POLITECNICO DI MILANO

Near-Duplicate Video Detection Exploiting Noise Residual Traces

S. Lameri, L. Bondi, P. Bestagini, S. Tubaro

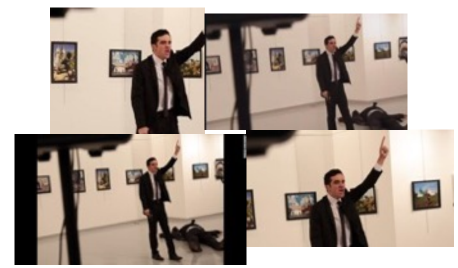
Motivations

- Increasing amount of **user-generated** content online

Motivations

- Increasing amount of **user-generated** content online

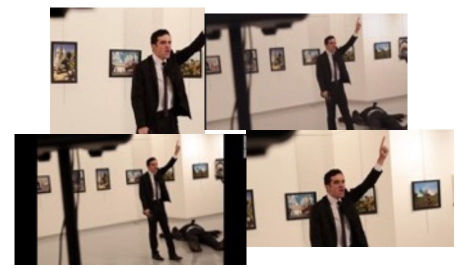
- The **same original content** can be edited and republished several times, thus generating different **near-duplicate** objects



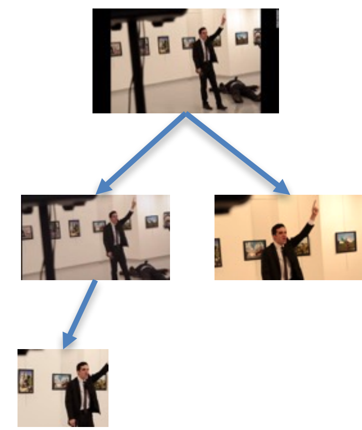
Motivations

- Increasing amount of **user-generated** content online

- The **same original content** can be edited and republished several times, thus generating different **near-duplicate** objects



- **Video Phylogeny** jointly analyses multiple versions of the same object
  - To identify the original content that give birth to the NDs
  - To infer the generative structure behind NDs creation

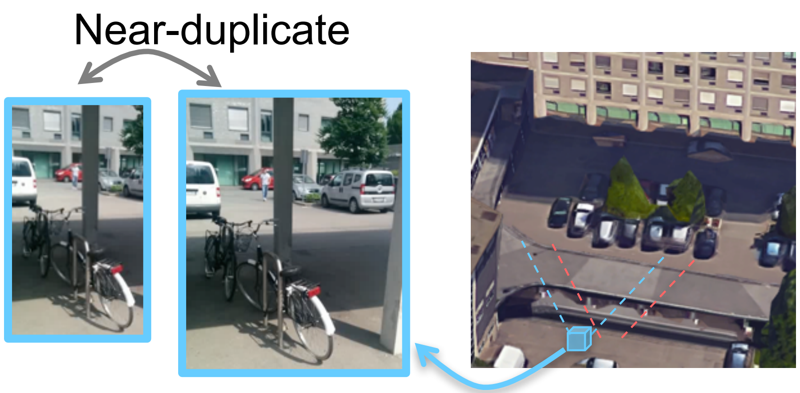


Problem Formulation

- Fundamental step in video phylogeny application is the **detection** of set of **near-duplicate** videos

Problem Formulation

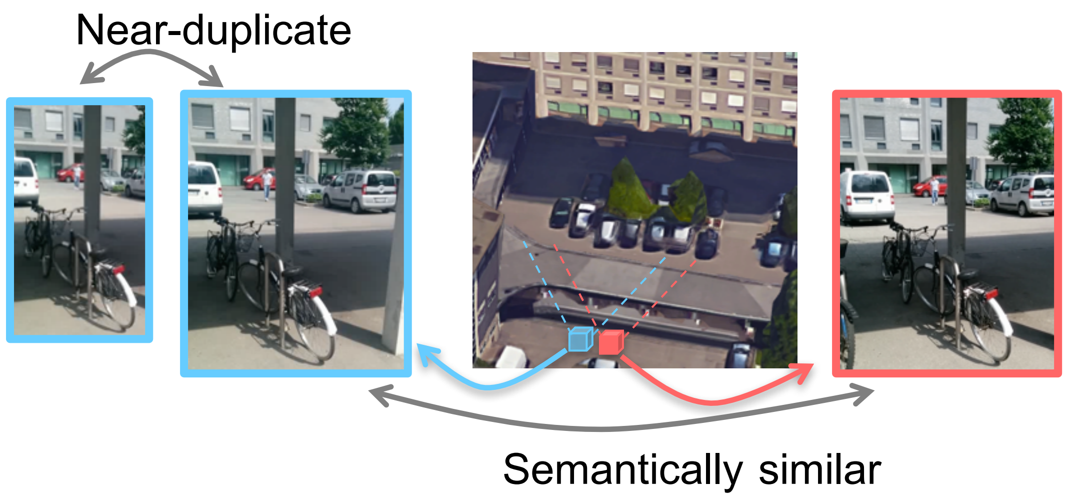
- Fundamental step in video phylogeny application is the **detection** of set of **near-duplicate** videos
- *Scenario*: We can find several videos depicting the **same scene**



- **Near-duplicates (ND)**, are edited copies of the **same video**

Problem Formulation

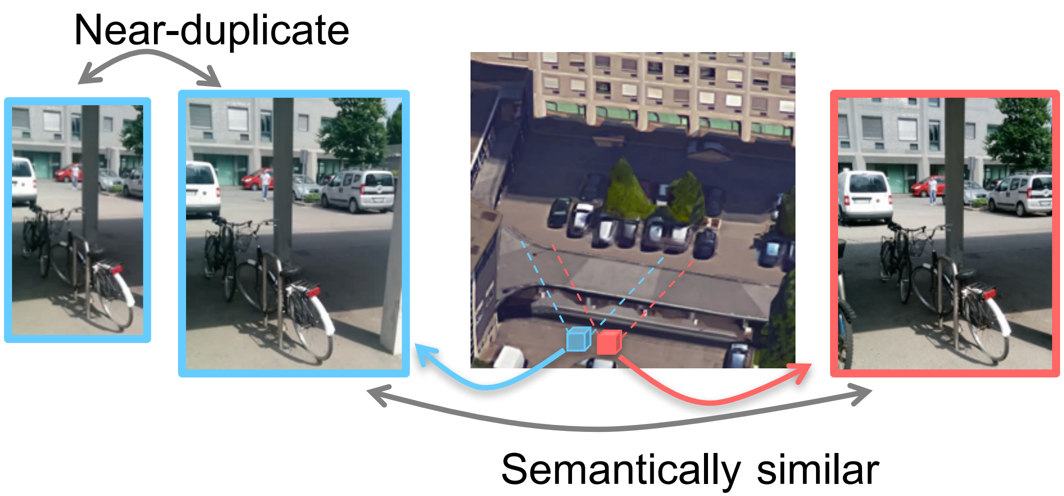
- Fundamental step in video phylogeny application is the **detection** of set of **near-duplicate** videos
- *Scenario*: We can find several videos depicting the **same scene**



- **Near-duplicates (ND)**, are edited copies of the **same video**
- **Semantically similar (SSI)** videos capture the scene from different view points, with **different devices**

Problem Formulation

- Fundamental step in video phylogeny application is the **detection** of set of **near-duplicate** videos
- *Scenario*: We can find several videos depicting the **same scene**



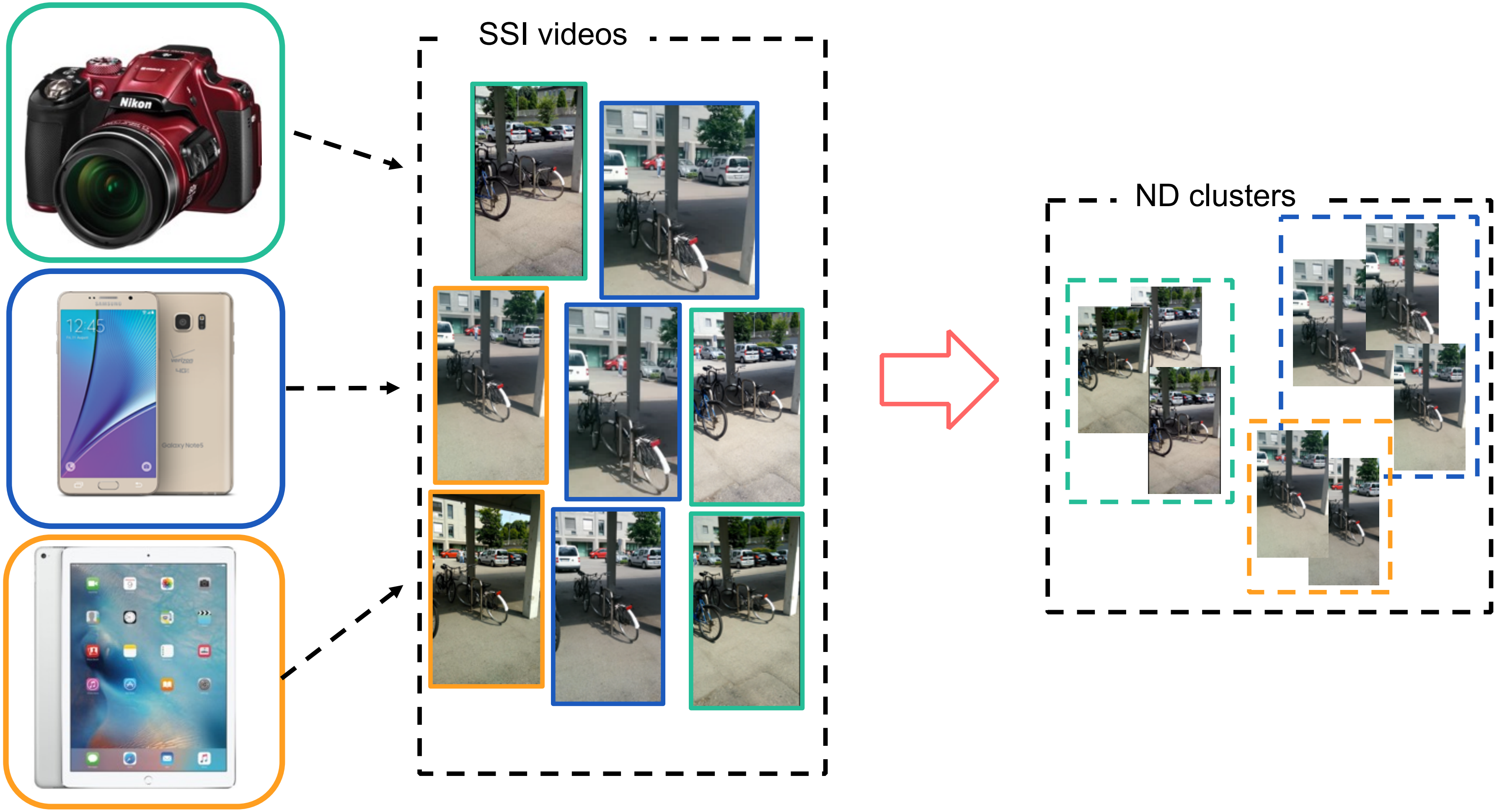
  - **Near-duplicates (ND)**, are edited copies of the **same video**
  - **Semantically similar (SSI)** videos capture the scene from different view points, with **different devices**
- *Problem*: SSI videos can be confused with ND

