Story Tracking in Video News Broadcasts

Ph.D. Dissertation

Jedrzej Miadowicz

June 4, 2004

Acknowledgements

Motivation

- Modern world is awash in information
  - Coming from multiple sources
  - Around the clock
  - Lately much of the information is delivered visually by means of video
- Usefulness of this information is limited by the lack of adequate means of accessing it
- Particularly in video news
  - Numerous television stations broadcast continuously
  - Much of the news is irrelevant the viewer
  - In order to see everything that is interesting he or she would need to view the entire broadcast

Problem

- Lack of adequate methods of accessing video content
- Video Information Retrieval
  - Is the broad research addressing this problem
  - Provide users with effective and intuitive access to video content relevant to their information needs
- **Story Tracking in Video News Broadcasts**
  - Is one of the main tasks of Video Information Retrieval
  - Consists in detecting and reporting to the user portions of the news broadcast relevant to the news story the user is interested in
  - This work addresses the problem of story tracking in video news broadcasts

Proposed Solution

- Observation
  - News stations reuse video footage in order to provide visual clues for the viewers.
- Thesis
  - Accurate detection of repeated video footage can be used to effectively track stories in live video news broadcasts.

Presentation Outline

- Story tracking stages
  - Temporal Video Segmentation
  - Repeated Video Sequence Detection
  - Story tracking
- Conclusions
- Future Work
- Questions and Discussion

Temporal Video Segmentation

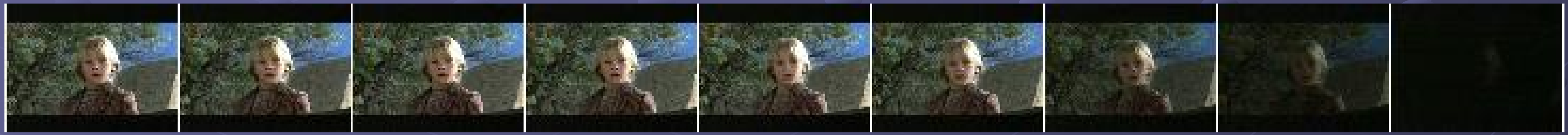
Problem Definition

- Recover the basic structure of video
  - Detect Shots and Transitions
- Shot
  - Sequence of consecutive frames
  - Single camera working continuously
- Transition
  - Sequence of frames combining two shots
  - Wide variety of transition effects are used (cuts, fades, dissolves, wipes, etc.)

Transition Examples

Cut

Fade-out



Dissolve

Temporal Segmentation for Story Tracking

- Effective story tracking
  - Requires accurate identification of **short shots**
    - Repeated video clips are often only a few seconds in length
  - Emphasizes **accurate dissolve detection**
    - Repeated shots are frequently introduced using dissolves
- Additional Challenges
  - On-screen captions
  - Picture-in-picture

Principles of Transition Detection

- Observation
  - Frame content changes radically during transition
- Detect changes in frame content
  - Compare pixels
    - Sensitive to Noise
    - Computationally intensive
  - Compare image features
    - Reflect changes in image content
    - Address the problems above
    - Variety of features available
      - Color histogram, Texture, Motion, **Color Moments**

Related Work

- Research in Temporal Segmentation is well established
  - Different image features have been used to detect cuts
    - Gargi, Lienhart, Truong use intensity histogram,
    - Luptani, Shahraray use inter-frame motion,
    - Zabih utilizes edge pixels.
  - Image variance characteristics have been employed in fade and dissolve detection by Lienhart, Alattar, and Truong.
  - Zabih proposed gradual edge strength changes for recognition of fades and dissolves.
  - Lienhart introduced a neural network pattern recognition method
    - Good performance, but very slow
- Best results reported by Truong

