Content-Based Video Copy Detection: PRISMA at TRECVID 2010

Juan Manuel Barrios and Benjamin Bustos

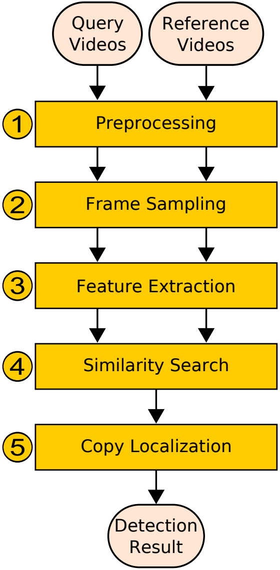
PRISMA Research Group
Department of Computer Science
University of Chile
{jbarrios,bebustos}@dcc.uchile.cl

November 17, 2010

PRISMA System Overview

- Copy Detection System developed for TRECVID 2010.
- Three Global descriptors.
- No Audio information.
- Pivot-based index with approximate search.
- Voting algorithm for copy localization.
- Implemented in C with OpenCV library.
- System divided in five tasks/steps.

PRISMA System Overview

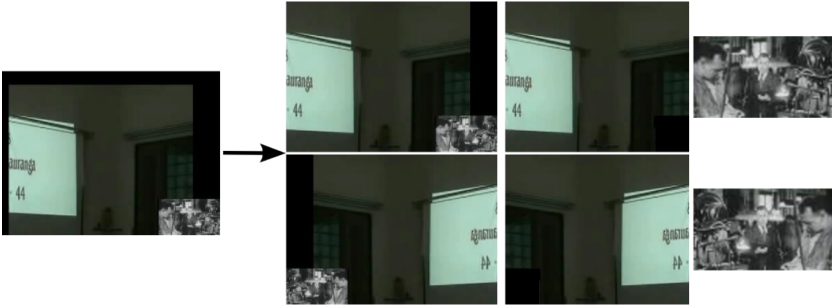


System Tasks

1. **Preprocessing**:
   - Skip irrelevant frames.
   - Remove black borders.
   - Inverse transformations for Camcording, PIP and Flip.

   Query videos increased from 1,608 to 5,378.

   Reference videos kept in 11,524.



System Tasks

2. **Frame Sampling**:
   - Divides each video in groups of similar consecutive frames (GF).
   - Uniform subsampling of 3 frames per second.
   - Similarity between frames defined as maximum difference between intensity of pixels.

Query Videos are divided into 1,000,000 groups.

Reference Videos are divided into 4,000,000 groups.

System Tasks

2. **Frame Sampling**:
   - Divides each video in groups of similar consecutive frames (GF).
   - Uniform subsampling of 3 frames per second.
   - Similarity between frames defined as maximum difference between intensity of pixels.

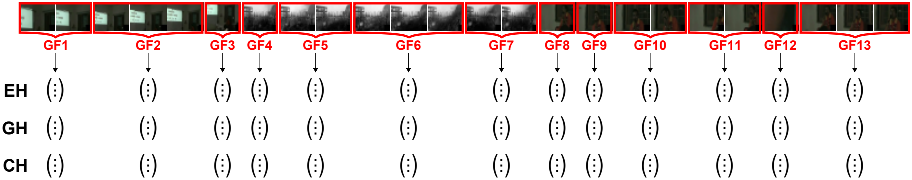
Query Videos are divided into 1,000,000 groups.

Reference Videos are divided into 4,000,000 groups.

System Tasks

3. **Feature Extraction**:
   - Descriptor of a group is the average of descriptors for each frame.
   - Extracts three global visual descriptors :
     - EH: Edge Histogram ($4 \times 4 \times 10 = 160$ dimensions)
     - GH: Gray Histogram ($3 \times 3 \times 20 = 180$ dimensions)
     - CH: RGB Histogram ($2 \times 2 \times 48 = 192$ dimensions)

     (1 byte per dimension)



System Tasks

4. **Similarity Search**:
   - Compares descriptors from query groups with descriptors from reference groups.
   - $DIST(G_i, G_j)$ is a distance function that measures the similarity between groups G_i and G_j.
   - $DIST$ is defined as a combination of two descriptors:
     - Run `ehdNgryhst`: $DIST$ combines EH and GH.
     - Run `ehdNclrhst`: $DIST$ combines EH and CH.

