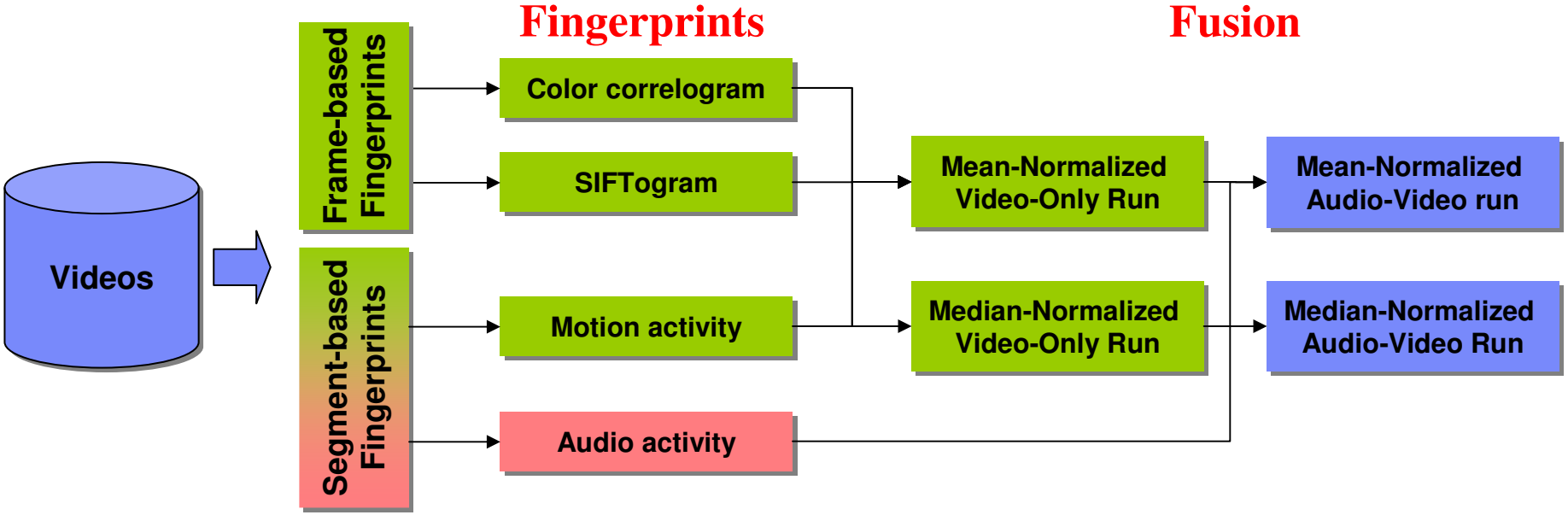
IBM Content Based Copy Detection System for TRECVID 2009

Speaker: Matt Hill

On behalf of:
Jane Chang, Michele Merler, Paul Natsev, John R. Smith

System Overview

- We explored 4 complementary approaches for video fingerprinting:
  - Two frame-based visual fingerprints (color correlogram and SIFTogram)
  - Two temporal sequence-based fingerprints (audio & motion activity)



- Key question: How far can we go with coarse-grain fingerprints?
  - Focus on common real-world transforms typical for video piracy detection
  - Focus on speed, space efficiency, lack of false alarms

We focused on CBCD transforms that represent typical video piracy scenarios (i.e., ignore PIP and post-production edits)

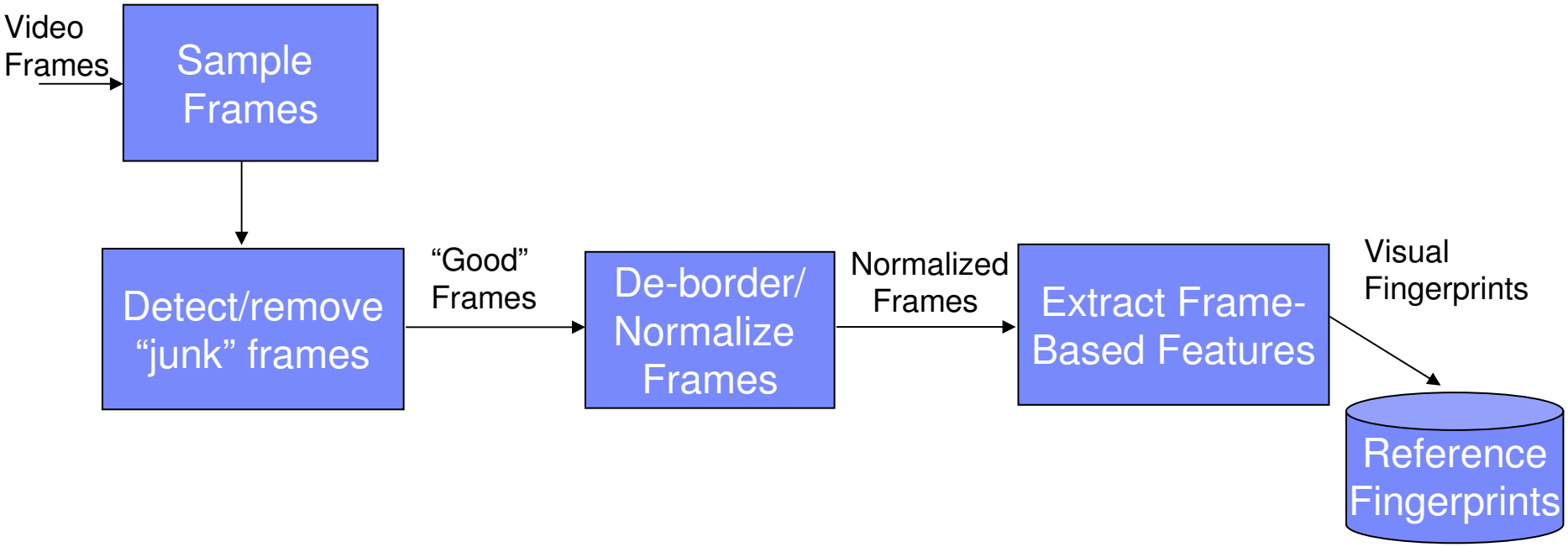
- T2: Picture in picture Type 1 (The original video is inserted in front)
- T3: Insertions of pattern
- T4: Strong re-encoding
- T5: Change of gamma
- T6: Decrease in quality -- This includes choosing randomly 3 transformations from the following: Blur, change of gamma, frame dropping, contrast, compression, ratio, white noise

We focused on these 4 transforms

- T8: Post production -- This includes choosing randomly 3 transformations from the following: Crop, Shift, Contrast, caption (text insertion), flip (mirroring), Insertion of pattern, Picture in Picture type 2 (the original video is in the background)
- T10: change to randomly choose 1 transformation from each of the 3 main categories.