Color Moments

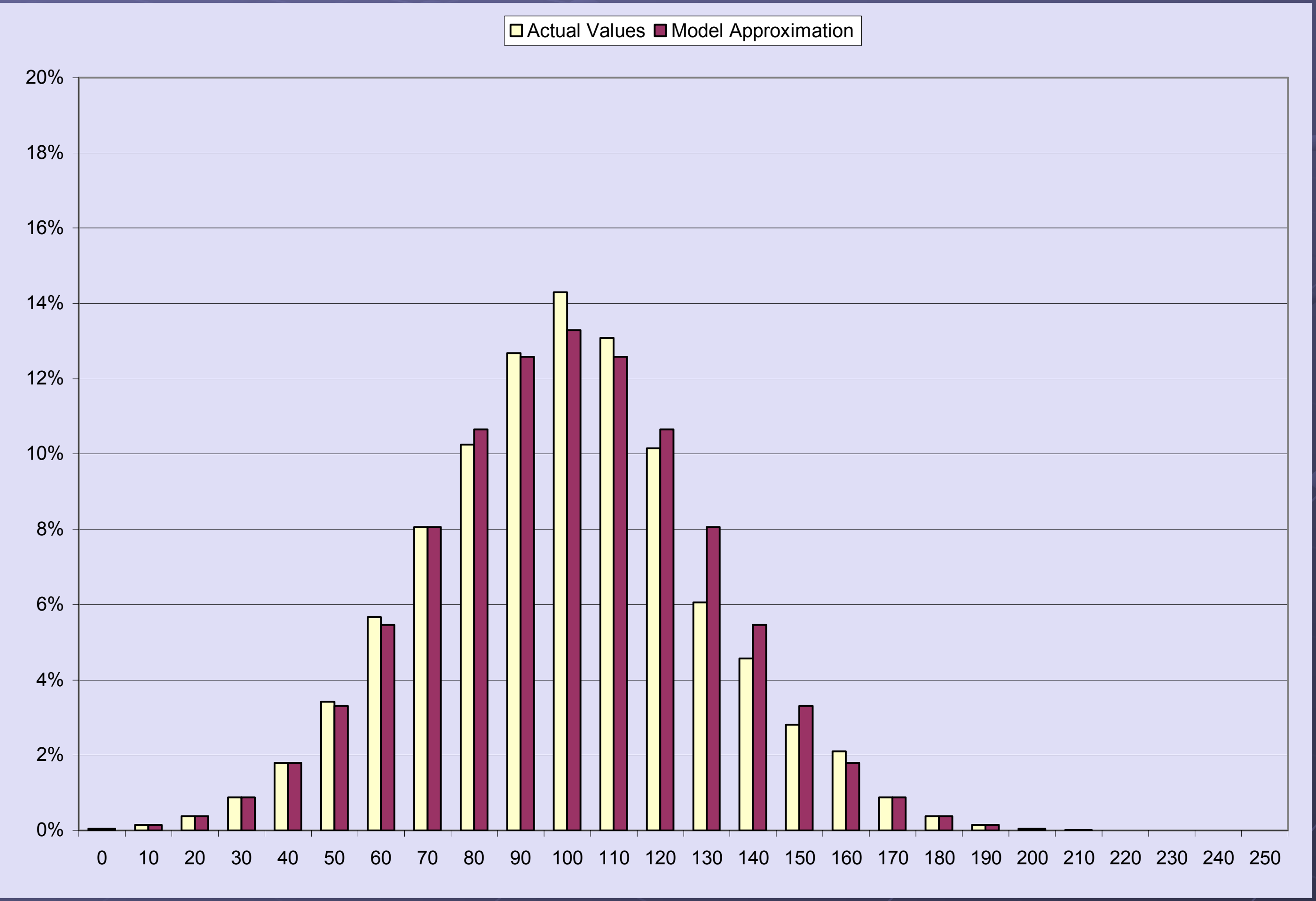
- In this work we use first three moments of the basic image components: red, green, and blue
  - Mean M(t,c)
  - Standard Deviation S(t,c)
  - Skew K(t,c)

$$M(t,c) = \frac{1}{N}\sum_{xy} I(x,y,t,c)$$

$$S(t,c)^2 = \frac{1}{N}\sum_{xy} \left[I(x,y,t,c) - M(t,c)\right]^2$$

$$K(t,c)^3 = \frac{1}{N}\sum_{xy} \left[I(x,y,t,c) - M(t,c)\right]^3$$

Color Moment as Histogram Approximation



Our Approaches to Temporal Segmentation

- Basic Algorithm
  - Analyzes color moment differences (*cross-difference*) over a certain window of frames
  - Detects transitions if the difference exceeds a predetermined threshold
- Transition Model Pattern Detection
  - Identifies *patterns in color moment time series* which are typical of individual transition types