Similarity Search Task

- Distance between groups is a static weighted combination of distance between descriptors (γ):

$$\delta(G_i, G_j) = w_1 \times \gamma_1(G_i, G_j) + w_2 \times \gamma_2(G_i, G_j)$$

- We defined γ as L_1 (Manhattan) distance for EHD, GH and CH vectors:

$$L_1(x, y) = \sum_{i=0}^{d} |x_i - y_i|$$

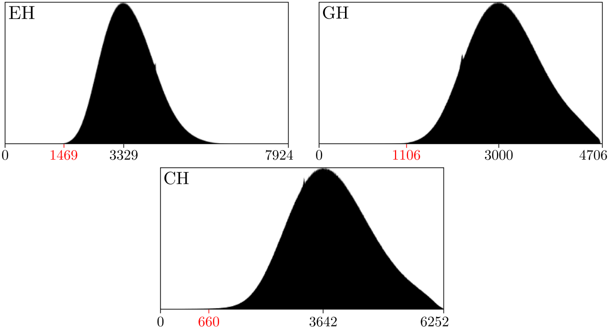
- Final distance between groups is the average of δ between three consecutive groups:

$$DIST(G_i, G_j) = \frac{\delta(G_{i-1}, G_{j-1}) + \delta(G_i, G_j) + \delta(G_{i+1}, G_{j+1})}{3}$$

- $DIST$ requires more than 1,000 operations to be evaluated.

Similarity Search Task

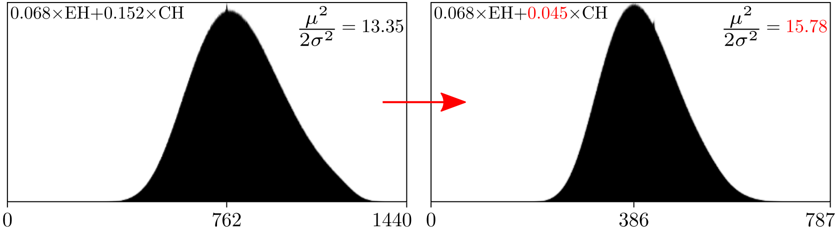
- We set weights for each descriptor using a histogram of distances between pairs of vectors.



- Weights normalize to 100 the distance that covers 0.01% of pairs on each histogram: $\frac{100}{1469} = 0.068$ $\frac{100}{1106} = 0.090$ $\frac{100}{660} = 0.152$
- `ehdNgryhst`: $\delta = 0.068 \times \text{EH} + 0.090 \times \text{GH}$
- `ehdNclrhst`: $\delta = 0.068 \times \text{EH} + 0.152 \times \text{CH}$

Similarity Search Task

- The intrinsic dimensionality $\frac{\mu^2}{2\sigma^2}$ quantifies how hard is to search on a metric space [Chávez et al, 2001].
- Move w_2 to a value that locally maximizes intrinsic dimensionality of δ.
- Iterative algorithm that converged to:
  - `ehdNgryhst`: $\delta = 0.068 \times \mathrm{EH} + 0.090 \times \mathrm{GH}$
  - `ehdNclrhst`: $\delta = 0.068 \times \mathrm{EH} + 0.045 \times \mathrm{CH}$



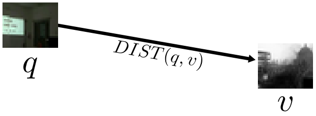
Similarity Search Task

- The output of the Similarity Search task is a Nearest-Neighbors Table with most similar reference groups for each query group.

Query	NN 1		NN 2		NN 3	
Query1Group1	Vid07_Grp54	dist	Vid08_Grp73	dist	Vid01_Grp68	dist
Query1Group2	Vid09_Grp13	dist	Vid02_Grp34	dist	Vid02_Grp33	dist
Query1Group3	Vid07_Grp34	dist	Vid03_Grp54	dist	Vid09_Grp14	dist
Query1Group4	Vid09_Grp15	dist	Vid02_Grp13	dist	Vid03_Grp65	dist
Query1Group5	Vid01_Grp88	dist	Vid01_Grp12	dist	Vid07_Grp58	dist
Query1Group6	Vid09_Grp54	dist	Vid09_Grp17	dist	Vid07_Grp59	dist
Query1Group7	Vid01_Grp45	dist	Vid03_Grp43	dist	Vid03_Grp20	dist
Query1Group8	Vid09_Grp19	dist	Vid01_Grp12	dist	Vid07_Grp61	dist
...	...		...		...	