Visual Fingerprint Extraction for Frame-Based Methods

- Sample 1 frame per second for visual feature extraction
- Throw out bad frames, normalize appearance of remaining frames
- Extract the relevant feature, i.e. color correlogram or SIFTogram
- Add reference content to the database for indexing



Color Correlogram-based Fingerprints

- A color correlogram expresses how the spatial correlation of colors changes within a local region neighborhood
  - Captures color and local structure, some invariability to view point changes
  - We use a “cross” formulation which also captures global layout & emphasizes the center of the image, while being invariant to flips
- Informally, a correlogram for an image is a table indexed by color pairs, where the *d*-th entry for row *(i,j)* specifies the probability of finding a pixel of color *j* at a distance *d* from a pixel of color *i* in this image
  - We use simplified auto-correlogram formulation, which captures conditional probability of seeing given color within a certain distance of same color pixel
- We compute the auto-correlogram in a 166-dimensional quantized HSV color space, resulting in a 332-dimensional cross-CC feature vector
- Pros/cons for correlogram fingerprints:
  - Robust w.r.t. brightness changes, aspect ratio, small crops, flipping, compression
  - Cons: non-linear intensity transforms (e.g., gamma), changes in hue, saturation

Visual Word-based Fingerprints (SIFTograms)

- Histogram of SIFT-based Codewords
  - We use U. of Amsterdam's tools to detect interest points and extract SIFT descriptors
  - We build a codebook of visual words using k-means clustering to quantize the SIFT features
    - Harris-Laplace, SIFT descriptor, soft assignment
  - We then compute a histogram of the quantized SIFT features (SIFTogram), making a global feature for each frame sampled at 1fps
  - The # of codewords is the dimensionality of the feature vector, in our case, 1000
- "**SIFTogram**" is robust w.r.t. gamma, color, rotation, scale, blur, borders and some overlaid graphics
- Cons: compute intensive, space inefficient, does not handle compression well

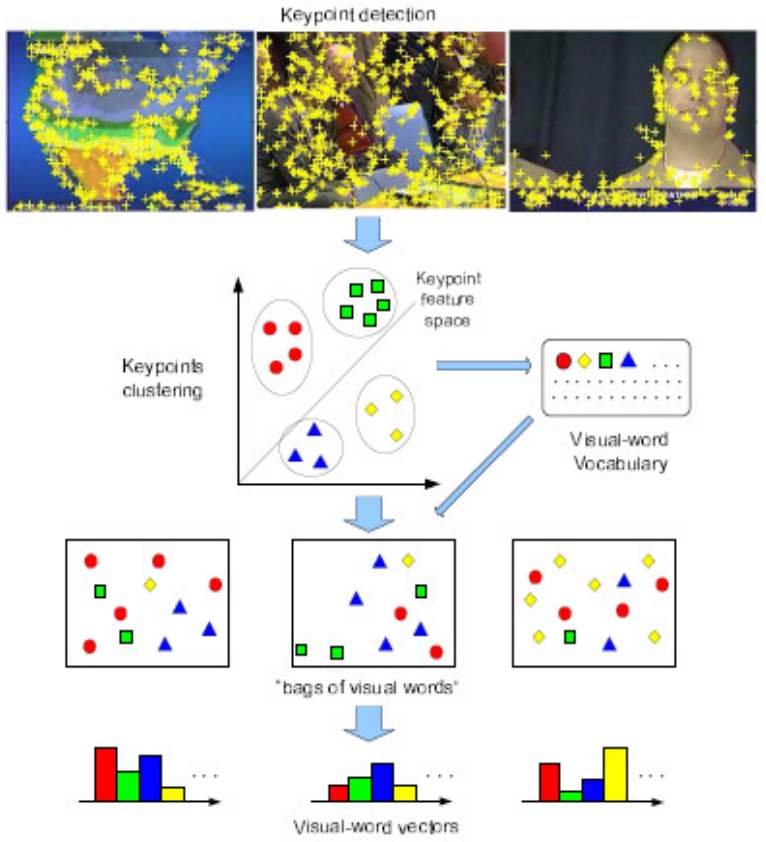
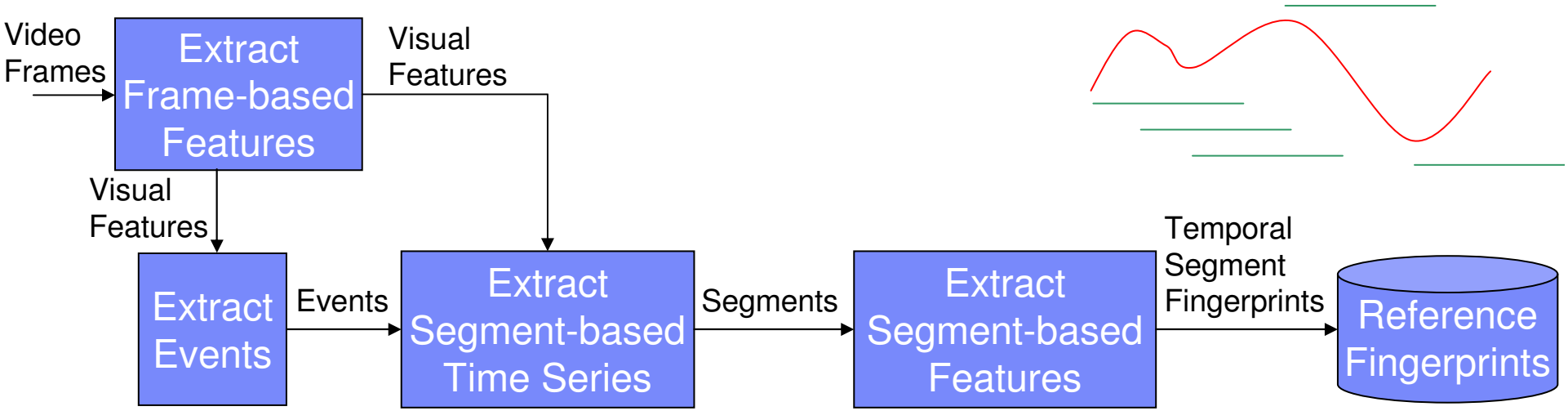