Cross-Difference Algorithm

- Cross-Difference

$$CrossDiff = \sum_{i=t-w}^{t+w} \sum_{j=i+1}^{t+w} a_{ij} d_{ij} \quad where \quad a_{ij} = \begin{cases} 1 & if\ i < t\ or\ j \geq t \\ -1 & otherwise \end{cases}$$

  - d_{ij} is the average color moment difference between frames *i* and *j*
  - *t* is the frame at which transition potentially occurred
  - *w* is a predefined size of a frame window
- Fast and simple
- Inadequate performance
  - Differences in moments may result from motion
  - The algorithm is unable to distinguish well between effects of motion and gradual transitions

Mathematical Models of Transition Effects

- Cut
  - Direct concatenation of two shots not involving any transitional frames, and so the transition sequence is empty
- Fade
  - is a sequence of frames *I(x, y, c, t)* of duration *T* resulting from scaling pixel intensities of the sequence $I_1(x, y, c, t)$ by a temporally monotone function *f(t)*

$$I(x,y,c,t) = f(t)\cdot I_1(x,y,c,t), \quad t \in [0,T]$$

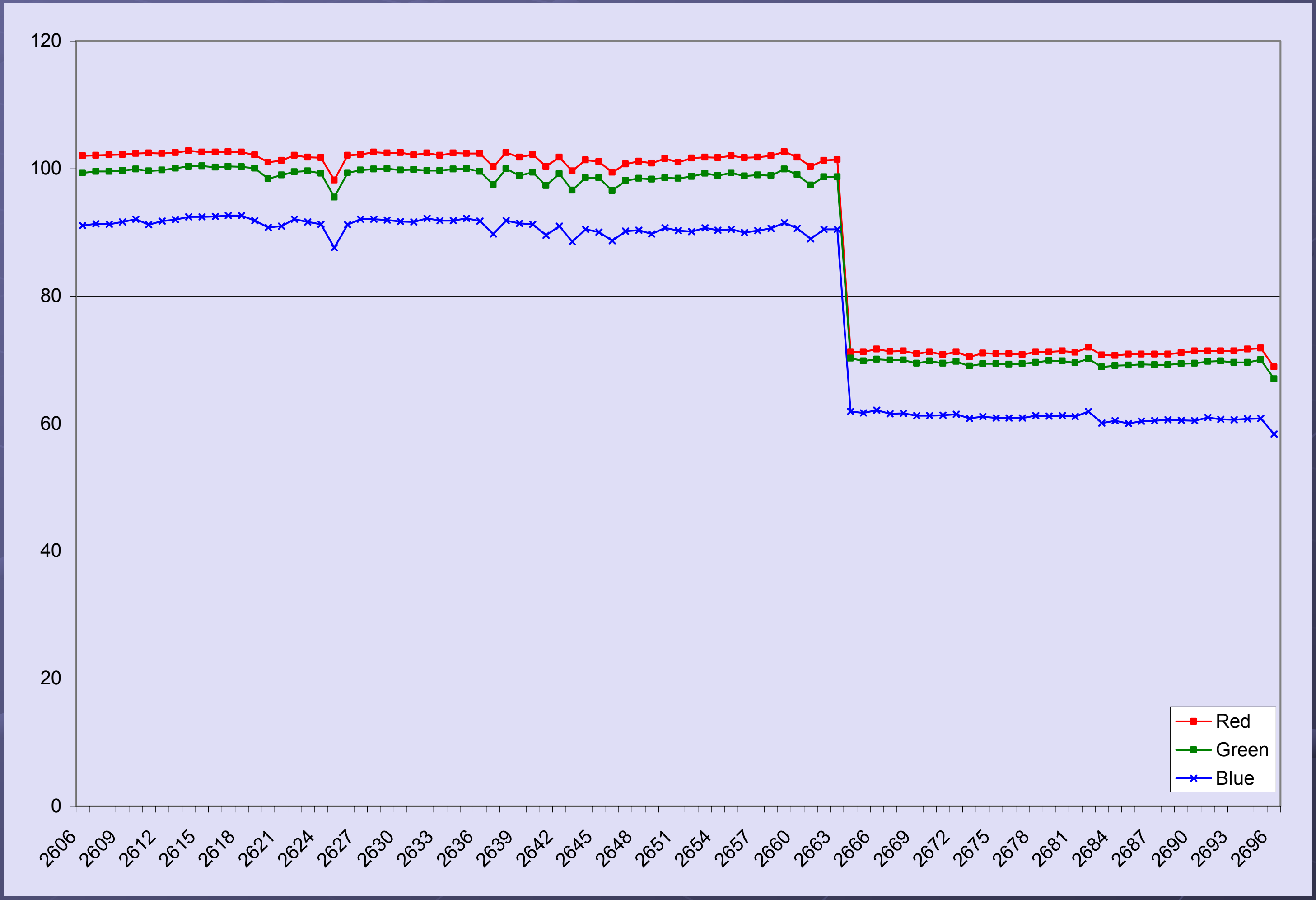
- Dissolve
  - is a sequence *I(x, y, c, t)* of duration *T* resulting from combining two video sequences $I_1(x, y, c, t)$ and $I_2(x, y, c, t)$, where the first sequence is fading out while the second is fading in