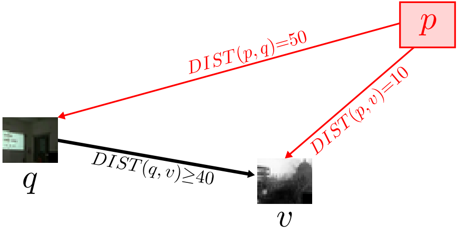
- A naive approach would evaluate 1,000,000 $\times$ 4,000,000 times $DIST$ (this takes about 11 month!).

- $DIST$ complies with metric properties: Reflexivity, Non-Negativity, Symmetry, and Triangle Inequality.
- Let q be a group of frames from a query video, and v be a group of frames from a reference video.
- A lower bound for $DIST(q, v)$ can be calculated with pivots:



Similarity Search Task

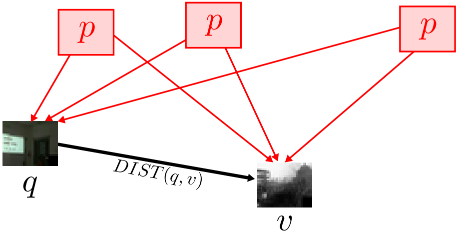
- $DIST$ complies with metric properties: Reflexivity, Non-Negativity, Symmetry, and Triangle Inequality.
- Let q be a group of frames from a query video, and v be a group of frames from a reference video.
- A lower bound for $DIST(q, v)$ can be calculated with pivots:



- Lower Bound: $DIST(q, v) \geq |DIST(p, q) - DIST(p, v)|$

Similarity Search Task

- $DIST$ complies with metric properties: Reflexivity, Non-Negativity, Symmetry, and Triangle Inequality.
- Let q be a group of frames from a query video, and v be a group of frames from a reference video.
- A lower bound for $DIST(q, v)$ can be calculated with pivots:

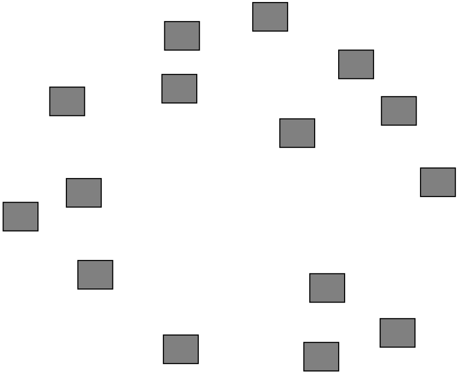


- Let $S = \{p_1, ..., p_m\}$ be a set of pivots, then:

$$DIST(q, v) \geq \max_{p \in S} \{|DIST(p, q) - DIST(p, v)|\}$$

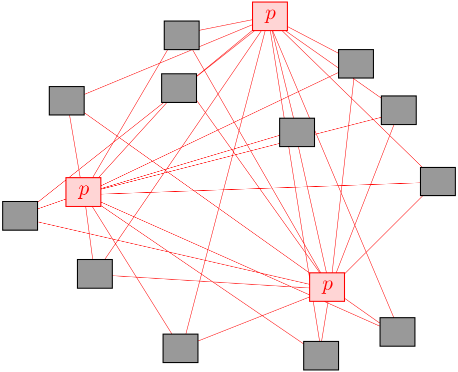
Similarity Search Task

- Index creation:
  - The system selects 4 sets of 9 pivots with the incremental SSS algorithm [Bustos et al, 2008].
    - Each set requires a table with 9 × 4,000,000 distances.
  - The system compares the 4 sets and selects the set that has the greatest average lower bound and discards the others [Zezula et al, 2005].



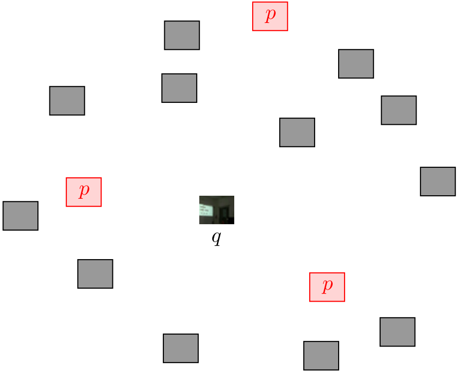
Similarity Search Task

- Index creation:
  - The system selects 4 sets of 9 pivots with the incremental SSS algorithm [Bustos et al, 2008].
    - Each set requires a table with 9 × 4,000,000 distances.
  - The system compares the 4 sets and selects the set that has the greatest average lower bound and discards the others [Zezula et al, 2005].