Goal

- *Goal*: Given a **pool of video** depicting the same scene, we want to detect pool of **ND** videos, distinguish them form **SSI**

Goal

- *Goal*: Given a **pool of video** depicting the same scene, we want to detect pool of **ND** videos, distinguish them form **SSI**



Basic Idea

- By definition **ND** videos
  - Depict the **same scene**
  - Are acquired by the **same device**
- Conversely **SSI** videos are acquired by **different devices**

Basic Idea

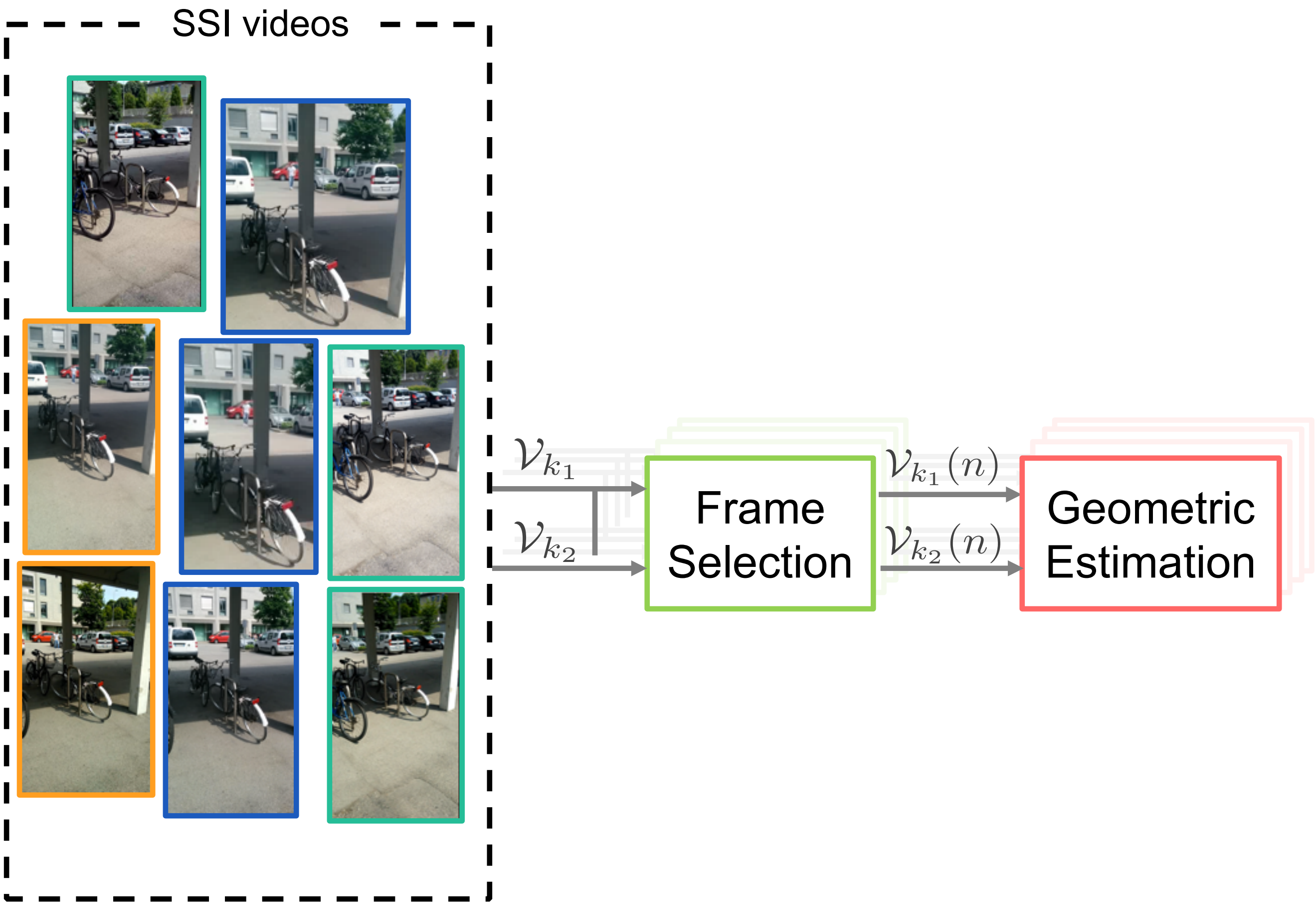
- By definition **ND** videos
  - Depict the **same scene**
  - Are acquired by the **same device**
- Conversely **SSI** videos are acquired by **different devices**

- ***ND clustering based on sensor noise analysis***

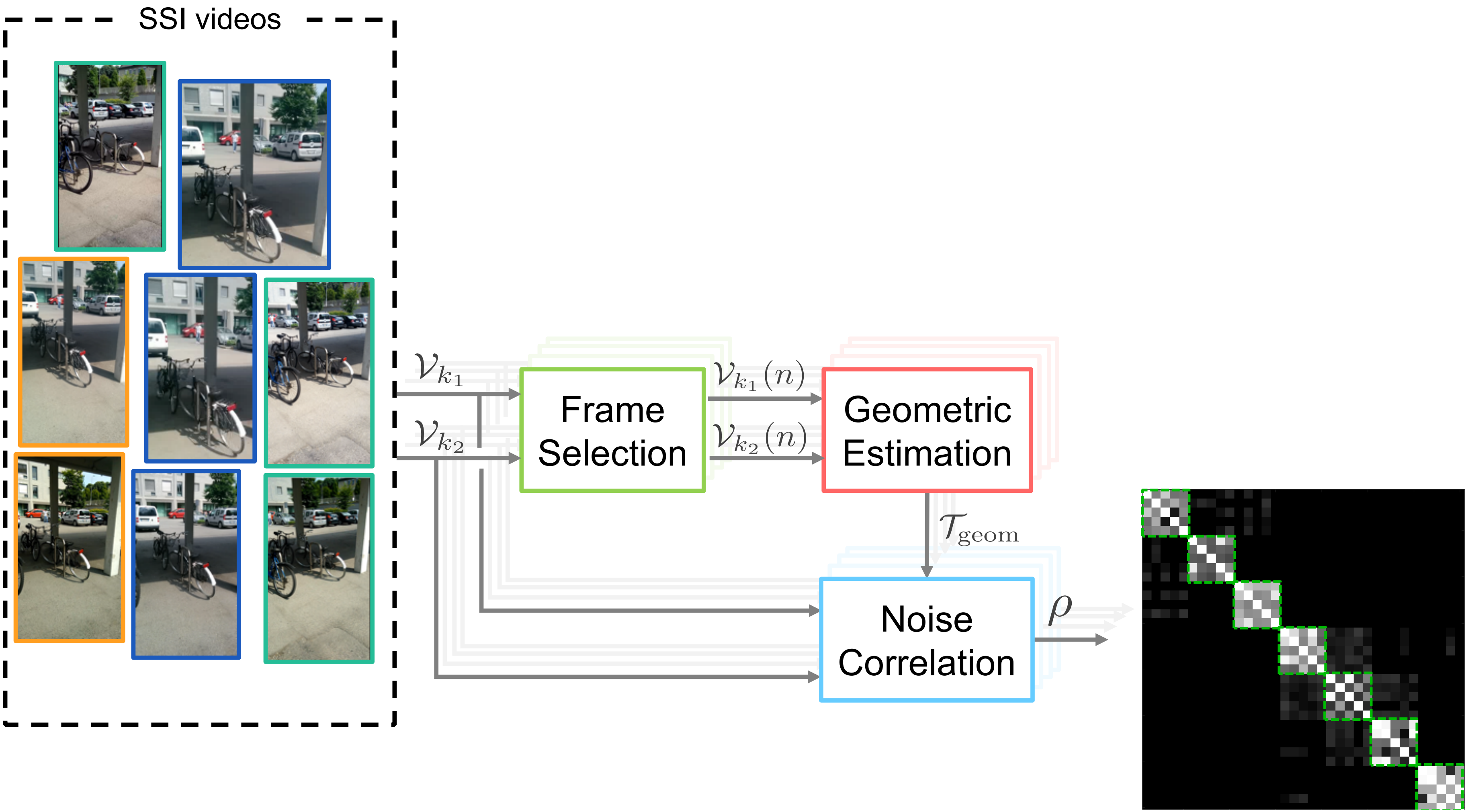
- *What is sensor noise?*
  - Due to its imperfections, every sensor cast a very weak noise-like pattern on every image it takes
  - This noise pattern plays the role of a **sensor fingerprint**