$$I(x,y,c,t) = f_1(t)\cdot I_1(x,y,c,t) + f_2(t)\cdot I_2(x,y,c,t), \quad t \in [0,T]$$

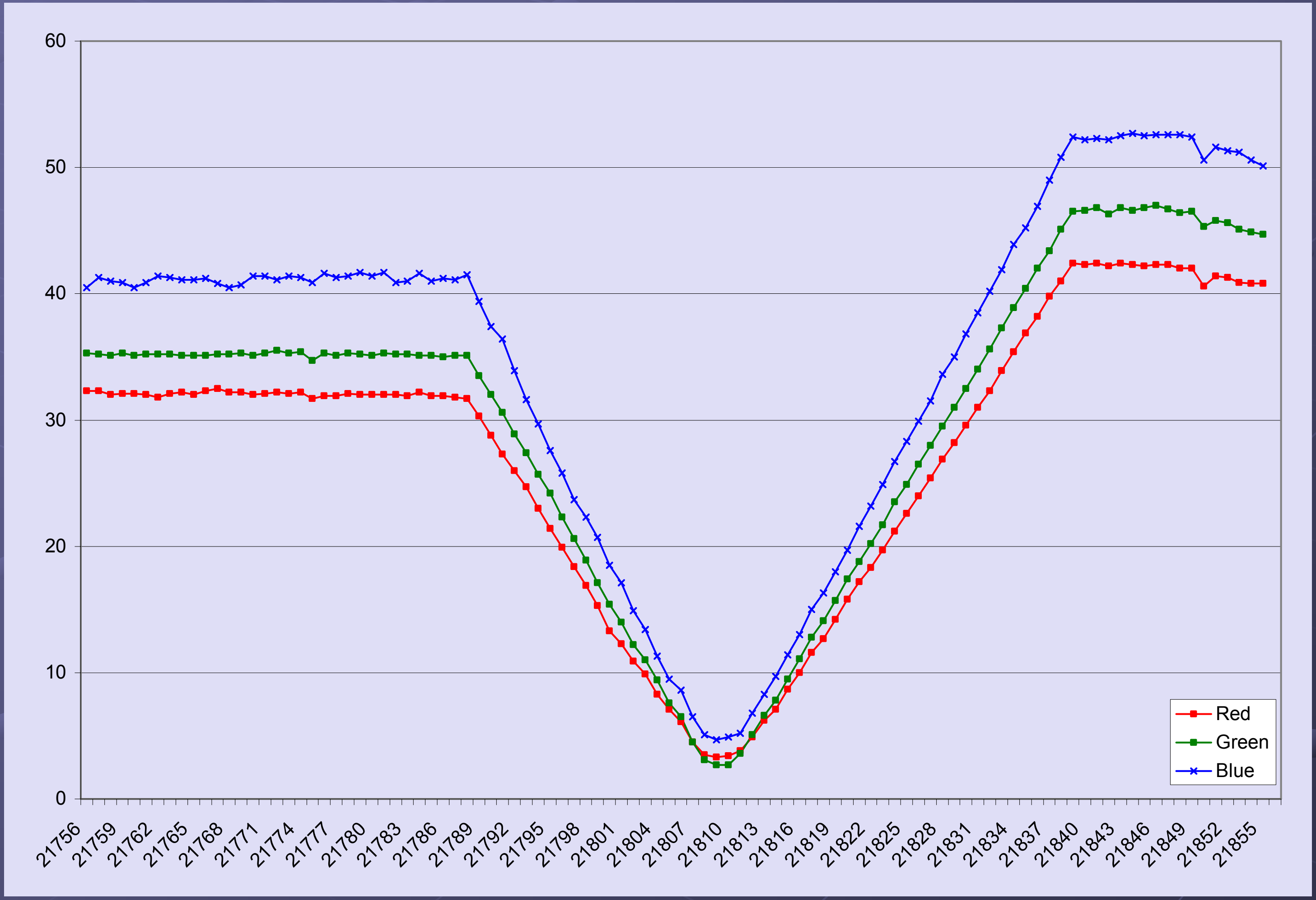
Model-based Detection Methods

- Implications of the transition models
  - Characteristic patterns in image feature time series
  - Transitions may be detected by recognizing patterns typical of each transition type
- Cut Detection
  - Identify abrupt changes in the time series
- Fade Detection
  - Find monotonically increasing or decreasing image variance sequences which start or end on a monochrome frame
- Dissolve Detection
  - Recognize parabolic sequences in the time series of image variance