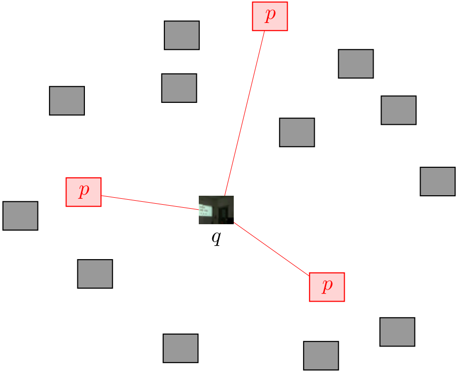
Similarity Search Task

- Similarity search for a query group q:
  - For every pivot p evaluate $DIST(q, p)$.
  - For every reference group v calculate a lower bound for $DIST(q, v)$
    - Only 9 operations to calculate each lower bound.
  - Select 4,000 objects (0.1%) with lowest lower bounds.
  - Calculate actual $DIST(q, v)$ just for the 4,000 objects and select the NNs between them.



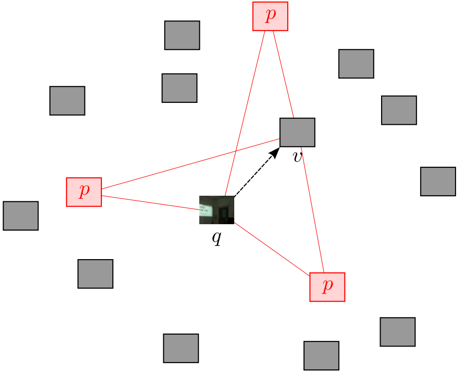
Similarity Search Task

- Similarity search for a query group q:
  - For every pivot p evaluate $DIST(q,p)$.
  - For every reference group v calculate a lower bound for $DIST(q,v)$
    - Only 9 operations to calculate each lower bound.
  - Select 4,000 objects (0.1%) with lowest lower bounds.
  - Calculate actual $DIST(q,v)$ just for the 4,000 objects and select the NNs between them.



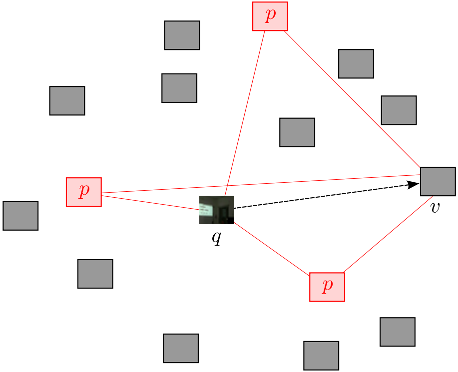
Similarity Search Task

- Similarity search for a query group q:
  - For every pivot p evaluate $DIST(q, p)$.
  - For every reference group v calculate a lower bound for $DIST(q, v)$
    - Only 9 operations to calculate each lower bound.
  - Select 4,000 objects (0.1%) with lowest lower bounds.
  - Calculate actual $DIST(q, v)$ just for the 4,000 objects and select the NNs between them.



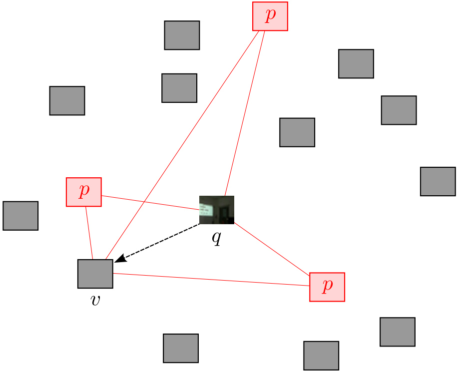
Similarity Search Task

- Similarity search for a query group q:
  - For every pivot p evaluate $DIST(q, p)$.
  - For every reference group v calculate a lower bound for $DIST(q, v)$
    - Only 9 operations to calculate each lower bound.
  - Select 4,000 objects (0.1%) with lowest lower bounds.
  - Calculate actual $DIST(q, v)$ just for the 4,000 objects and select the NNs between them.



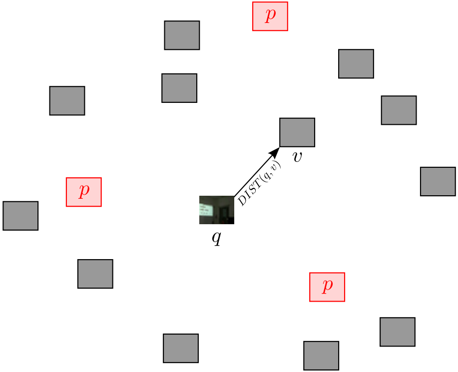
Similarity Search Task

- Similarity search for a query group q:
  - For every pivot p evaluate $DIST(q, p)$.
  - For every reference group v calculate a lower bound for $DIST(q, v)$
    - Only 9 operations to calculate each lower bound.
  - Select 4,000 objects (0.1%) with lowest lower bounds.
  - Calculate actual $DIST(q, v)$ just for the 4,000 objects and select the NNs between them.