Figure: C.W. Ngo

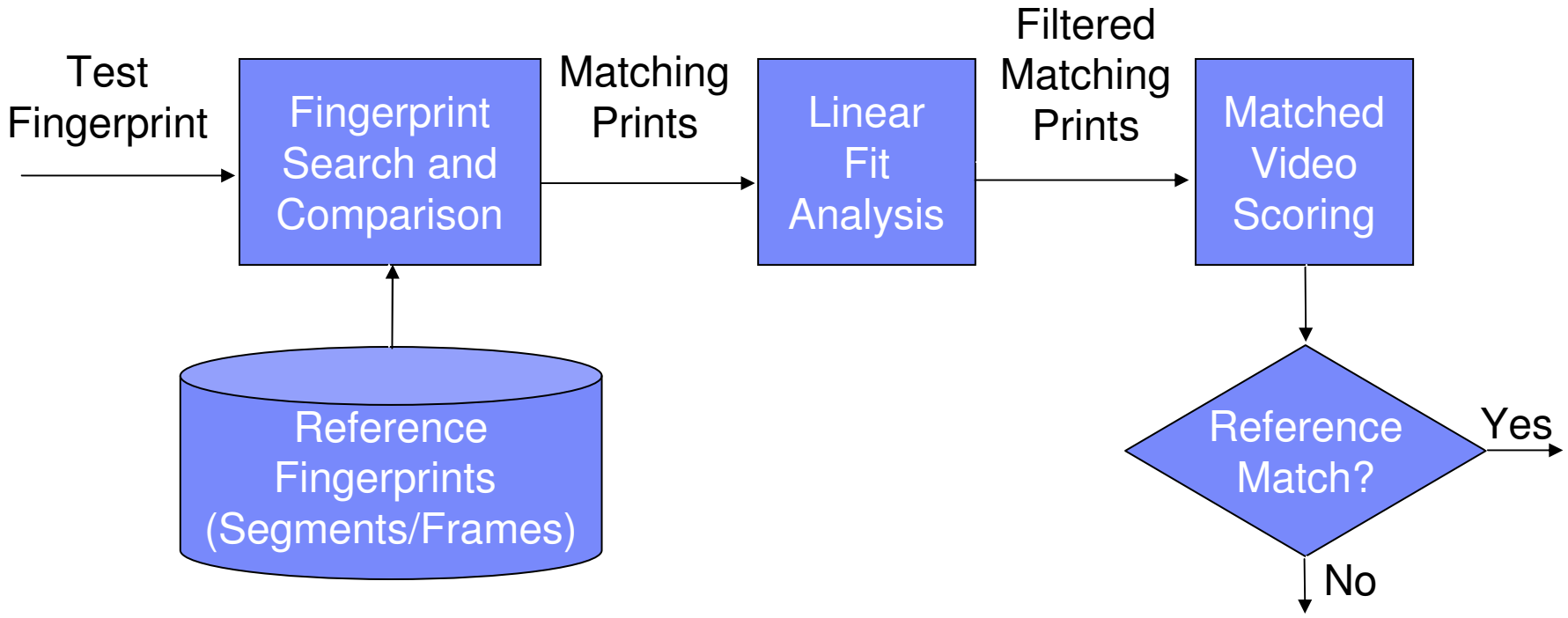
Temporal Fingerprint Extraction for Segment-Based Methods

- We apply this method to describe overall audio or motion activity
- We scan the audio/video as a time series of audio/visual features and detect “interesting points” along the feature trajectory (e.g., valleys, peaks, flat regions)
- We form overlapping segments covering multiple “events” on the trajectory, normalize the segments, and represent each with a compact fixed dimensionality descriptor based on uniform re-sampling of the segment (64-bytes)
- This process results in many overlapping fingerprint sequences of varying lengths, based on min/max segment duration constraints
- Robust w.r.t. color transforms, blur, noise, compression and geometric transforms
- Doesn’t works well for short copied segments, or segments with little activity



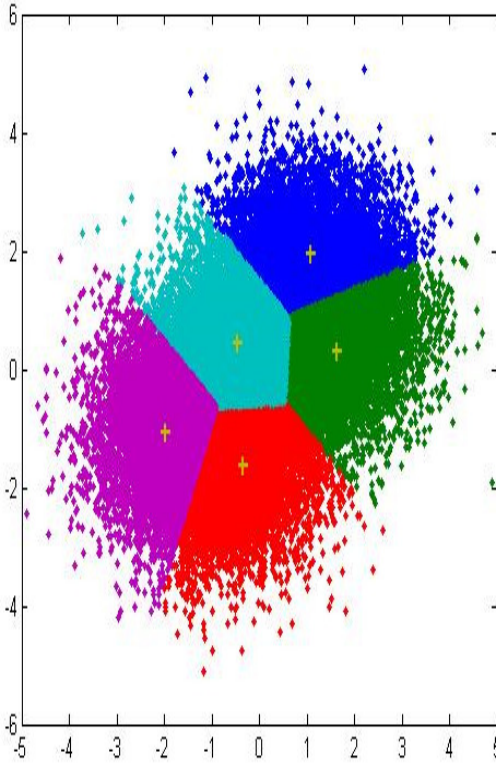
Fingerprint Matching

- For each test segment, find matching reference segments / frames
- For each reference video, collect all matching segments / frames and find the subset of matching segments that produces the best linear fit
- For each reference video, compute an overall matching score based on the matched segments / frames conforming with the computed linear fit params
- Determine copy / no copy status based on overall score threshold



Indexing for Fast Nearest Neighbor Lookup

- We use FLANN (Fast Library for Approximate Nearest Neighbor) open source library to enable fast lookups of the fingerprints
  - Authors: Marius Muja and David G. Lowe, Univ. of British Columbia
  - http://www.cs.ubc.ca/~mariusm/index.php/FLANN/FLANN
- Given the set to index, FLANN can auto-selec algorithms (kd-tree, hierarchical k-means, hybrid) and parameters
- Speed gains of 50x compared to linear scan with color feature method enabled us to tune matching params for better performance
- SIFTogram lookup relies on indexing even more