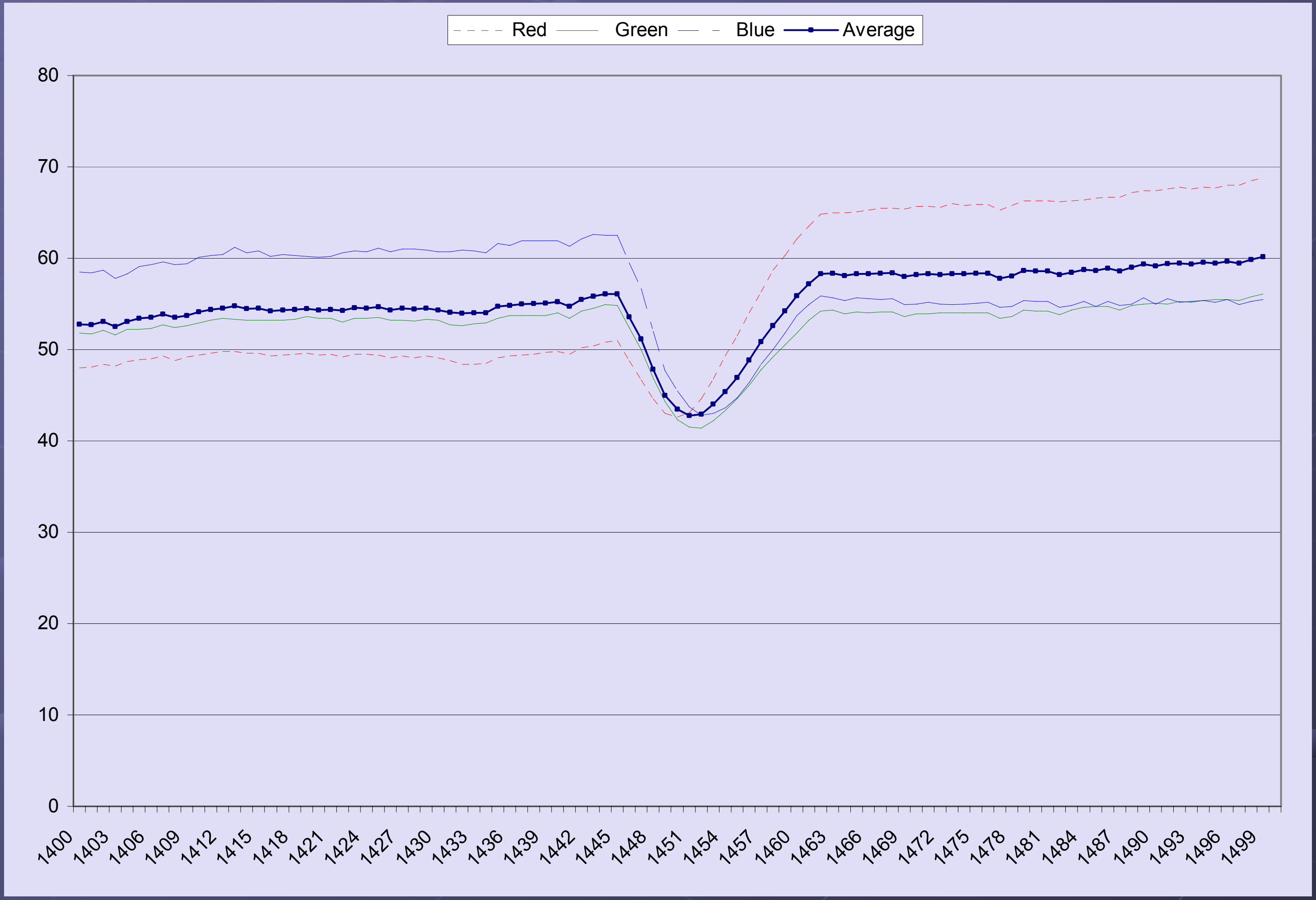
Cut Reflected in Color Mean



Fade-out and Fade-in
Reflected in Color Standard Deviation



Dissolve Reflected in Color Standard Deviation



Performance Evaluation

