \cdots, 8 \qquad\qquad \sigma = 0.02, 0.04, 0.06, 0.08$$

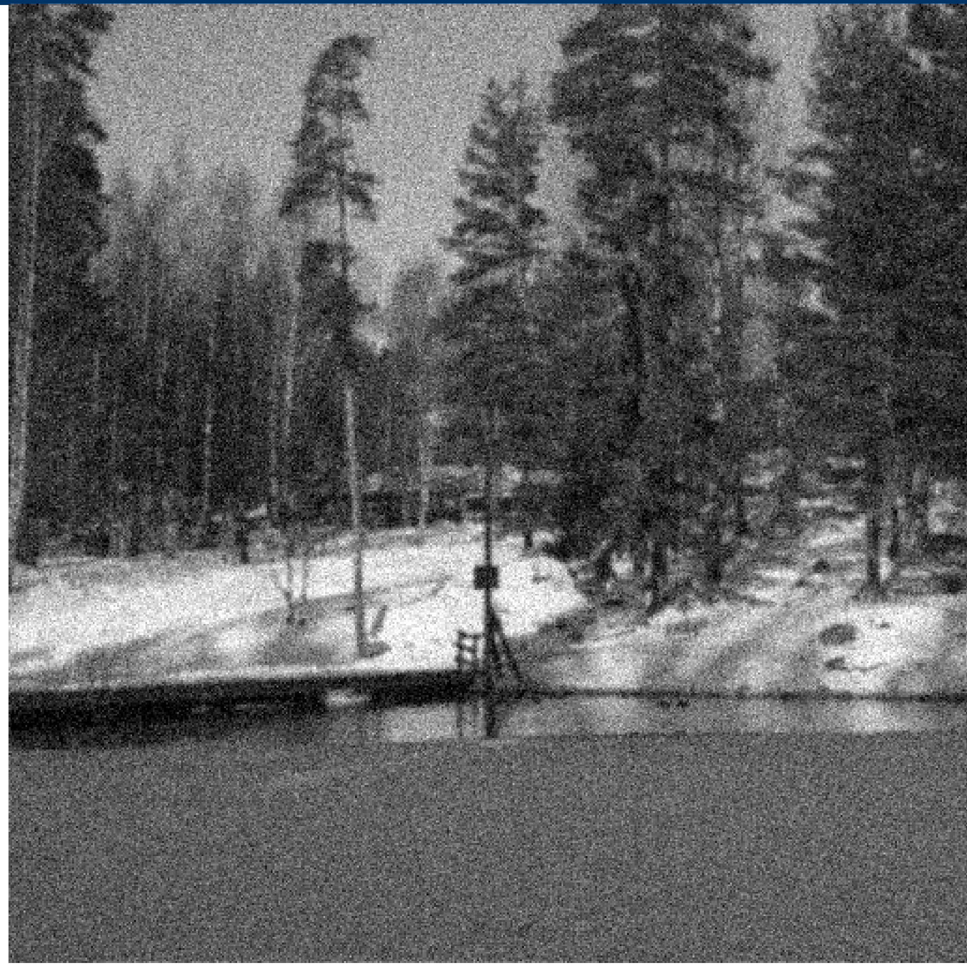
- Each sequence contains 1000 blur-free images (500 are used for training the CI-CUSUM) and 1000 blurred images.
- Results have been averaged among 100 sequences for each parameter pair

Experiments on Sythetically Blurred Images

- Two blur operators:
  - the blur affects the **whole** image

$$\sigma = 0.08$$

$$\nu = 1$$



Experiments on Sythetically Blurred Images

- Two blur operators:
  - the blur affects the **whole** image

$$\sigma = 0.08$$

$$\nu = 4$$

Experiments on Sythetically Blurred Images

- Two blur operators:
  - the blur affects the **whole** image

$$\sigma = 0.08$$

$$\nu = 8$$

Experiments on Sythetically Blurred Images

- Two blur operators:
  - the blur affects the **whole** image
  - the blur affects **part** of the image

$$\sigma = 0.02$$

$$\nu = 1$$

Experiments on Sythetically Blurred Images

- Two blur operators:
  - the blur affects the **whole** image
  - the blur affects **part** of the image