Proposed Algorithm

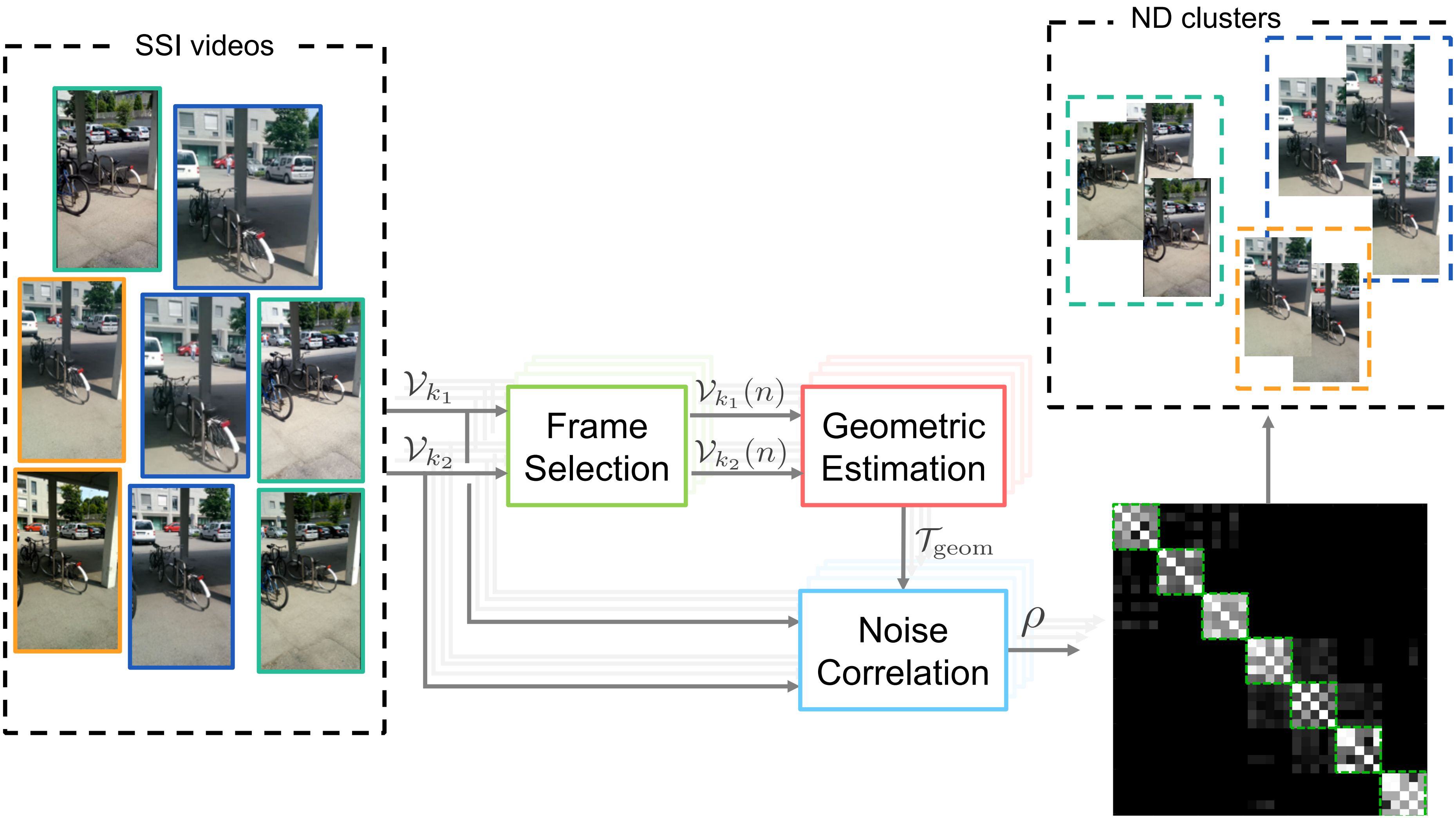
Proposed Algorithm



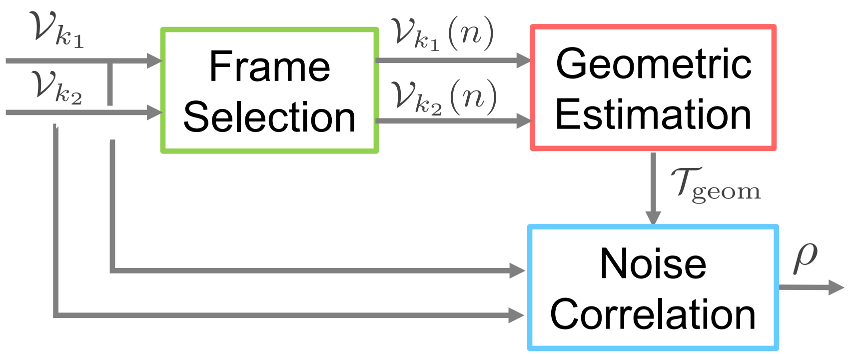
Proposed Algorithm



Proposed Algorithm

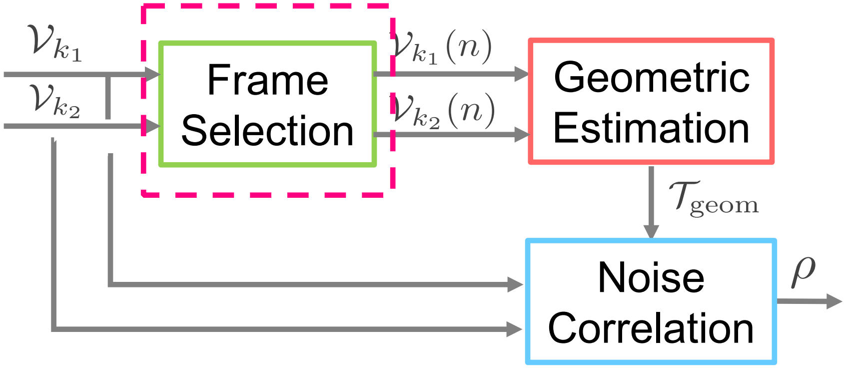


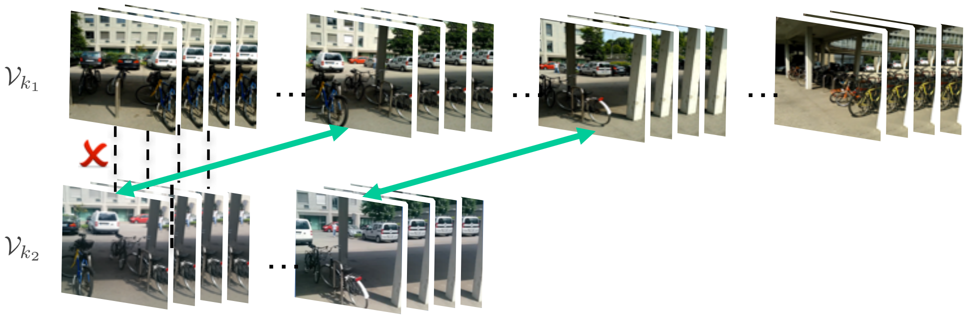
Proposed Algorithm



- Each pair of videos $\mathcal{V}_{k_1}$, $\mathcal{V}_{k_2}$ is processed separately

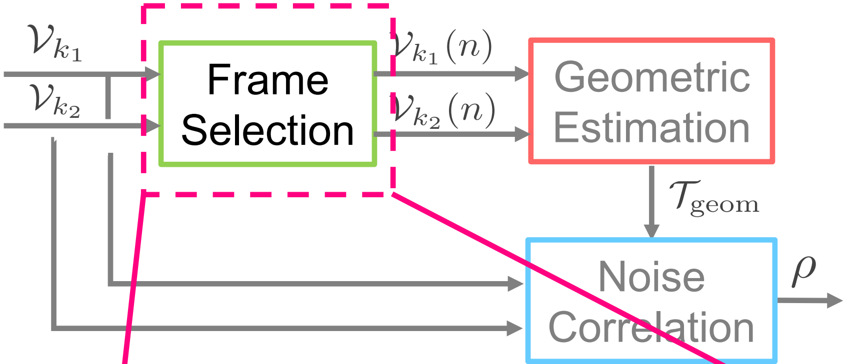
Proposed Algorithm
Frame Selection



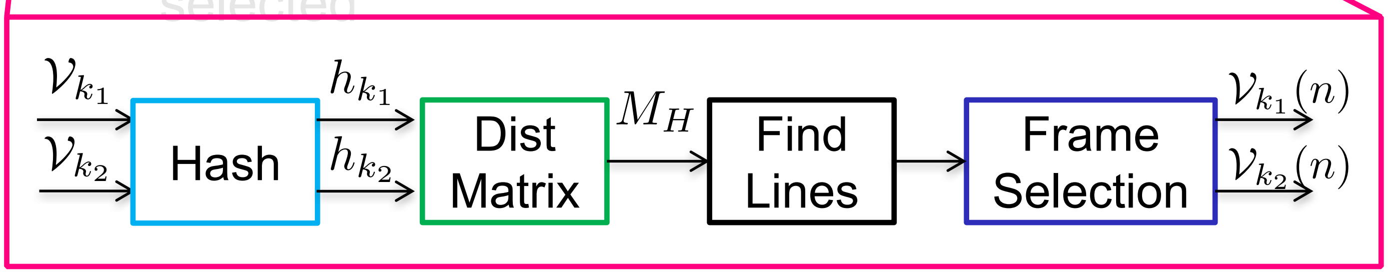


- Each pair of videos $\mathcal{V}_{k_1}, \mathcal{V}_{k_2}$ is processed separately
  - A pair of **synchronized frames** $\mathcal{V}_{k_1}(n), \mathcal{V}_{k_2}(n)$ is detected and selected