$$recall_x = R^x = \frac{number\ of\ correctly\ reported\ transitions\ x}{number\ of\ all\ transitions\ x}$$

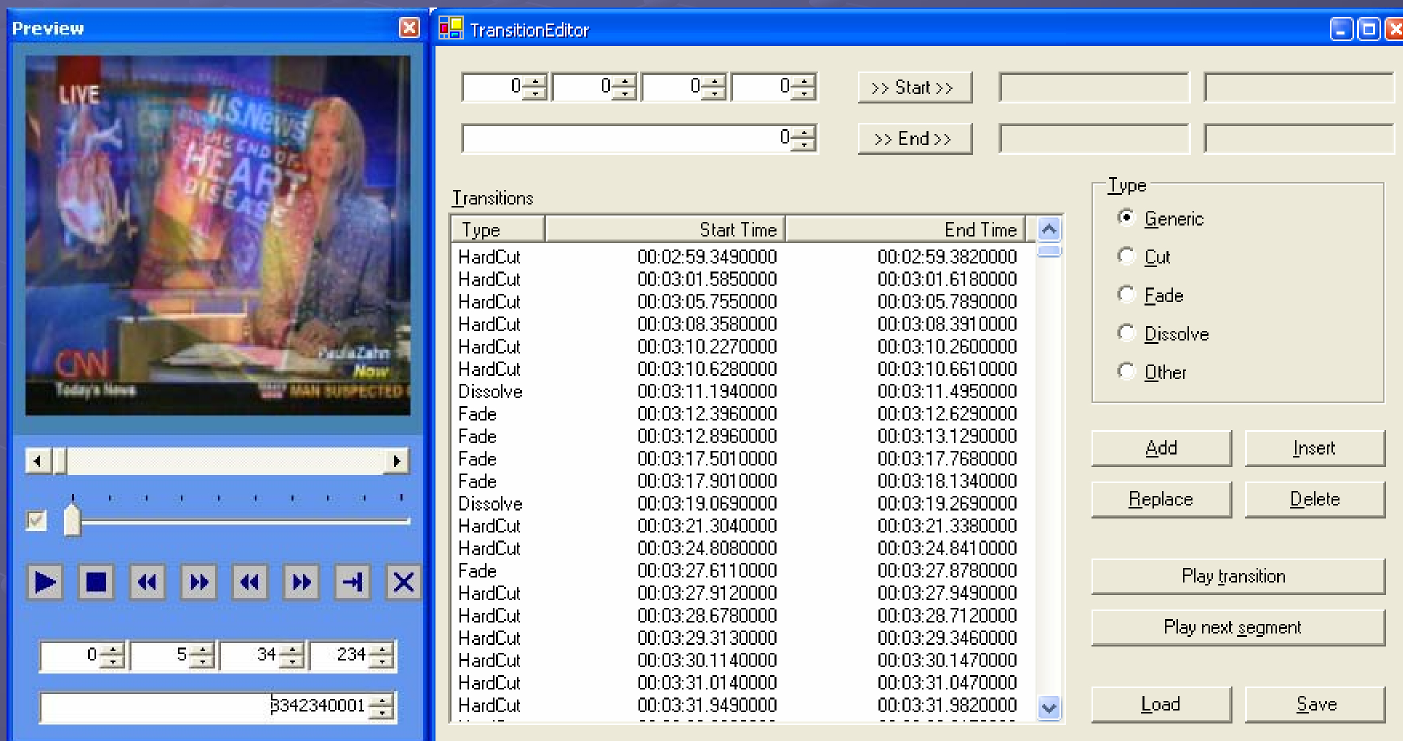
$$precision_x = P^x = \frac{number\ of\ correctly\ reported\ transitions\ x}{number\ of\ all\ reported\ transitions\ x}$$