$$\sigma = 0.02$$

$$\nu = 4$$

Experiments on Sythetically Blurred Images

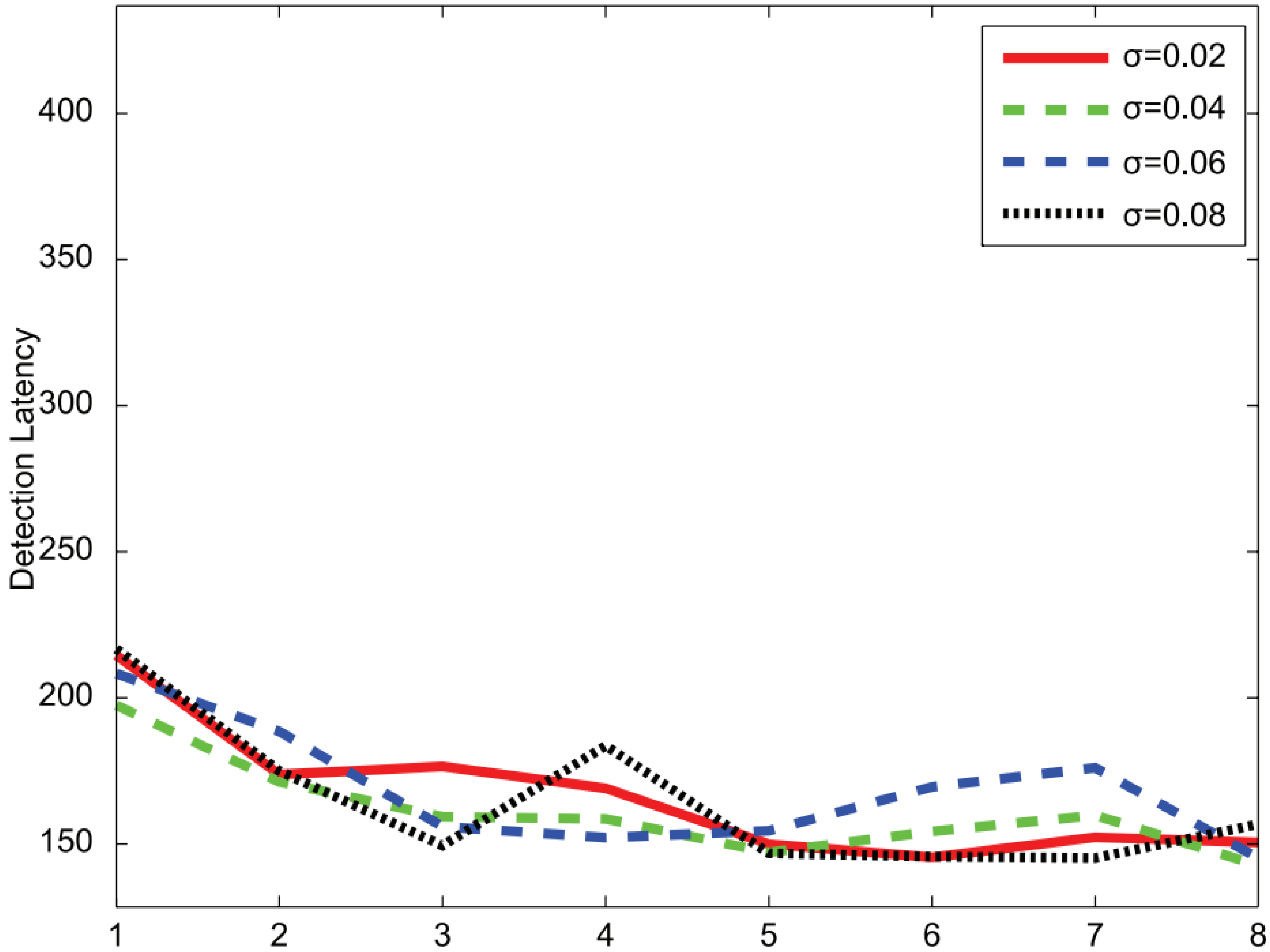
- Two blur operators:
  - the blur affects the **whole** image
  - the blur affects **part** of the image