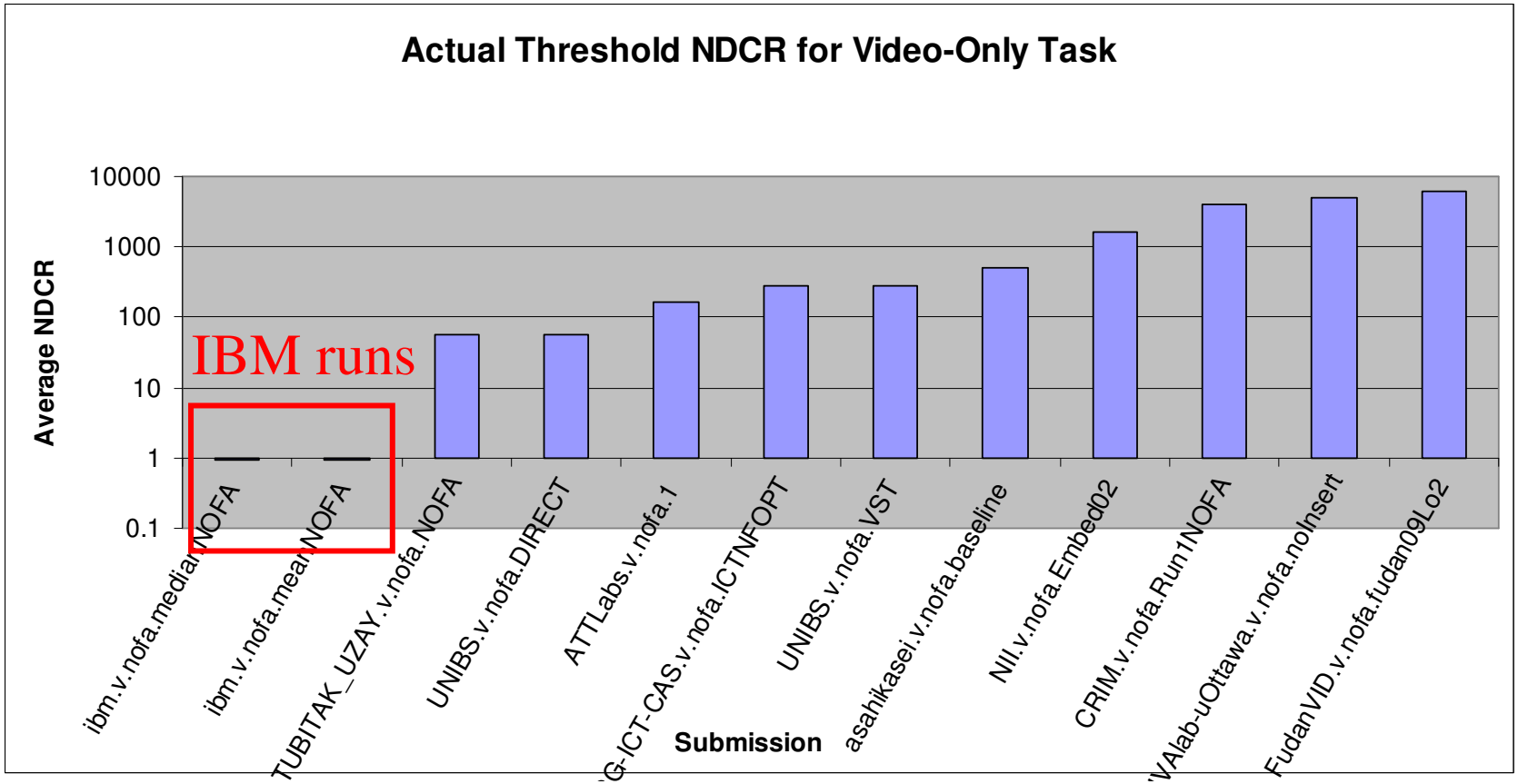


Performance Analysis

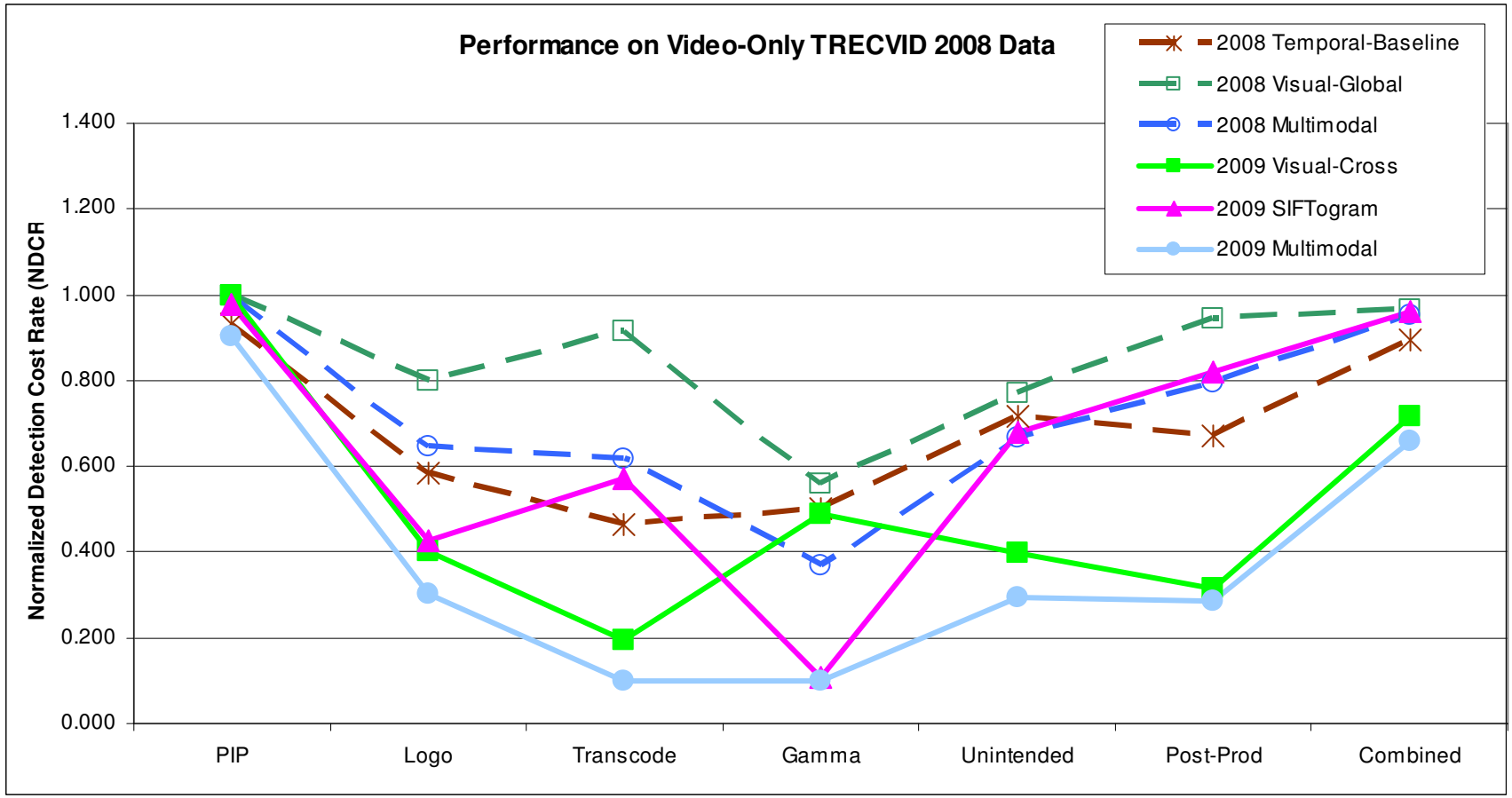
- We use NOFA profile as BALANCED profile turns out to be very similar:
  - NDCR = $P_{miss} + \beta \cdot R_{FA}$ *where* $\beta = C_{FA} / (C_{miss} \cdot R_{target}) = 2$ for BALANCED profile
  - $R_{FA} = FP / T_{queries}$ where $T_{queries} \approx 7.3$ hours for the 2009 dataset (201 queries)
  - Therefore, NDCR ≈ $P_{miss} + 0.28\ FP$, or each false alarm increases NDCR by 0.28!
  - Note that we can obtain trivial NDCR = 1.0 by submitting empty result set
  - Therefore, BALANCED profile is essentially a "3-false-alarm profile"
- Our performance analysis is focused on:
  - NOFA profile
  - Optimal NDCR rather than actual NDCR (since most runs had actual NDCR>1)
  - Transforms T3-T6 (typical for video piracy, esp. T6)
  - In some cases, we report aggregate performance over multiple transforms
    - To compute meaningful optimal NDCR scores when aggregating across transforms, we modify the ground truth to map multiple transforms to a single virtual transform
    - This forces evaluation script to use the same optimal threshold across all transforms