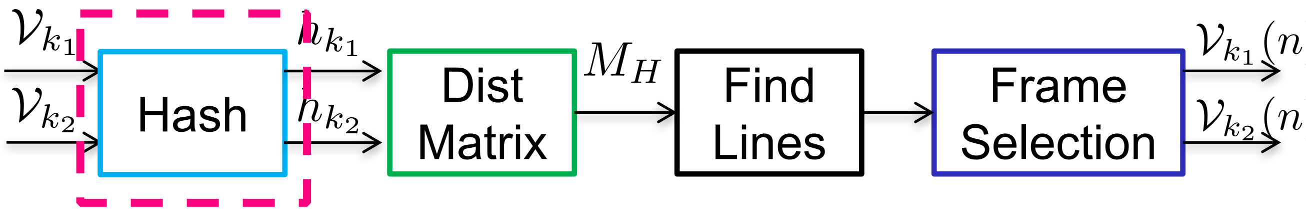
Proposed Algorithm
Frame Selection



- Each pair of videos $\mathcal{V}_{k_1}$, $\mathcal{V}_{k_2}$ is processed separately
  - A pair of **synchronized frames** $\mathcal{V}_{k_1}(n)$, $\mathcal{V}_{k_2}(n)$ is detected and selected



Proposed Algorithm
Frame Selection

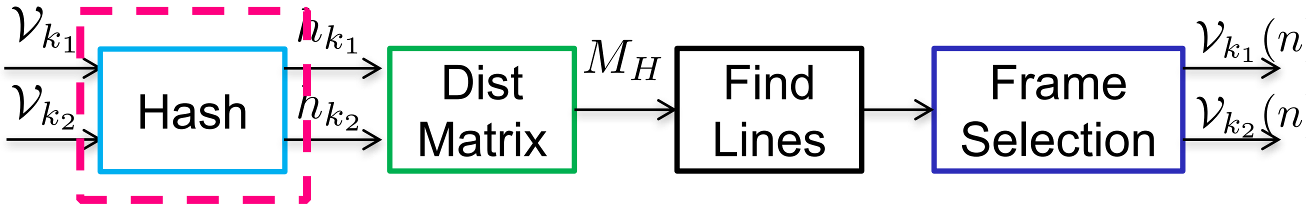


Video content over time is described by **binary hashes**

$\mathcal{V}_{k_1}$

$$h^1_{k_1} = 1001\ 0010\ \ldots\ 1011$$

Proposed Algorithm
Frame Selection



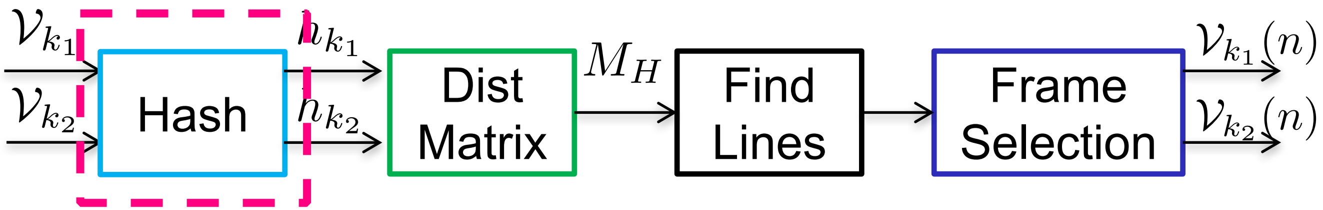
Video content over time is described by **binary hashes**

$\mathcal{V}_{k_1}$

$$h^1_{k_1} = 1001\ 0010\ \ldots\ 1011$$

$$h^2_{k_1} = 1011\ 0110\ \ldots\ 0010$$

Proposed Algorithm
Frame Selection



Video content over time is described by **binary hashes**

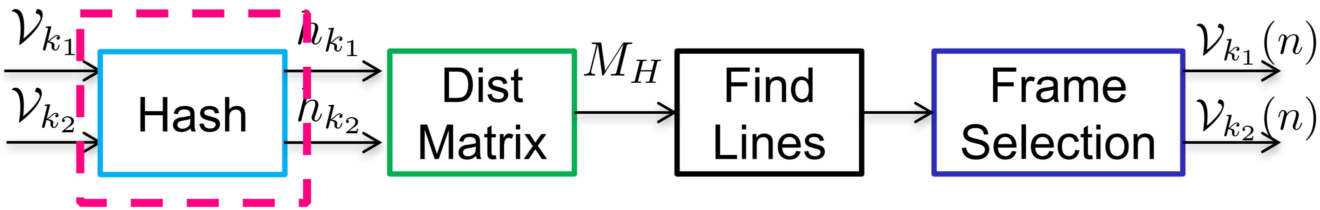
$\mathcal{V}_{k_1}$

$$h_{k_1}^1 = 1001\ 0010\ \ldots\ 1011$$

$$h_{k_1}^2 = 1011\ 0110\ \ldots\ 0010$$

$$h_{k_1}^3 = 1110\ 1110\ \ldots\ 1010$$

Proposed Algorithm
Frame Selection



Video content over time is described by **binary hashes**

$\mathcal{V}_{k_1}$

$$h_{k_1}^{1} = 1001\ 0010\ \ldots\ 1011$$

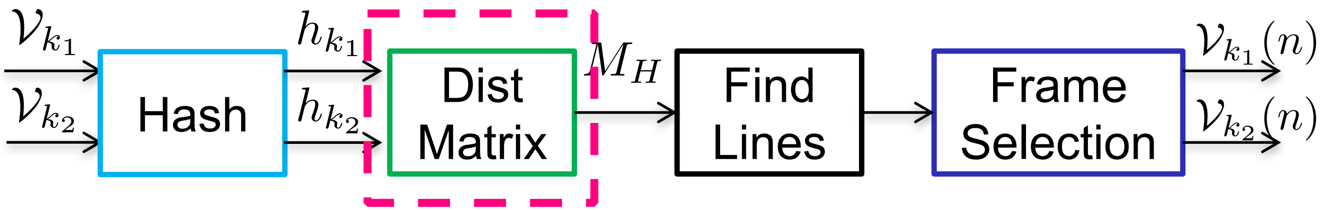
$$h_{k_1}^{2} = 1011\ 0110\ \ldots\ 0010$$

$$h_{k_1}^{3} = 1110\ 1110\ \ldots\ 1010$$

$$\ldots$$

Proposed Algorithm

Frame Selection



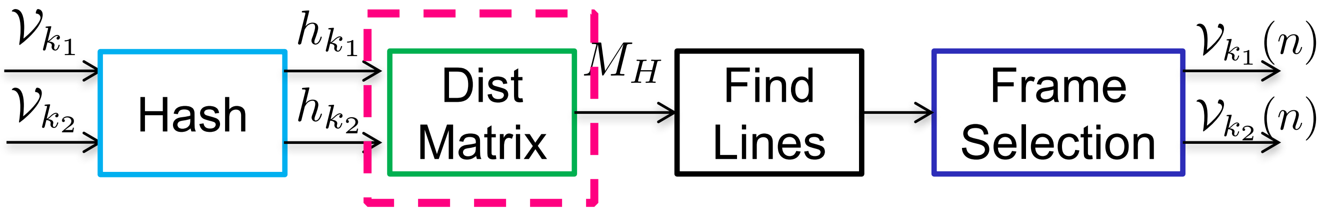
Pairs of hashes from different sequences are compared through **Hamming distance**

$\mathcal{V}_{k_1}$

$\mathcal{V}_{k_2}$

$$
\begin{array}{lcl}
1001\;0010\;\ldots\;1011 & \oplus & 0001\;0011\;\ldots\;1011 = 4 \\
1011\;0110\;\ldots\;0010 & \oplus & 1111\;0110\;\ldots\;0010 = 30 \\
\ldots & & \ldots \\
1110\;1110\;\ldots\;1010 & \oplus & 1110\;1110\;\ldots\;1010 = 0
\end{array}
$$

Proposed Algorithm
Frame Selection



Pairs of hashes from different sequences are compared through **Hamming distance**