- Correctly reported transitions
  - Reported transitions which overlap some actual transitions of the *same type*
- Missed transitions
  - Actual transitions which did not overlap *any* detected transitions
- False alarms
  - Detected transitions which did not overlap *any* actual transitions

Experimental Data

- Video
  - 60 minutes of a CNN News broadcast from Nov 11, 2003
  - Recorded using Windows Media Encoder
  - Format: 160x120 pixels, approx. 30 fps
- Ground Truth
  - Established manually – tedious!
  - 618 Cuts, 89 Fades, 189 Dissolves, 70 Special Effects

Transition Annotation GUI



Cut Detection

- Detect differences in color moments between consecutive frames
  - Declare a cut if difference exceeds an adaptive threshold
  - Threshold: Weighted sum of mean and standard deviation of moment difference over a window of frames

Cut Detection Performance

$$utility = \alpha \cdot recall + (1-\alpha) \cdot precision \qquad with\ \alpha = 0.5$$

Standard Deviation Coefficient (columns); Mean Coefficient (rows)

%	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5
0.5	50.39	49.84	49.39	49.26	48.97	47.76	46.26	2.91	0.00	0.00
1.0	51.05	51.99	53.86	59.98	76.12	90.58	84.29	0.00	0.00	0.00
1.5	62.62	71.51	81.91	90.12	92.09	87.80	58.87	0.00	0.00	0.00
2.0	81.18	87.19	90.98	92.20	88.90	78.98	51.45	0.00	0.00	0.00
2.5	88.74	90.99	91.37	89.56	83.97	71.42	0.00	0.00	0.00	0.00
3.0	90.94	91.24	89.88	85.80	78.29	62.97	0.00	0.00	0.00	0.00
3.5	91.01	89.73	86.87	81.90	73.37	58.45	0.00	0.00	0.00	0.00
4.0	89.63	88.01	83.53	78.11	68.52	55.12	0.00	0.00	0.00	0.00
4.5	88.47	85.51	80.48	74.57	63.65	53.07	0.00	0.00	0.00	0.00
5.0	86.42	82.39	78.35	71.84	60.32	51.88	0.00	0.00	0.00	0.00

Fade Detection

- Similar to algorithms existing in literature
- Algorithm
  - Detect monochrome frame sequences
  - Detect potential fade sequences around them
    - Search for peaks in a smoothed first derivative
  - Test for the following criteria
    - Slope minimum and maximum
    - Slope dominance threshold
- Performance is very high and equivalent to other available methods