Why we use optimal NDCR rather than actual NDCR?

- NOFA penalty resulted in very high costs on actual threshold measure
  - "balanced" profile also allows very few false alarms
- Ours are the only runs with scores less than the trivial NDCR score of 1!

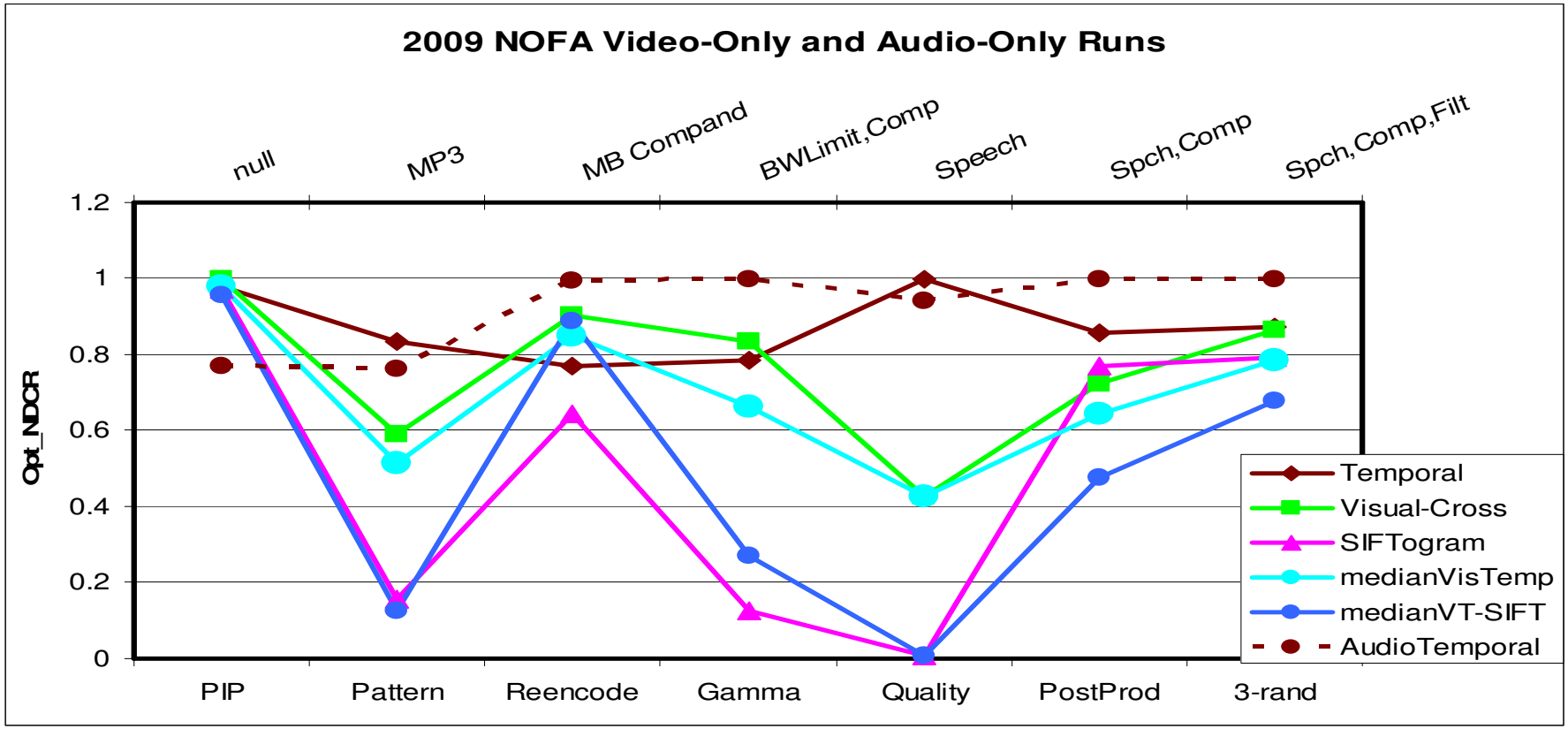


Comparison of Fingerprinting Approaches on CBCD 2008 Data



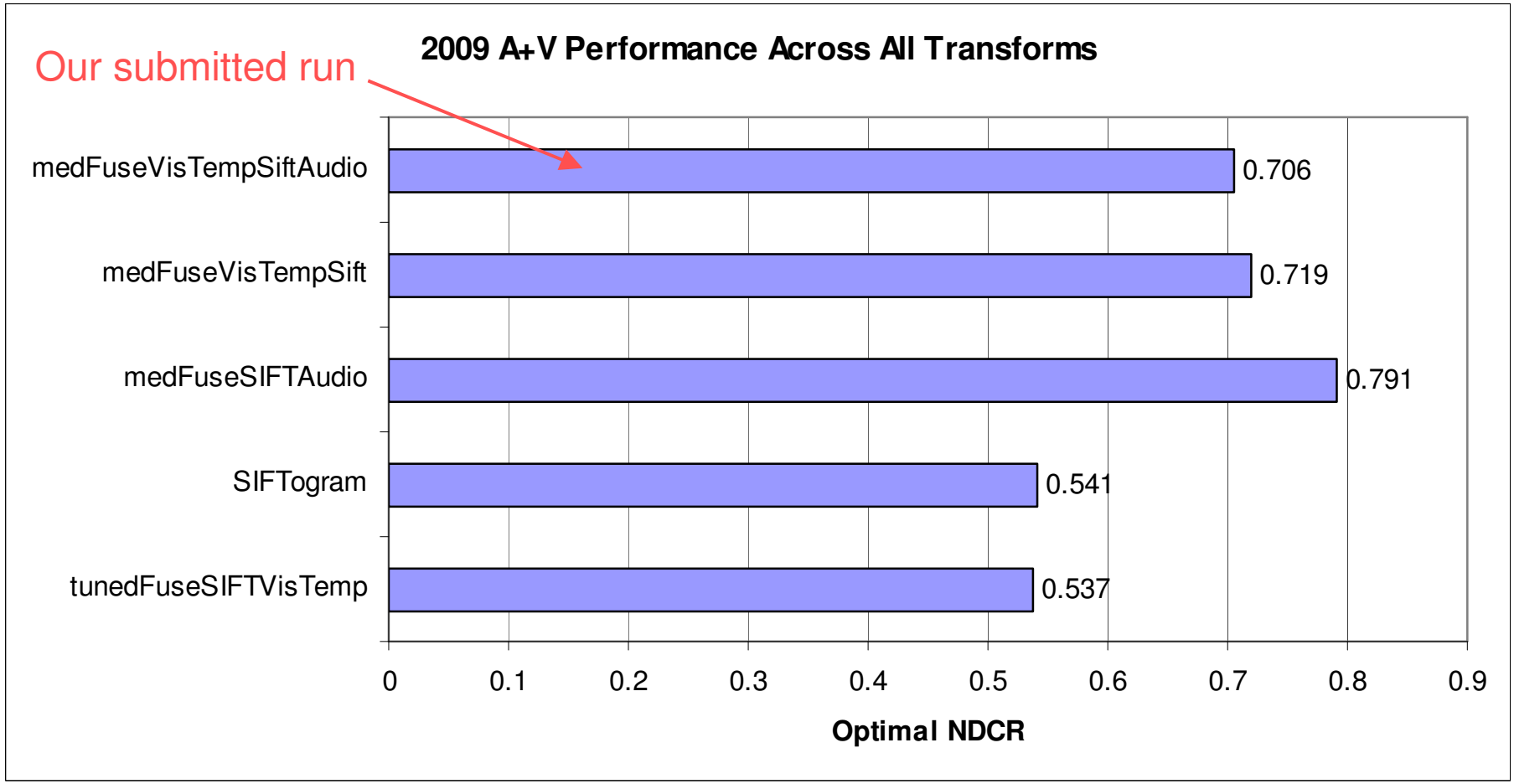
- Multimodal fusion approach consistently outperforms all constituent runs