$\mathcal{V}_{k_1}$

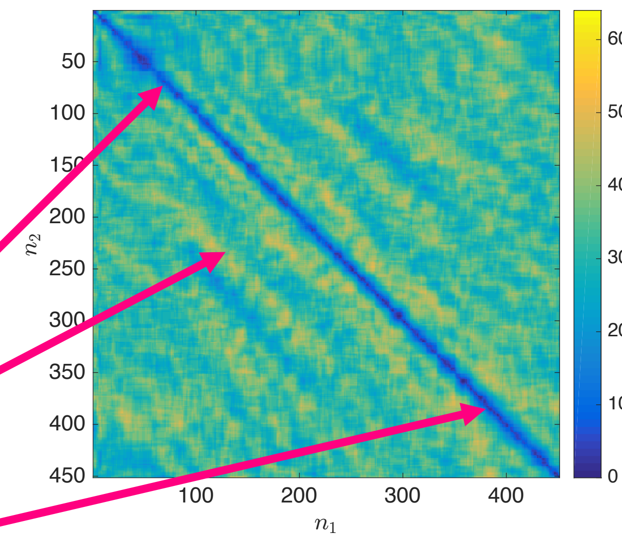
$\mathcal{V}_{k_2}$

$$1001\ 0010\ \dots\ 1011 \quad \oplus \quad 0001\ 0011\ \dots\ 1011 = 4$$

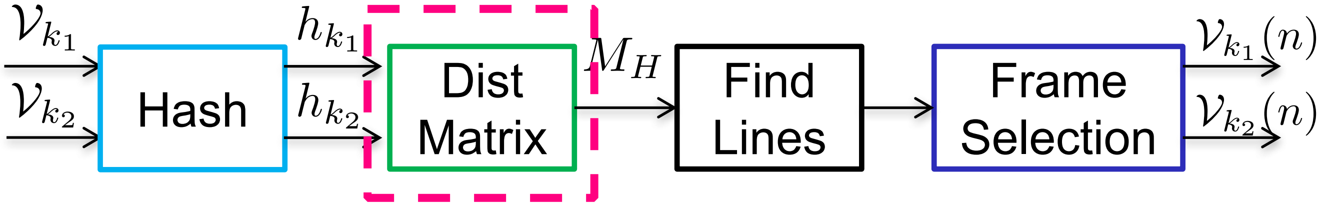
$$1011\ 0110\ \dots\ 0010 \quad \oplus \quad 1111\ 0110\ \dots\ 0010 = 30$$

$$\dots \qquad\qquad \dots$$

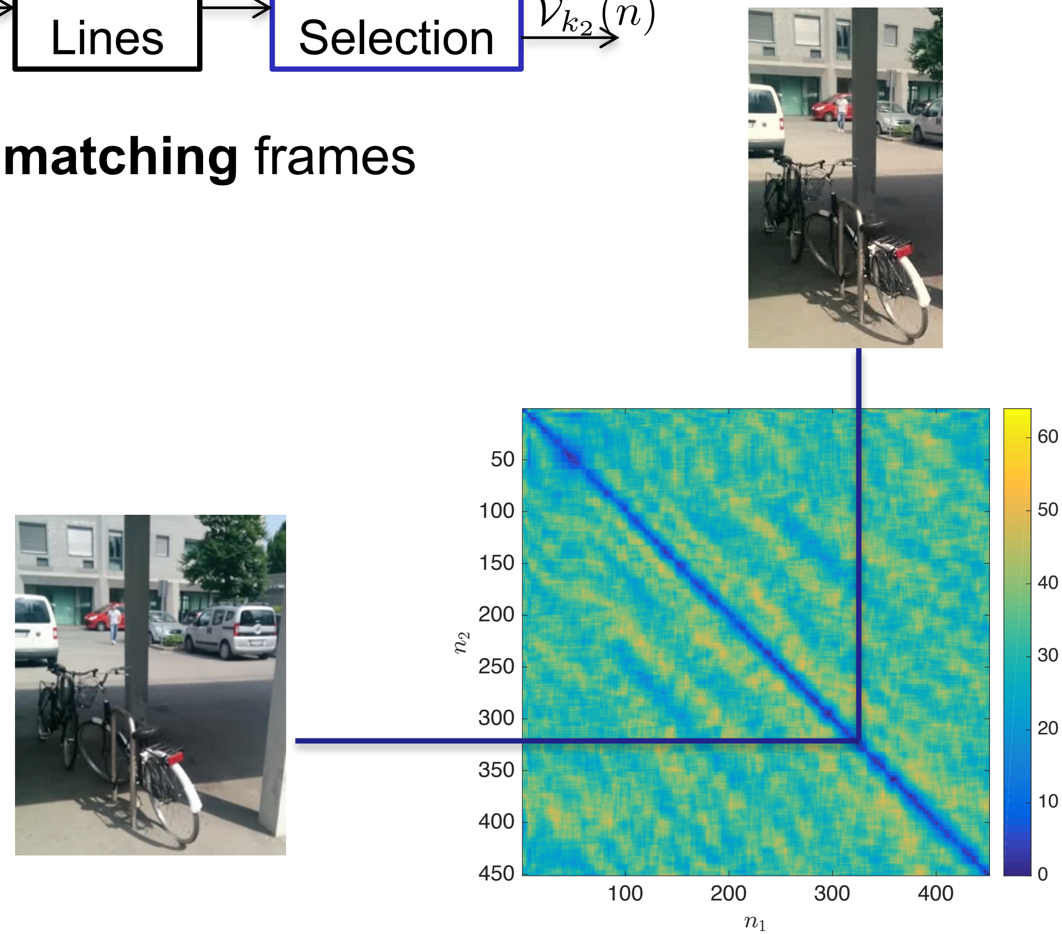
$$1110\ 1110\ \dots\ 1010 \quad \oplus \quad 1110\ 1110\ \dots\ 1010 = 0$$