$$\sigma = 0.02$$

$$\nu = 8$$

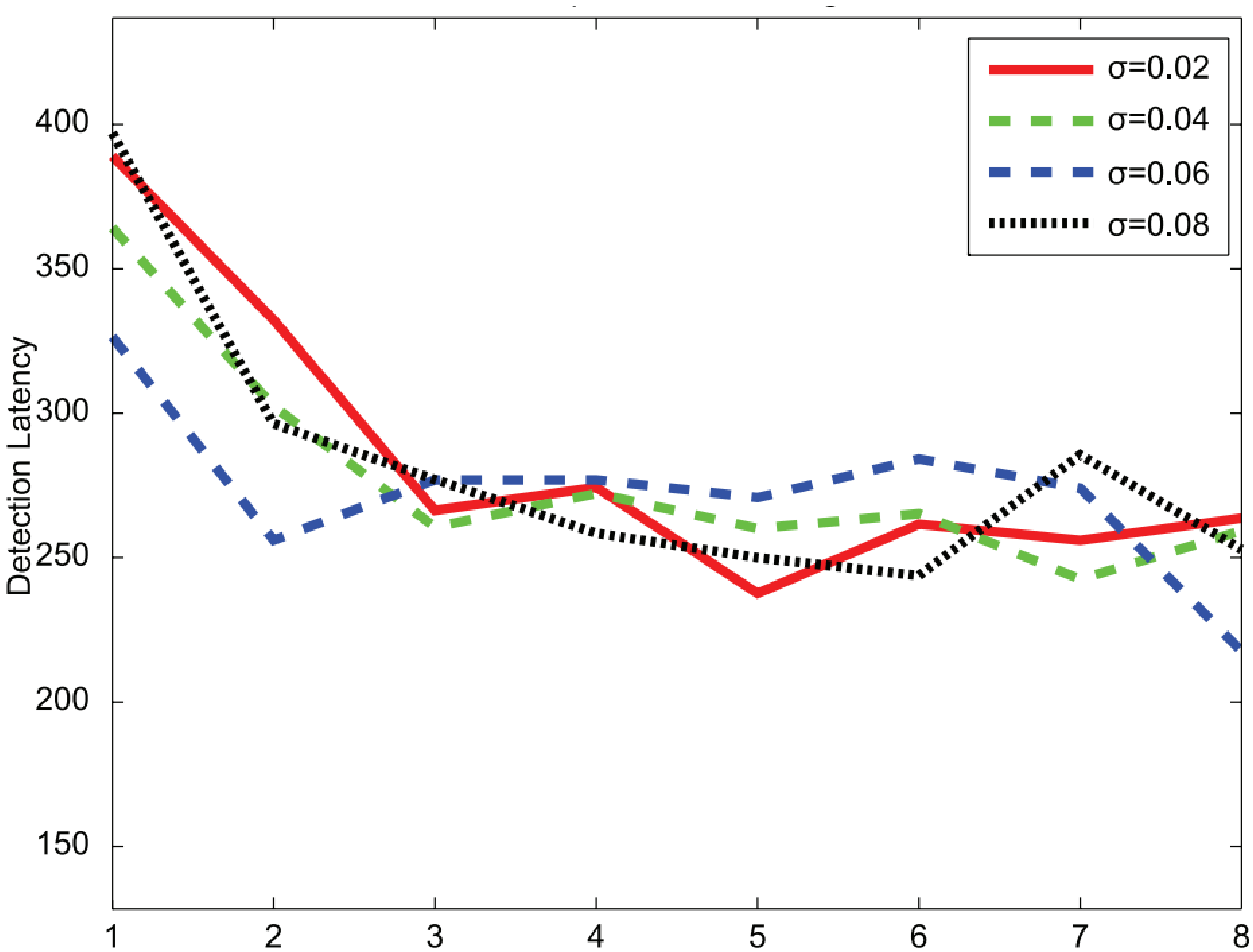
Experiments on Sythetically Blurred Images

- Detection Latency: Blur on the whole image



Experiments on Sythetically Blurred Images

- Detection Latency: Blur on part of images



Experiments on Sythetically Blurred Images

- False Positive and False Negative results

Blur	σ	Detection	ν = 1	2	3	4	5	6	7	8
FULL	0.02	$FP(\%)$	10	18	18	7	10	10	14	7
		$FN(\%)$	1	0	1	0	4	2	0	0
FULL	0.04	$FP(\%)$	14	13	9	9	12	16	11	13
		$FN(\%)$	6	0	0	1	0	3	0	1
FULL	0.06	$FP(\%)$	8	15	9	9	9	6	9	17
		$FN(\%)$	2	2	3	1	3	2	0	0
FULL	0.08	$FP(\%)$	9	11	4	12	4	13	10	8
		$FN(\%)$	6	0	1	1	1	0	1	0
PART	0.02	$FP(\%)$	11	8	6	11	8	15	15	13
		$FN(\%)$	34	17	13	11	11	3	6	5
PART	0.04	$FP(\%)$	12	11	10	7	4	11	10	9
		$FN(\%)$	37	16	10	12	7	7	6	5
PART	0.06	$FP(\%)$	12	11	19	12	10	8	8	14
		$FN(\%)$	38	19	4	12	7	8	8	5
PART	0.08	$FP(\%)$	5	12	8	12	13	8	13	16
		$FN(\%)$	36	20	8	11	11	7	7	4

Experiments on Sythetically Blurred Images

- False Positive are independend of the amount of blur

Blur	σ	Detection	ν 1	2	3	4	5	6	7	8
FULL	0.02	$FP(\%)$	10	18	18	7	10	10	14	7
		$FN(\%)$	1	0	1	0	4	2	0	0
FULL	0.04	$FP(\%)$	14	13	9	9	12	16	11	13
		$FN(\%)$	6	0	0	1	0	3	0	1
FULL	0.06	$FP(\%)$	8	15	9	9	9	6	9	17
		$FN(\%)$	2	2	3	1	3	2	0	0
FULL	0.08	$FP(\%)$	9	11	4	12	4	13	10	8
		$FN(\%)$	6	0	1	1	1	0	1	0
PART	0.02	$FP(\%)$	11	8	6	11	8	15	15	13
		$FN(\%)$	34	17	13	11	11	3	6	5
PART	0.04	$FP(\%)$	12	11	10	7	4	11	10	9
		$FN(\%)$	37	16	10	12	7	7	6	5
PART	0.06	$FP(\%)$	12	11	19	12	10	8	8	14
		$FN(\%)$	38	19	4	12	7	8	8	5
PART	0.08	$FP(\%)$	5	12	8	12	13	8	13	16
		$FN(\%)$	36	20	8	11	11	7	7	4