- 2009 approaches dramatically improve over 2008 runs (2-3x improvement)

Component Runs Compared with Fused Runs on 2009 Data



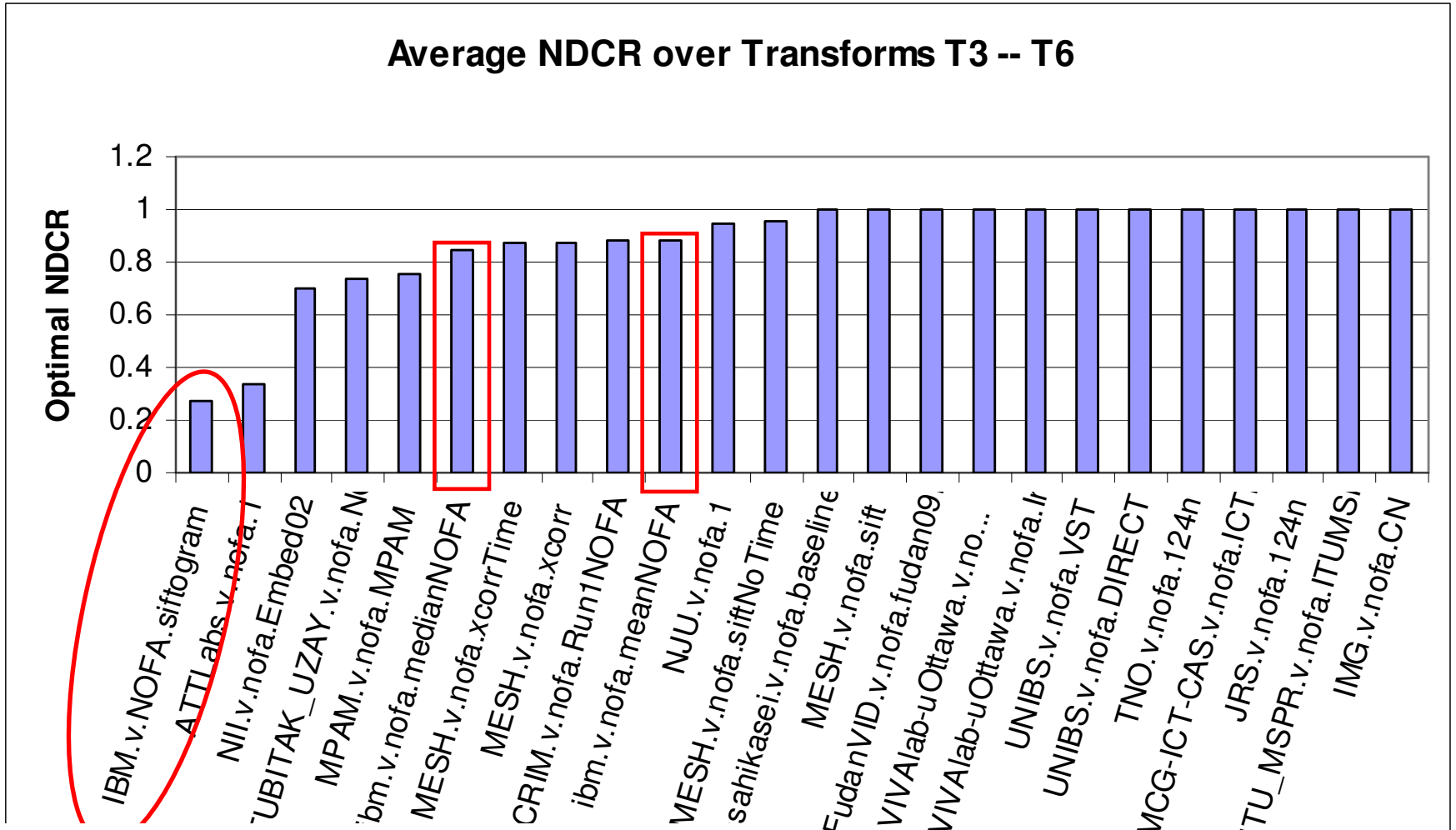
- Performance on re-encoding worse than on 2008 data, all other transforms improve
- SIFTogram performs much better on 2009 than 2008, outperforms all else
- Fusion did not generalize (likely due to SIFTogram performance change)
- Overall, excellent performance on 3 of 4 target transforms