Similarity Search Task

- Similarity search for a query group q:
  - For every pivot p evaluate $DIST(q, p)$.
  - For every reference group v calculate a lower bound for $DIST(q, v)$
    - Only 9 operations to calculate each lower bound.
  - Select 4,000 objects (0.1%) with lowest lower bounds.
  - Calculate actual $DIST(q, v)$ just for the 4,000 objects and select the NNs between them.



5. **Copy Localization**:
   - Takes NNs table and searches for chains of groups belonging to a same reference video with temporal coherence.
   - Voting algorithm based on NN rank, NN distance and spread of votes in chain.
   - Copy localization set as start/end of chain.

Query	NN 1	NN 2	NN 3
Query1Group1	Vid07_Grp54 dist	Vid08_Grp73 dist	Vid01_Grp68 dist
Query1Group2	Vid09_Grp13 dist	Vid02_Grp34 dist	Vid02_Grp33 dist
Query1Group3	Vid07_Grp34 dist	Vid03_Grp54 dist	Vid09_Grp14 dist
Query1Group4	Vid09_Grp15 dist	Vid02_Grp13 dist	Vid03_Grp65 dist
Query1Group5	Vid01_Grp88 dist	Vid01_Grp12 dist	Vid07_Grp58 dist
Query1Group6	Vid09_Grp54 dist	Vid09_Grp17 dist	Vid07_Grp59 dist
Query1Group7	Vid01_Grp45 dist	Vid03_Grp43 dist	Vid03_Grp20 dist
Query1Group8	Vid09_Grp19 dist	Vid01_Grp12 dist	Vid07_Grp61 dist
...	...	...	...

5. **Copy Localization**:
   - Takes NNs table and searches for chains of groups belonging to a same reference video with temporal coherence.
   - Voting algorithm based on NN rank, NN distance and spread of votes in chain.
   - Copy localization set as start/end of chain.

Query	NN 1	NN 2	NN 3
Query1Group1	**Vid07_Grp54** dist	Vid08_Grp73 dist	Vid01_Grp68 dist
Query1Group2	Vid09_Grp13 dist	Vid02_Grp34 dist	Vid02_Grp33 dist
Query1Group3	Vid07_Grp34 dist	Vid03_Grp54 dist	Vid09_Grp14 dist
Query1Group4	Vid09_Grp15 dist	Vid02_Grp13 dist	Vid03_Grp65 dist
Query1Group5	Vid01_Grp88 dist	Vid01_Grp12 dist	**Vid07_Grp58** dist
Query1Group6	Vid09_Grp54 dist	Vid09_Grp17 dist	**Vid07_Grp59** dist
Query1Group7	Vid01_Grp45 dist	Vid03_Grp43 dist	Vid03_Grp20 dist
Query1Group8	Vid09_Grp19 dist	Vid01_Grp12 dist	**Vid07_Grp61** dist
...	...	...	...

score Vid07= 2.2

5. **Copy Localization**:
   - Takes NNs table and searches for chains of groups belonging to a same reference video with temporal coherence.
   - Voting algorithm based on NN rank, NN distance and spread of votes in chain.
   - Copy localization set as start/end of chain.

Query	NN 1	NN 2	NN 3
Query1Group1	Vid07_Grp54 dist	Vid08_Grp73 dist	Vid01_Grp68 dist
Query1Group2	**Vid09_Grp13** dist	Vid02_Grp34 dist	Vid02_Grp33 dist
Query1Group3	Vid07_Grp34 dist	Vid03_Grp54 dist	**Vid09_Grp14** dist
Query1Group4	**Vid09_Grp15** dist	Vid02_Grp13 dist	Vid03_Grp65 dist
Query1Group5	Vid01_Grp88 dist	Vid01_Grp12 dist	Vid07_Grp58 dist
Query1Group6	Vid09_Grp54 dist	**Vid09_Grp17** dist	Vid07_Grp59 dist
Query1Group7	Vid01_Grp45 dist	Vid03_Grp43 dist	Vid03_Grp20 dist
Query1Group8	**Vid09_Grp19** dist	Vid01_Grp12 dist	Vid07_Grp61 dist
...	...	...	...