Experiments on Sythetically Blurred Images

- False Positive are independend of the amount of blur

Blur	σ	Detection	ν 1	2	3	4	5	6	7	8
FULL	0.02	$FP(\%)$	10	18	18	7	10	10	14	7
		$FN(\%)$	1	0	1	0	4	2	0	0
FULL	0.04	$FP(\%)$	14	13	9	9	12	16	11	13
		$FN(\%)$	6	0	0	1	0	3	0	1
FULL	0.06	$FP(\%)$	8	15	9	9	9	6	9	17
		$FN(\%)$	2	2	3	1	3	2	0	0
FULL	0.08	$FP(\%)$	9	11	4	12	4	13	10	8
		$FN(\%)$	6	0	1	1	1	0	1	0
PART	0.02	$FP(\%)$	11	8	6	11	8	15	15	13
		$FN(\%)$	34	17	13	11	11	3	6	5
PART	0.04	$FP(\%)$	12	11	10	7	4	11	10	9
		$FN(\%)$	37	16	10	12	7	7	6	5
PART	0.06	$FP(\%)$	12	11	19	12	10	8	8	14
		$FN(\%)$	38	19	4	12	7	8	8	5
PART	0.08	$FP(\%)$	5	12	8	12	13	8	13	16
		$FN(\%)$	36	20	8	11	11	7	7	4

Experiments on Sythetically Blurred Images

- False Negative decreases as the blur amount increass

Blur	σ	Detection	ν = 1	2	3	4	5	6	7	8
FULL	0.02	$FP(\%)$	10	18	18	7	10	10	14	7
		$FN(\%)$	1	0	1	0	4	2	0	0
FULL	0.04	$FP(\%)$	14	13	9	9	12	16	11	13
		$FN(\%)$	6	0	0	1	0	3	0	1
FULL	0.06	$FP(\%)$	8	15	9	9	9	6	9	17
		$FN(\%)$	2	2	3	1	3	2	0	0
FULL	0.08	$FP(\%)$	9	11	4	12	4	13	10	8
		$FN(\%)$	6	0	1	1	1	0	1	0
PART	0.02	$FP(\%)$	11	8	6	11	8	15	15	13
		$FN(\%)$	34	17	13	11	11	3	6	5
PART	0.04	$FP(\%)$	12	11	10	7	4	11	10	9
		$FN(\%)$	37	16	10	12	7	7	6	5
PART	0.06	$FP(\%)$	12	11	19	12	10	8	8	14
		$FN(\%)$	38	19	4	12	7	8	8	5
PART	0.08	$FP(\%)$	5	12	8	12	13	8	13	16
		$FN(\%)$	36	20	8	11	11	7	7	4

Experiments on Camera Images

- We acquired 25 sequences of QVGA uncompressed frames
  - 1000 frames drop free
  - 1000 frames with drops
  - the first 500 drop-free frames have been used as traing set

Experiments on Camera Images

Experiments on Camera Images

- We acquired 25 sequences of QVGA uncompressed frames
  - 1000 frames drop free
  - 1000 frames with drops
  - the first 500 drop-free frames have been used as traing set

$FP(\%)$	$FN(\%)$	DL (Number of images)
16	4	161.0

Concluding Remarks

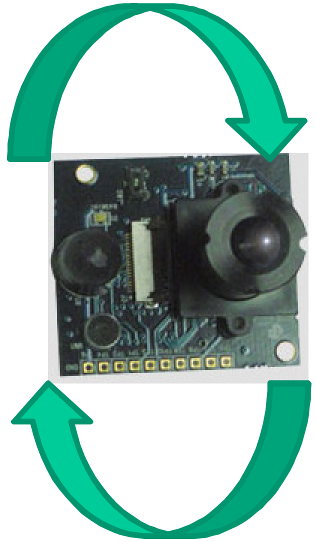
- When processing video sequences, the ***FP*** are typically determined by the presence of occluding objects, whenever these did not appear in the training set

- We need a training set which is **representative** of the scenario.

- In case some user-supervised information is available, this could be integrated by the test.

Ongoing Works

- Development of an ligheter test to be implemented directcly on the node.
- The nodes are able to monitor by themselves the degradation process
- Integration of light/time information in the test



Questions?