For 2009, SIFTogram outperformed our fusion run on A+V task



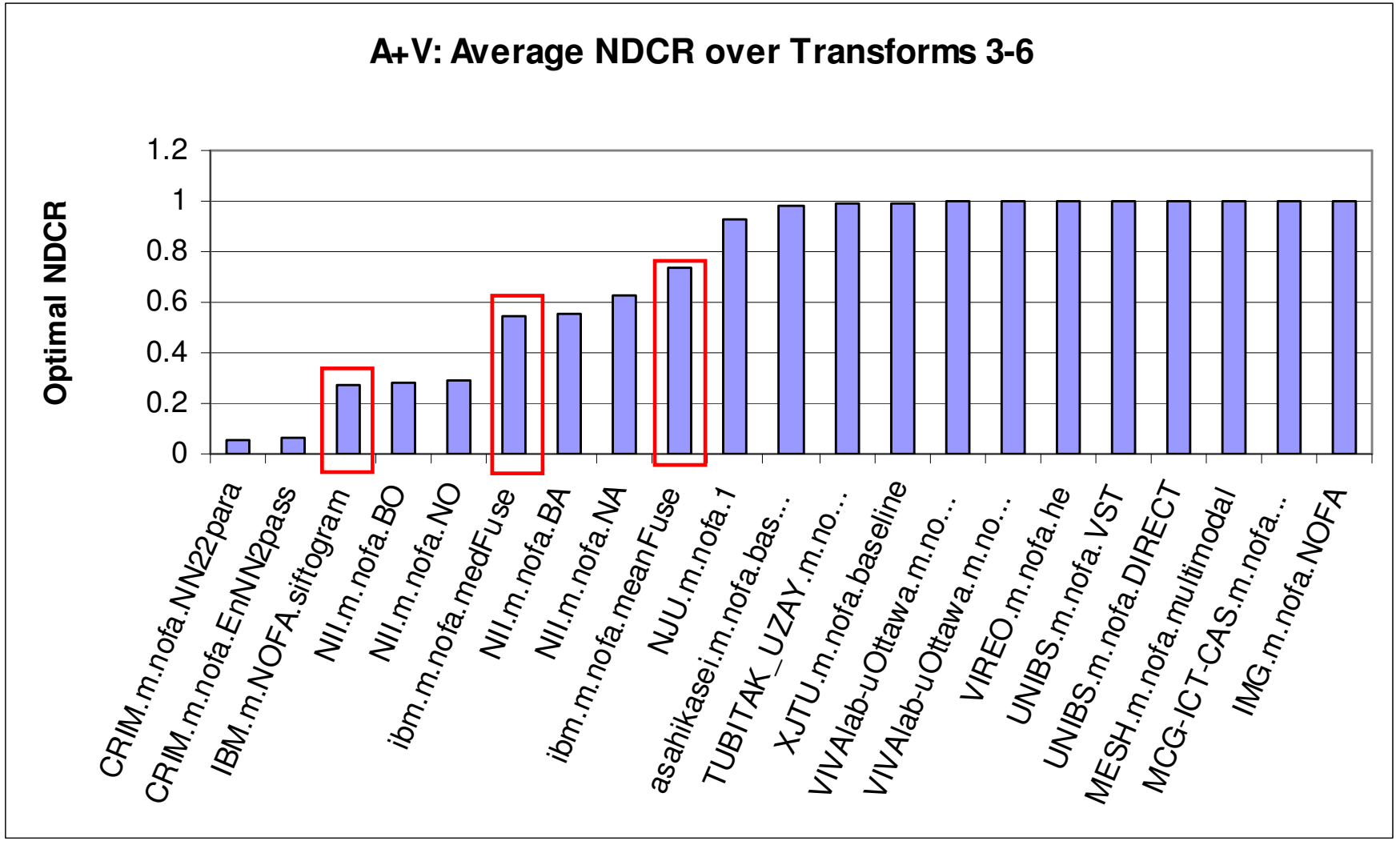
"Tuned" fusion, with knowledge of results, only slightly improves on our SIFTogram