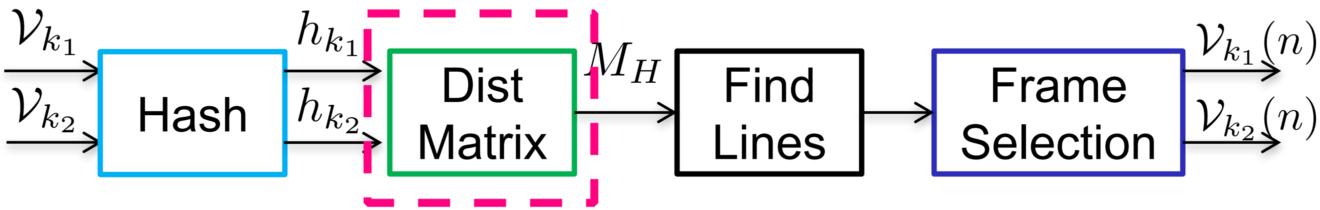
Proposed Algorithm
Frame Selection



Low hash distance indicates **matching** frames

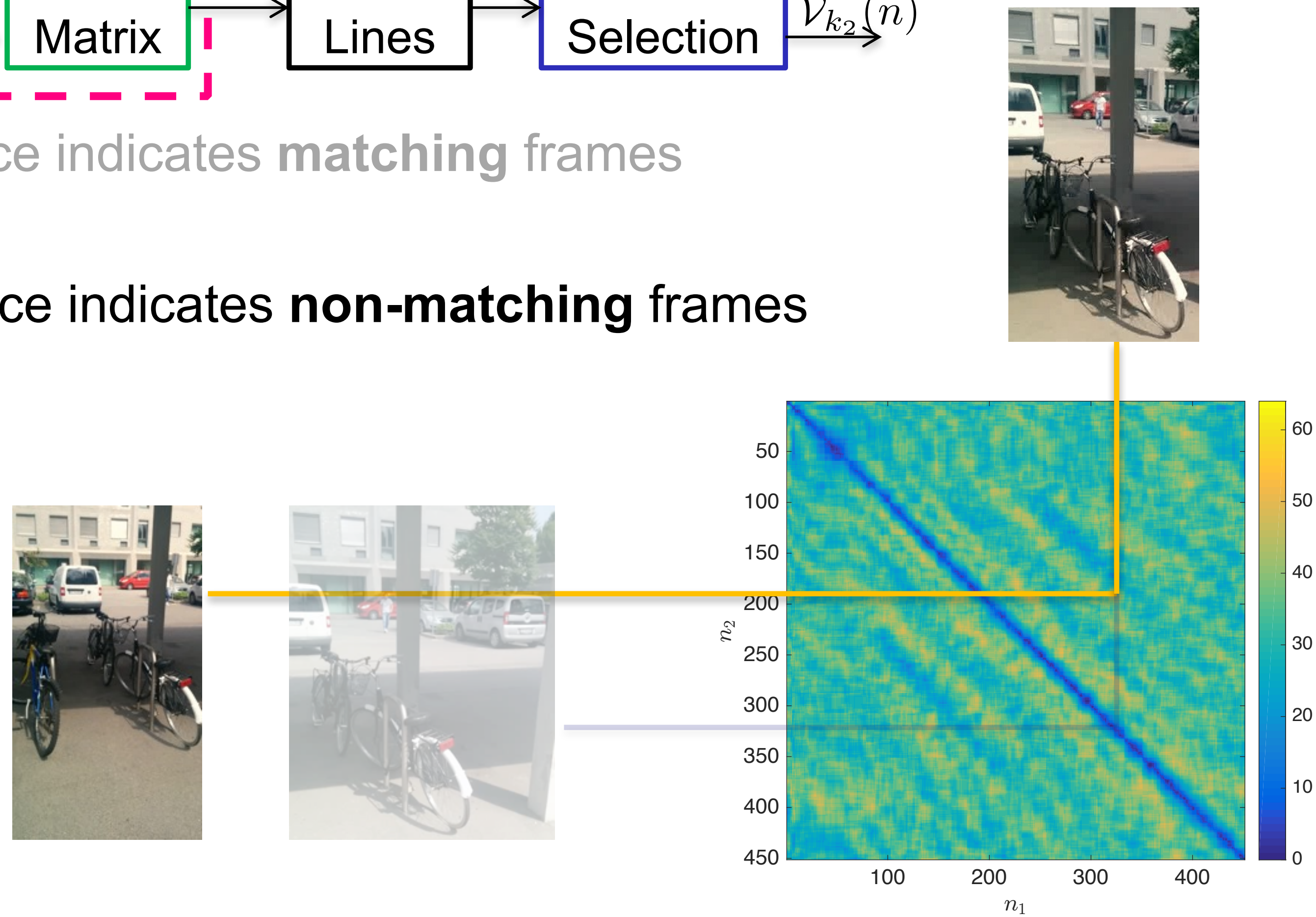


Proposed Algorithm
Frame Selection

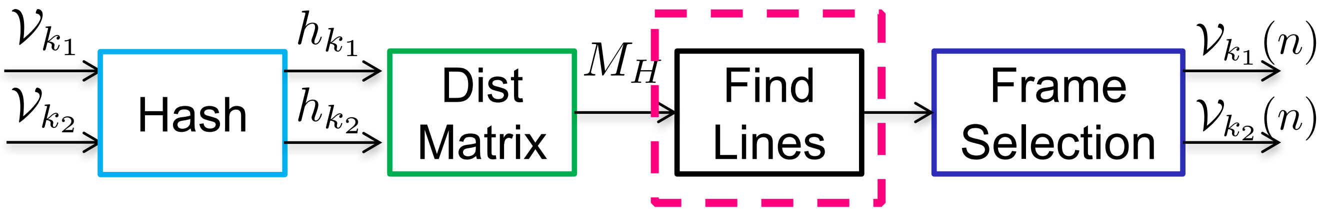


Low hash distance indicates **matching** frames

High hash distance indicates **non-matching** frames

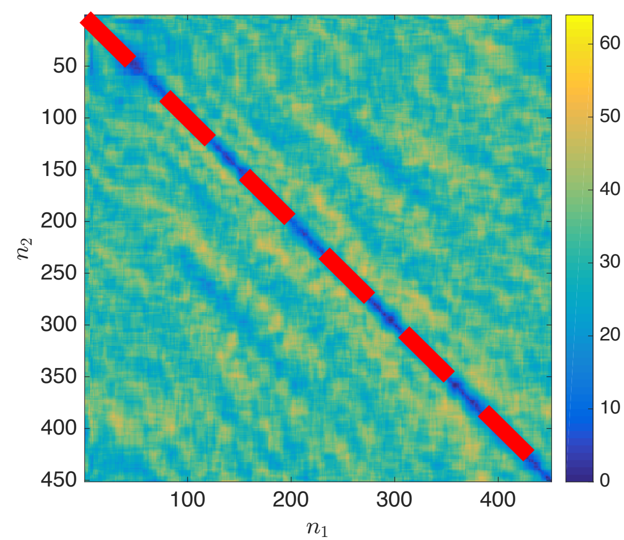


Proposed Algorithm
Frame Selection

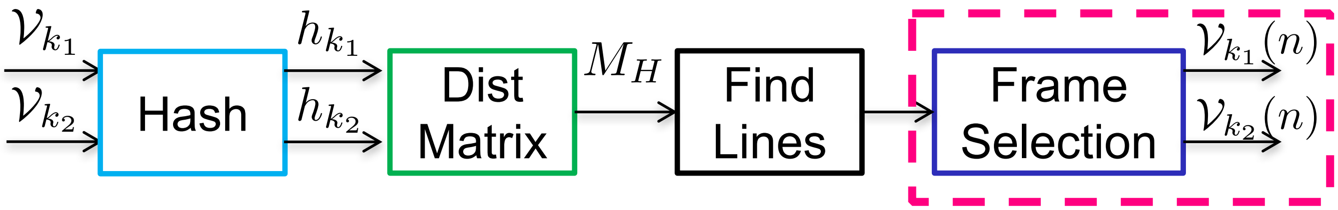


Frame selection procedure:

- Aligned low values are detected as they indicate **matching** frames

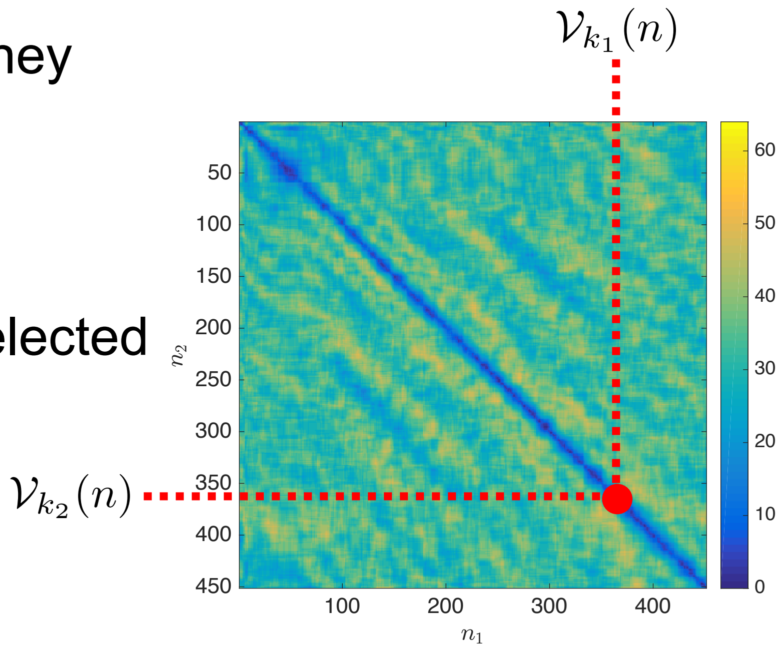


Proposed Algorithm
Frame Selection

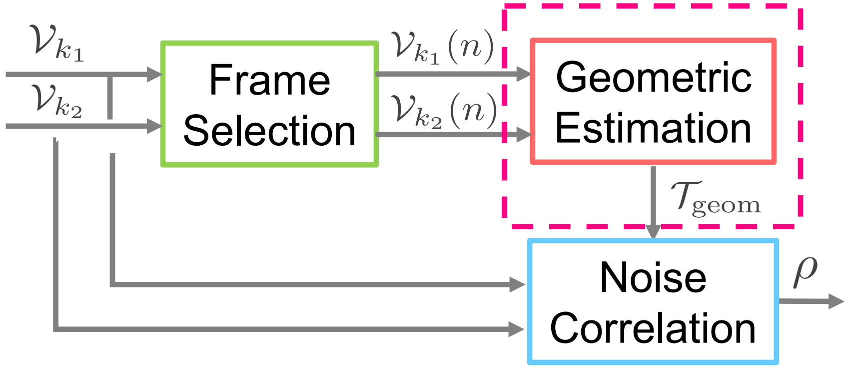


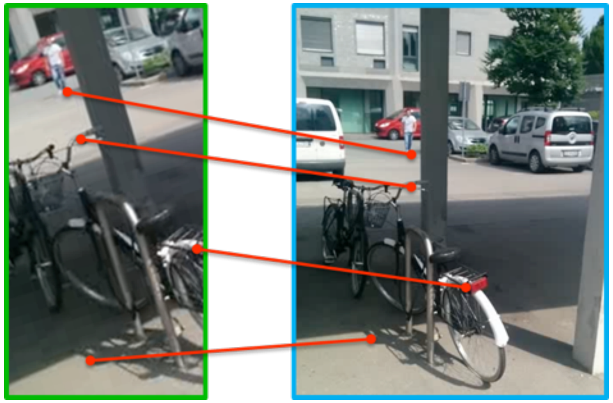
Frame selection procedure:

- Aligned low values are detected as they indicate **matching** frames
- The pair of matching frames with **minimum hash distance** value is selected



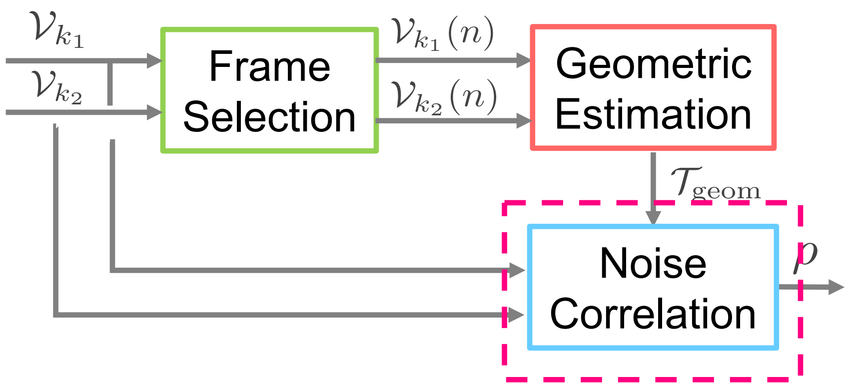
Proposed Algorithm
Geometric Estimation

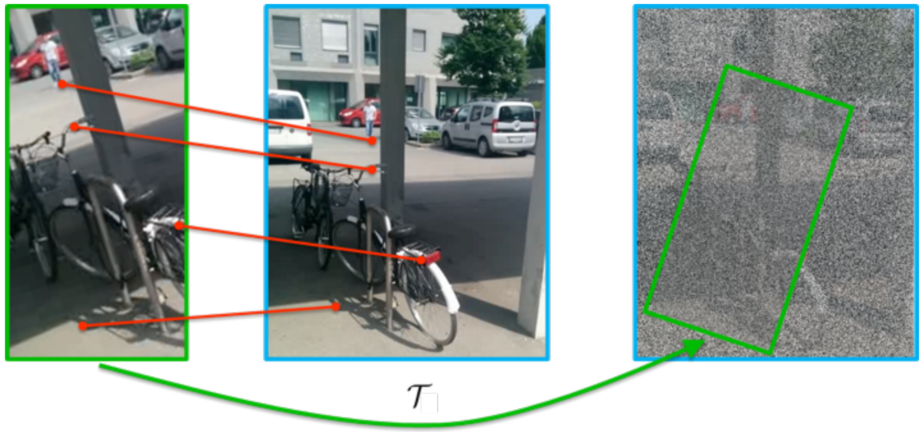




- Each pair of videos $\mathcal{V}_{k_1}, \mathcal{V}_{k_2}$ is processed separately
  - A pair of **synchronized frames** $\mathcal{V}_{k_1}(n), \mathcal{V}_{k_2}(n)$ is detected and selected
  - **Geometric transformation** is estimated between $\mathcal{V}_{k_1}(n)$ and $\mathcal{V}_{k_2}(n)$