score Vid09= 3.7

score Vid07= 2.2

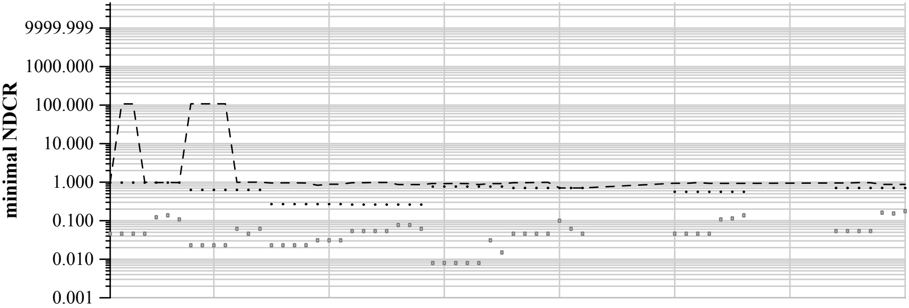
RESULTS

Results

- Submitted Runs:
  - `balanced.ehdNgryhst`: $\delta = 0.068 \times EH + 0.090 \times GH$
  - `balanced.ehdNclrhst`: $\delta = 0.068 \times EH + 0.045 \times CH$
  - `nofa.ehdNgryhst`: equal to `balanced.ehdNgryhst` with stricter voting algorithm.
  - `nofa.ehdNghT10`: equal to `nofa.ehdNgryhst` but with a different threshold.
- Analysis focused on Optimal NDCR.
- EH+GH slightly better than EH+CH.
- Better results in NOFA profile than in Balanced profile.

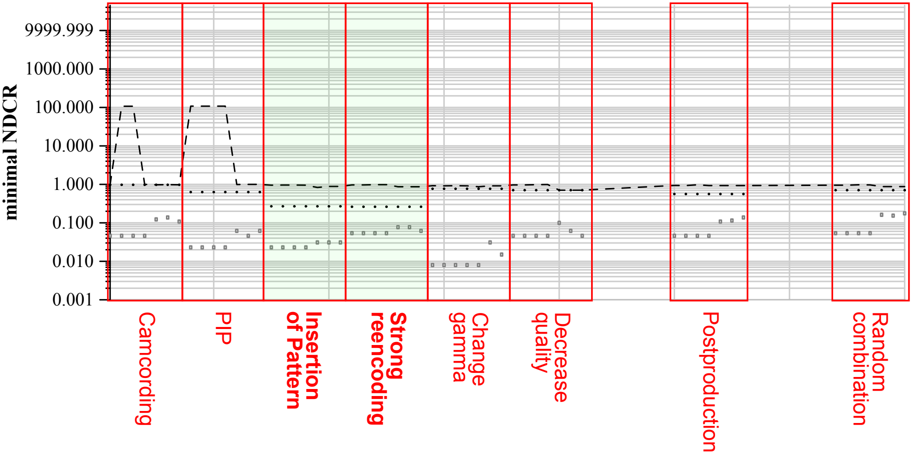
Results nofa.ehdNgryhst

- Optimal NDCR:
  - Lower NDCR than median for each transformation.
  - Better results for Insertion of Pattern and Strong Reencoding.



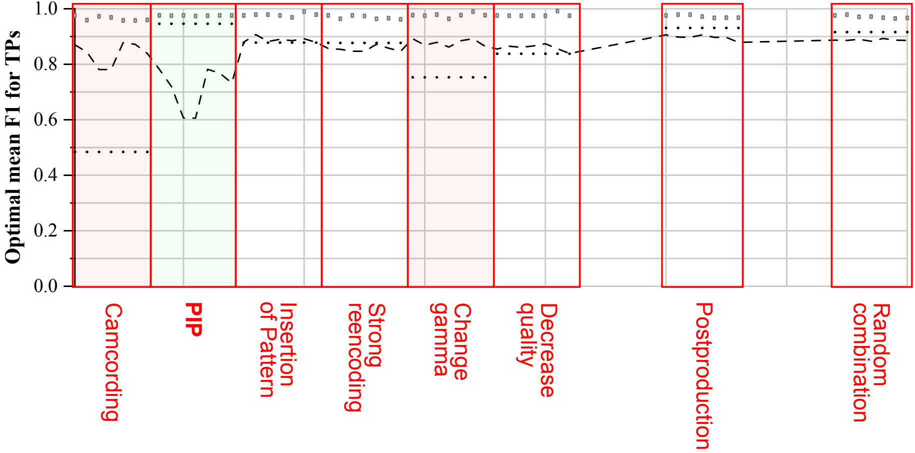
Results nofa.ehdNgryhst

- Optimal NDCR:
  - Lower NDCR than median for each transformation.
  - Better results for Insertion of Pattern and Strong Reencoding.