Aggregated Performance on T3-T6 Target Transforms for Video-Only Task



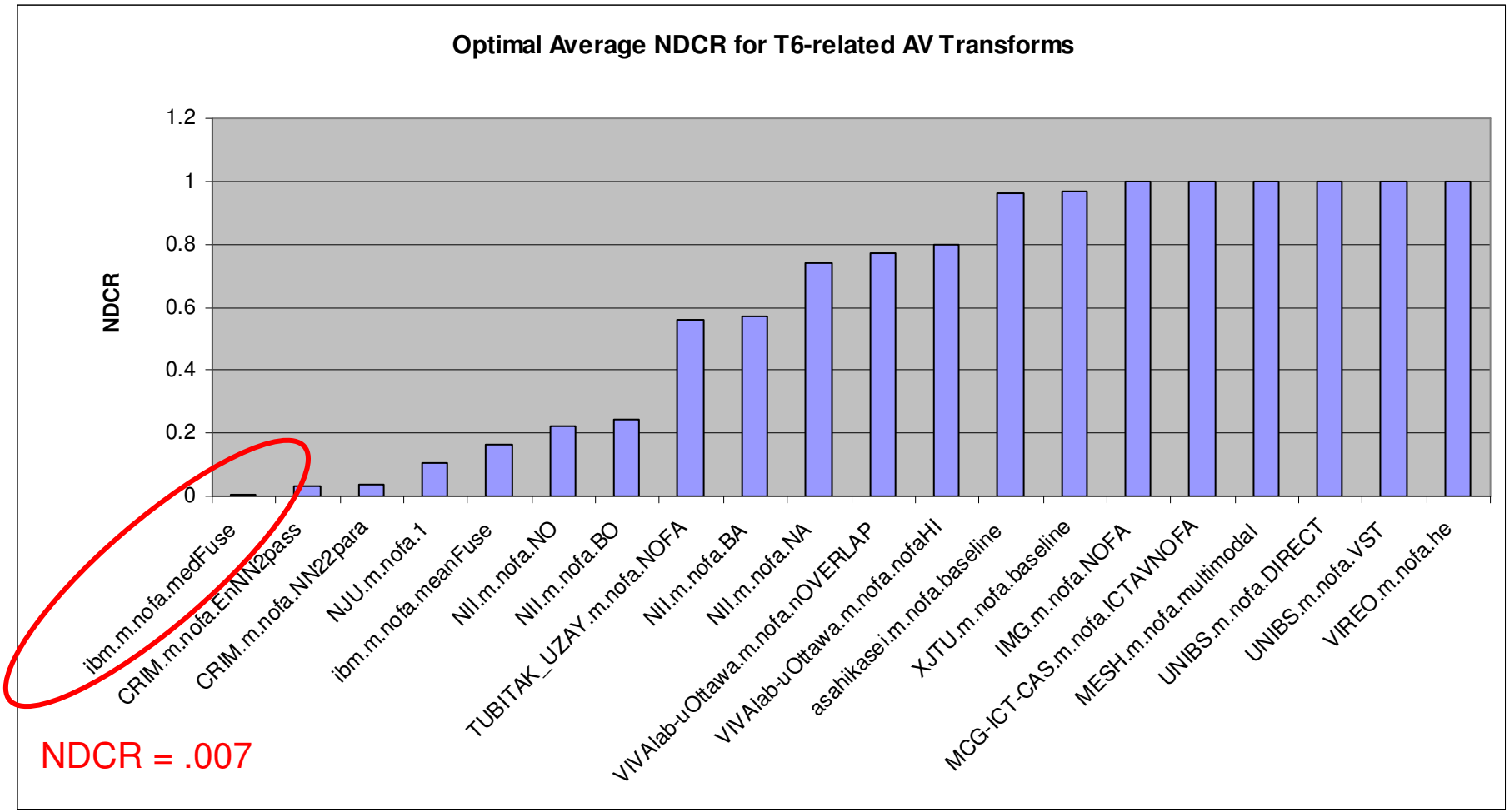
An "unofficial run" – 0.274 NDCR

Results for A+V Task on IBM's Targeted Transforms T3-T6

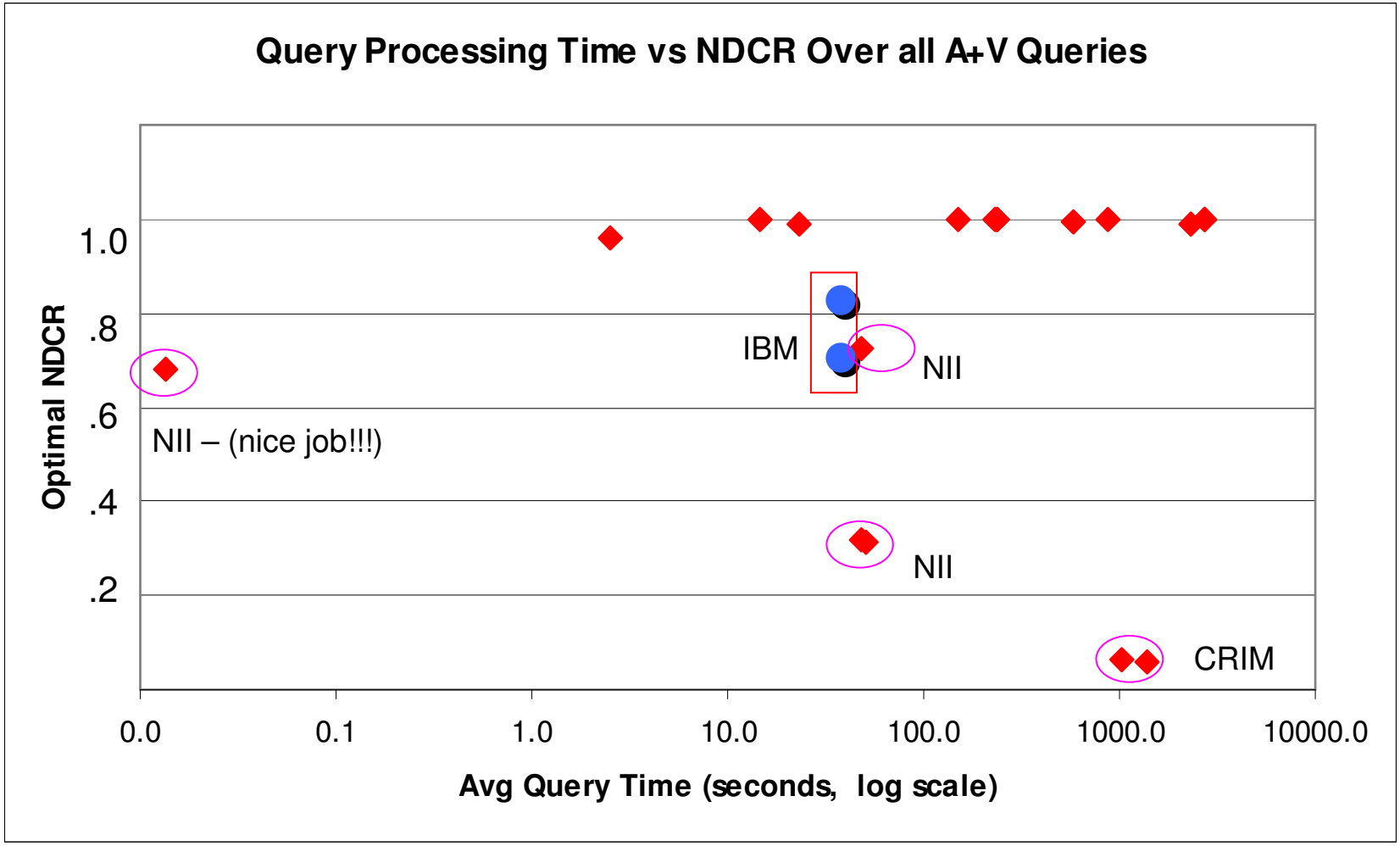


IBM had the best performance on T6 in the A+V task

Each T6 query had 3 of the following types of transforms: blur, change of gamma, frame dropping, contrast, compression, aspect ratio, white noise



Our Solution Provides a Good Trade-off Between Speed and Accuracy



Conclusions

- Coarse-grain fingerprinting methods provide timely and highly accurate results on transforms commonly seen “in the wild”
  - Perfect detection with 0 false alarms on most typical transforms (e.g., T6)
  - Good trade-off between speed, storage, and accuracy
- Fusion methods that worked well on the 2008 test set did not transfer directly to 2009 data
  - “Past results not necessarily an indicator of future performance”
  - Need to consider early fusion methods
- It’s difficult to pick operating thresholds
  - In deployment, they may have to be adjusted online, “in-situ”
- Using a toolbox of independent methods can be parallelized, but combining results for optimal detection is non-trivial