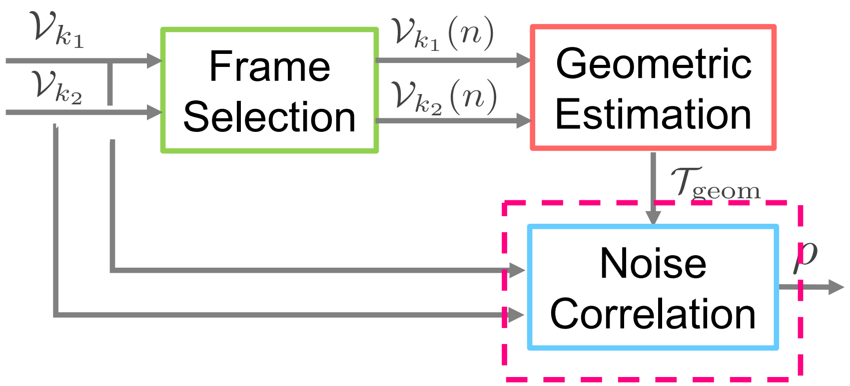
Proposed Algorithm
Noise Correlation

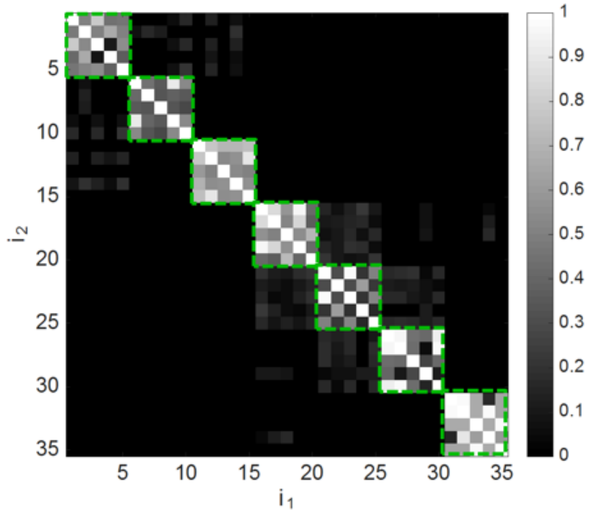




- Each pair of videos $\mathcal{V}_{k_1}$, $\mathcal{V}_{k_2}$ is processed separately
  - A pair of **synchronized frames** $\mathcal{V}_{k_1}(n)$, $\mathcal{V}_{k_2}(n)$ is detected and selected
  - **Geometric transformation** is estimated between $\mathcal{V}_{k_1}(n)$ and $\mathcal{V}_{k_2}(n)$
  - Noises are extracted from $\mathcal{V}_{k_1}$, $\mathcal{V}_{k_2}$ with a **PRNU** estimator, **registered** and correlated

Proposed Algorithm
Noise Correlation





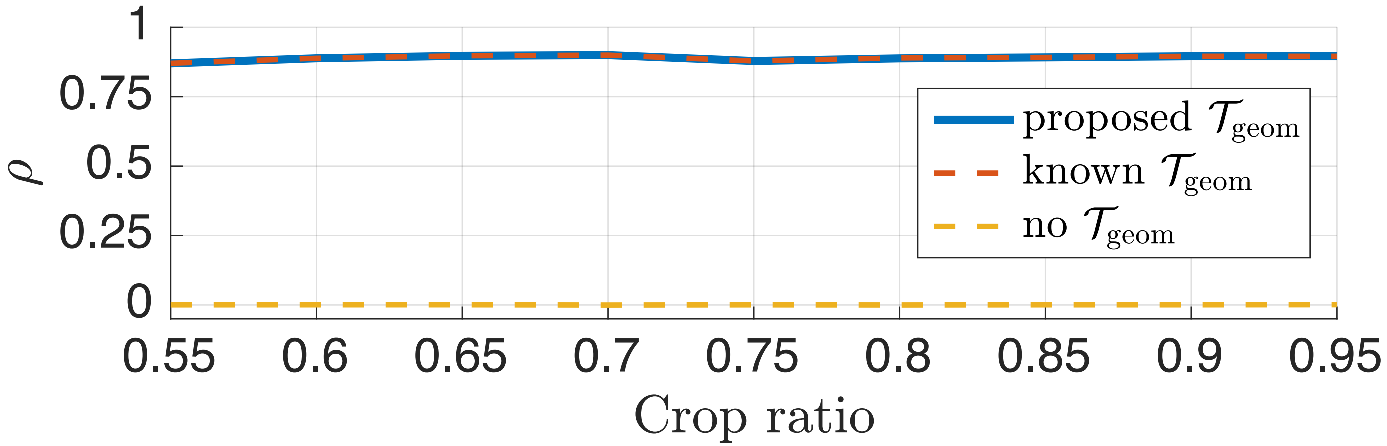
- Each pair of videos $\mathcal{V}_{k_1}, \mathcal{V}_{k_2}$ is processed separately
  - A pair of **synchronized frames** $\mathcal{V}_{k_1}(n), \mathcal{V}_{k_2}(n)$ is detected and selected
  - **Geometric transformation** is estimated between $\mathcal{V}_{k_1}(n)$ and $\mathcal{V}_{k_2}(n)$
  - Noises are extracted from $\mathcal{V}_{k_1}$, $\mathcal{V}_{k_2}$ with a **PRNU** estimator, **registered** and correlated, to build the noise **correlation matrix**

Experiments and results
Acquisitions

- 9 SSI videos acquired with 7 different smartphones
  - different view points and rotations, same filmed object
  - 15s to 40s sequence
  - some devices are the same model
  - no temporal synchronization
  - all videos resized to 640x360

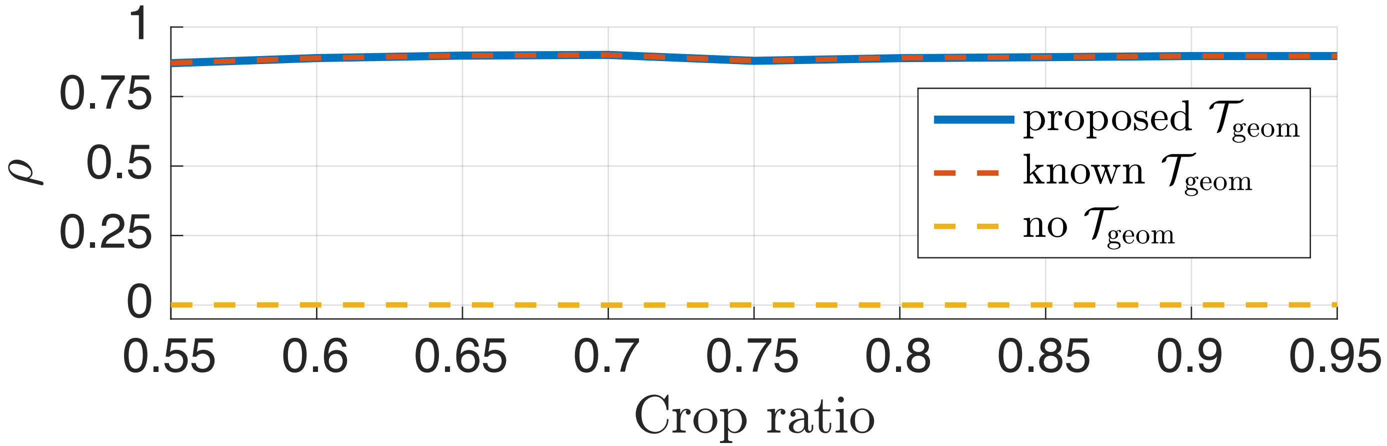
Experiments and results
Geometric Estimation effectiveness