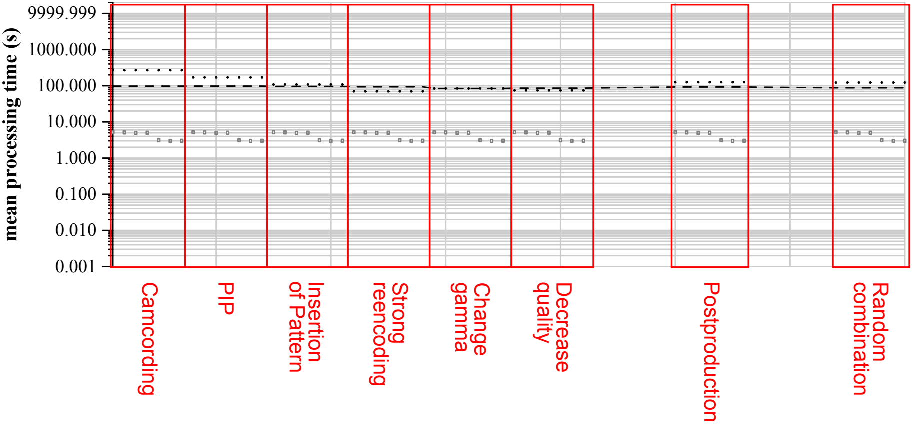
Results nofa.ehdNgryhst

- Optimal F1:
  - Good localization for PIP and bad localization for Camcording and Change in gamma.



Results `nofa.ehdNgryhst`

- Mean Time:
  - Slightly higher than the median, specially for camcording and PIP.



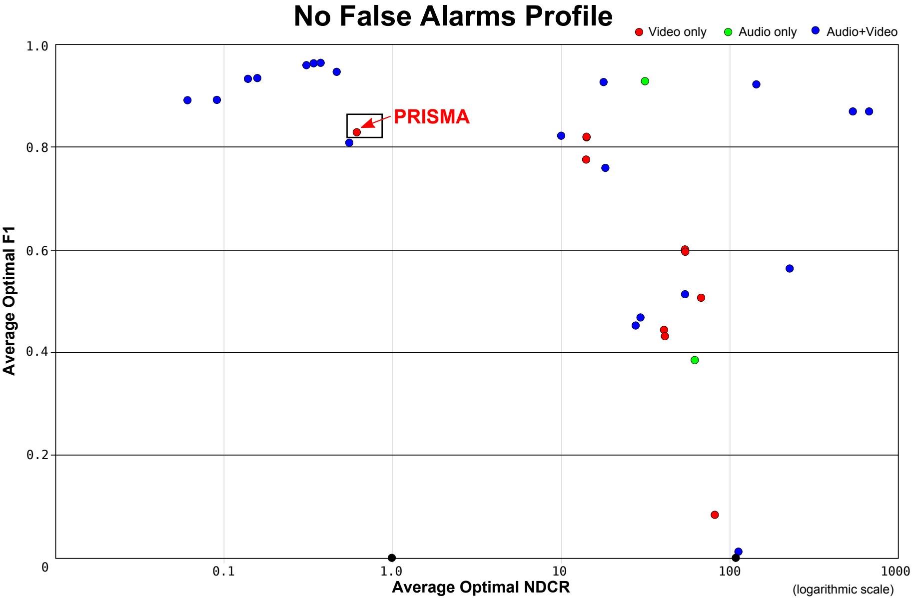
Comparison

- Comparison with Optimal NDCR averaged between all transformations.
- 22 teams, 41 submitted runs for balanced profile and 37 for nofa profile.