- *Crop dataset*: 693 ND with cropping ranging from 55% to 100%

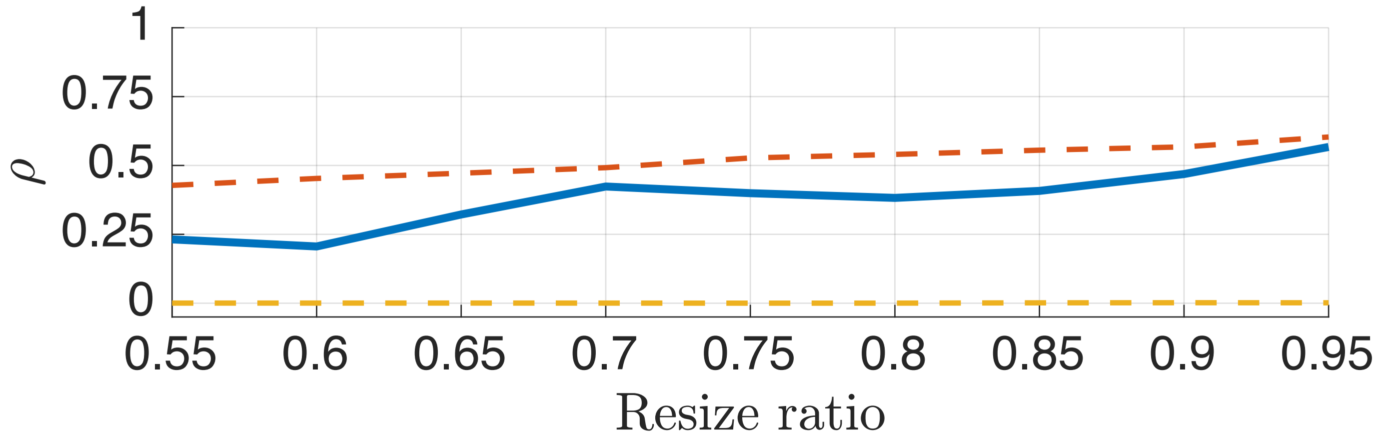


Geometric Estimation effectiveness

- *Crop dataset*: 693 ND with cropping ranging from 55% to 100%

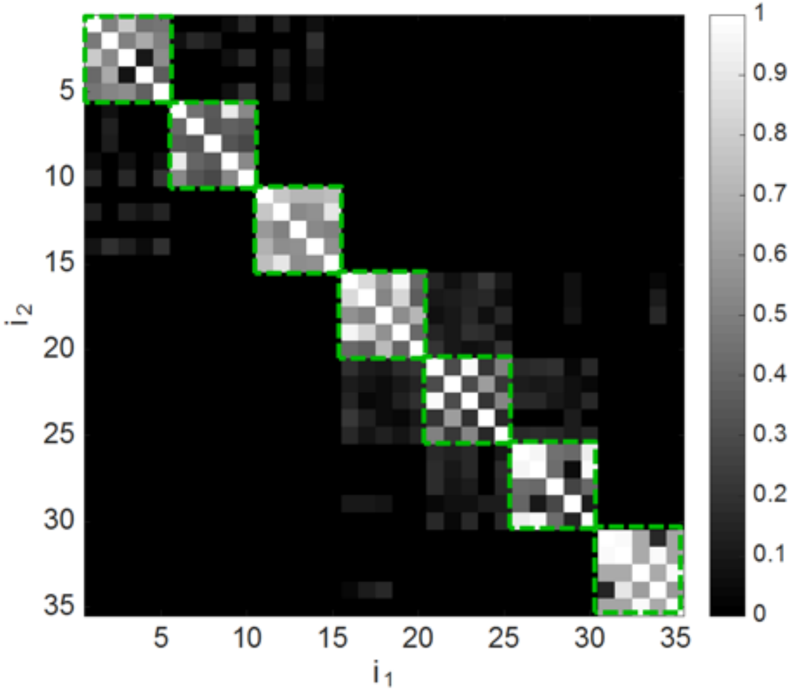


- *Resize dataset*: 693 ND with resizing ranging from 55% to 100%



Experiments and results
Clustering capabilities

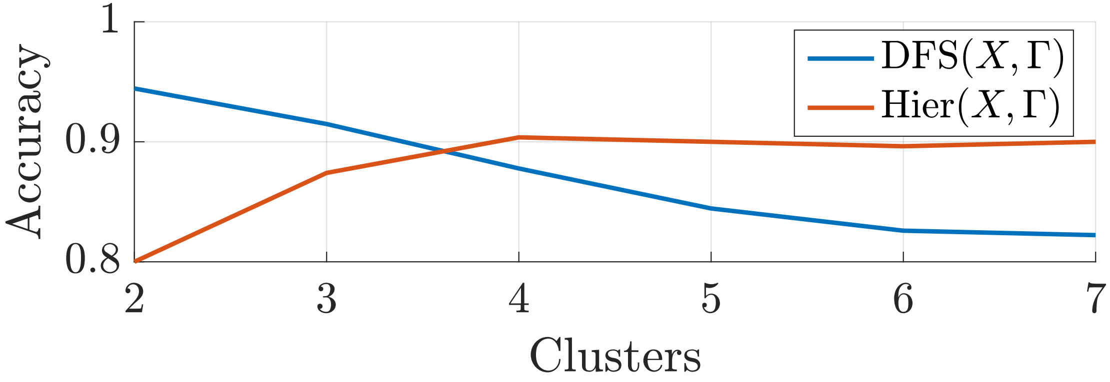
- *Clustering datasets*
  - 6 datasets with 2-7 ND clusters
  - Transformations obtained combining contrast enhancement, brightness adjustment, spatial cropping, resizing
  - More than 12k videos in total



- *Clustering approaches*
  - Depth First Search (DFS)
  - Hierarchical (Hier)

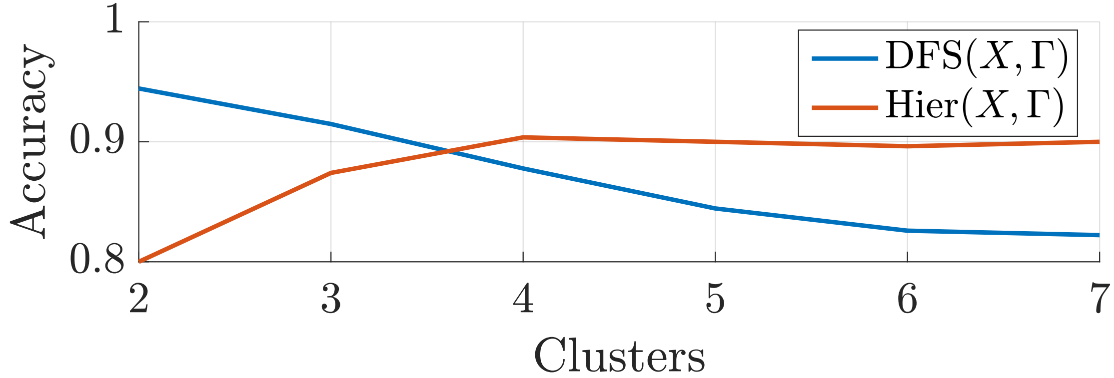
Experiments and results
Clustering capabilities

- Accuracy in detecting the **number of clusters**

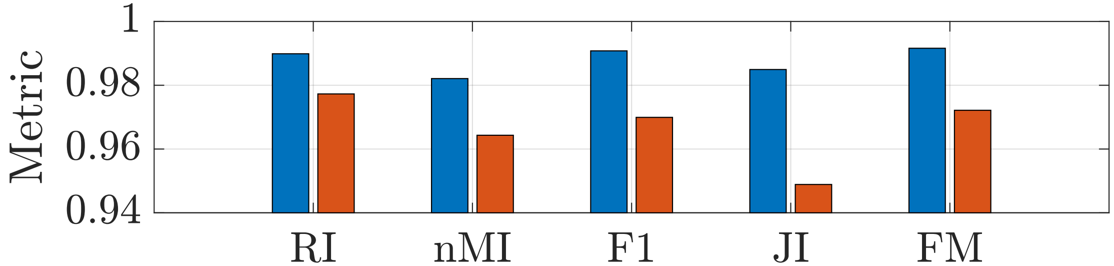


Clustering capabilities

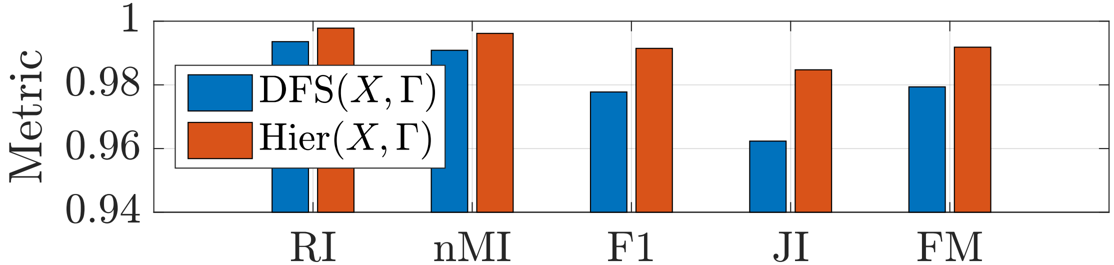
- Accuracy in detecting the **number of clusters**



- How well pairs of ND are assigned to the same cluster



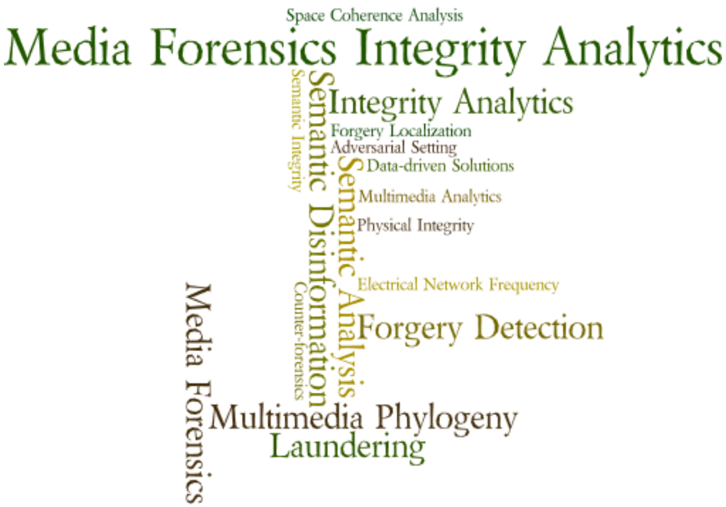
2 clusters



7 clusters

Conclusions

- We faced the problem of detecting **pool of near-duplicate** videos
- We proposed a pipeline based on the analysis of **noise residual** traces that
  - Disambiguates semantically similar videos form ND ones
  - Clusters together near-duplicate videos
- We verified the possibility of **geometrically registering** NDs noise residuals based on **keypoint matching**
- We showed the performances of **clustering** on different data realizations in terms of
  - Correctly detect the number of clusters
  - Separate SSI videos that are not NDs



Sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173

More information:
https://engineering.purdue.edu/MEDIFOR/

UNICAMP

POLITECNICO MILANO 1863

Near-Duplicate Video Detection Exploiting Noise Residual Traces

S. Lameri, L. Bondi, P. Bestagini, S. Tubaro