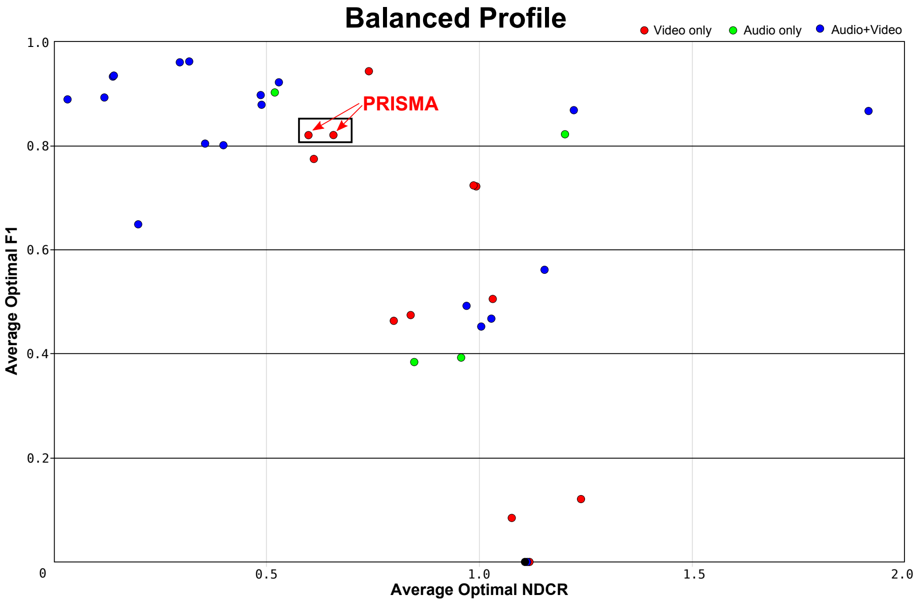
Run	Avg Opt NDCR	global rank	**video-only rank**
`balanced.ehdNgryhst`	0.597	14^{th} of 41	**1^{st} of 15**
`balanced.ehdNclrhst`	0.658	16^{th} of 41	**3^{rd} of 15**
`nofa.ehdNgryhst`	0.611	10^{th} of 37	**1^{st} of 14**
`nofa.ehdNghT10`	0.611	11^{th} of 37	**2^{nd} of 14**

Run	Avg Opt F1	global rank	**video-only rank**
`balanced.ehdNgryhst`	0.820	15^{th} of 41	**2^{nd} of 15**
`balanced.ehdNclrhst`	0.820	16^{th} of 41	**3^{rd} of 15**
`nofa.ehdNgryhst`	0.828	14^{th} of 37	**1^{st} of 14**
`nofa.ehdNghT10`	0.828	15^{th} of 37	**2^{nd} of 14**

Comparison


No False Alarms Profile
Video only
Audio only
Audio+Video
PRISMA
1.0
0.8
0.6
0.4
0.2
0
0.1
1.0
10
100
1000
Average Optimal F1
Average Optimal NDCR
(logarithmic scale)

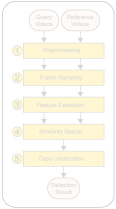
Comparison

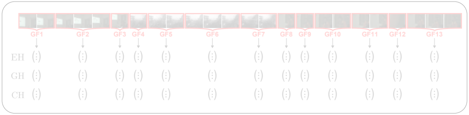

Balanced Profile
Video only
Audio only
Audio+Video
PRISMA
Average Optimal F1
Average Optimal NDCR
1.0
0.8
0.6
0.4
0.2
0
0.5
1.0
1.5
2.0

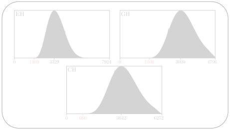
Conclusions

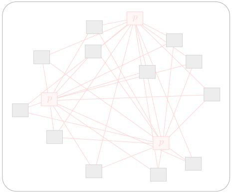
- Acceptable overall results:
  - Global descriptors can achieve competitive results with TRECVID transformations.
  - Pivot-based approximation enables to discard 99.9% of distance computations and still have good effectiveness.
- Two novel techniques:
  - Set weights maximizing intrinsic dimensionality.
  - Calculate actual distance just for 0.1% lowest lower bounds.
- Future work:
  - Improve the efficiency of preprocessing task.
  - Test other distances for descriptors instead of L_1 (in particular some non-metric similarity measure).
  - Test the inclusion of audio information and local descriptors.

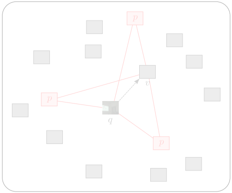
Thank you!

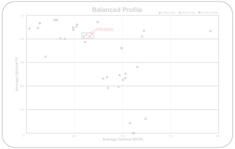


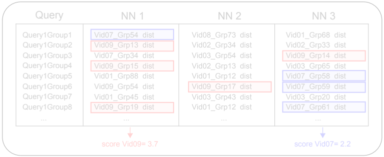












Juan Manuel Barrios and Benjamin Bustos

PRISMA Research Group
Department of Computer Science
University of Chile
{jbarrios,bebustos}@dcc.uchile.cl

November 17, 2010

Thank you!