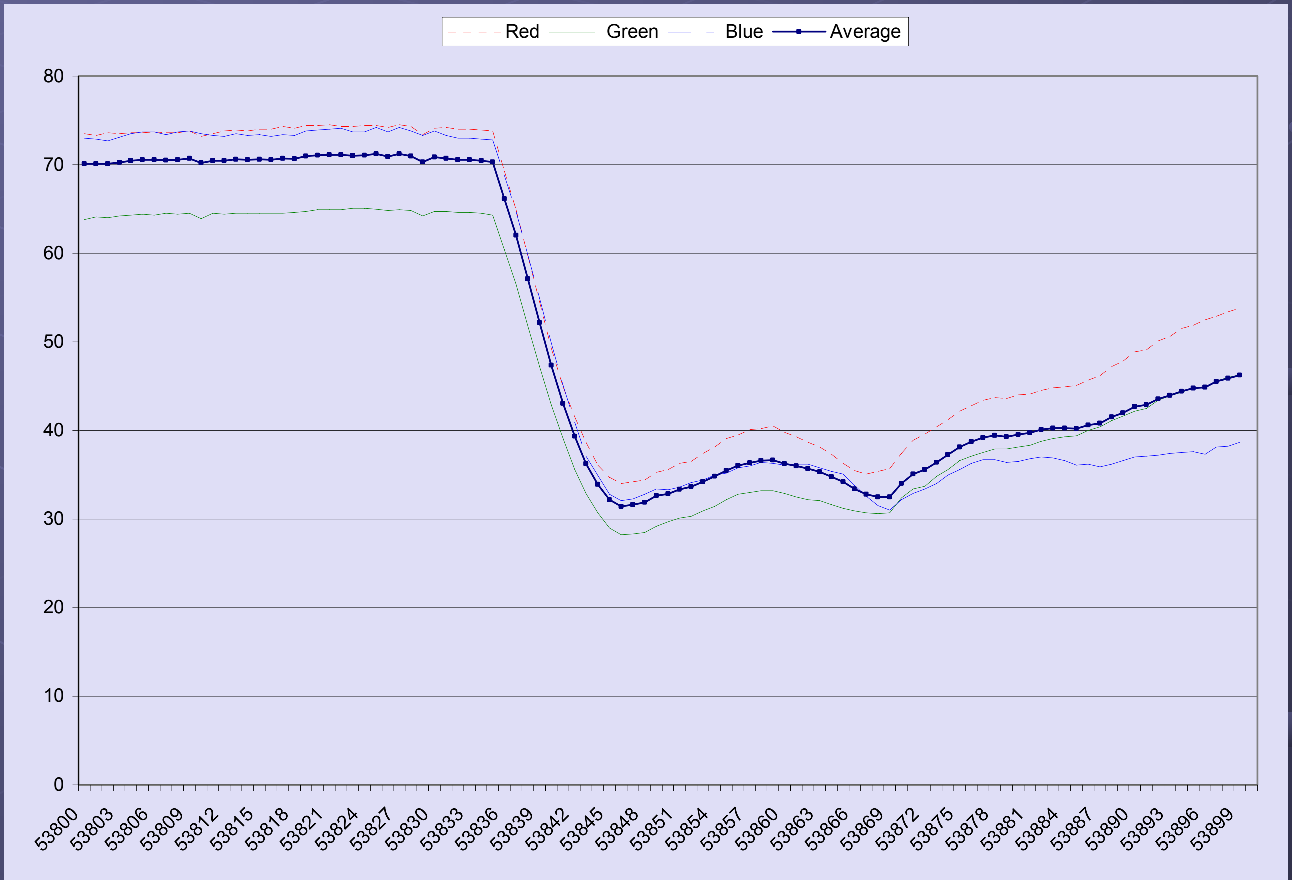
Fade Detection Performance

Minimal Slope	Recall	Precision	Utility
0.0	92.9%	97.5%	95.18%
0.5	92.9%	97.5%	**95.18%**
1.0	90.5%	98.7%	94.59%
1.5	82.1%	98.6%	90.36%
2.0	71.4%	98.4%	84.89%
2.5	67.9%	98.3%	83.07%
3.0	64.3%	98.2%	81.23%
3.5	58.3%	100.0%	79.17%
4.0	57.1%	100.0%	78.57%
4.5	51.2%	100.0%	75.60%
5.0	47.6%	100.0%	73.81%

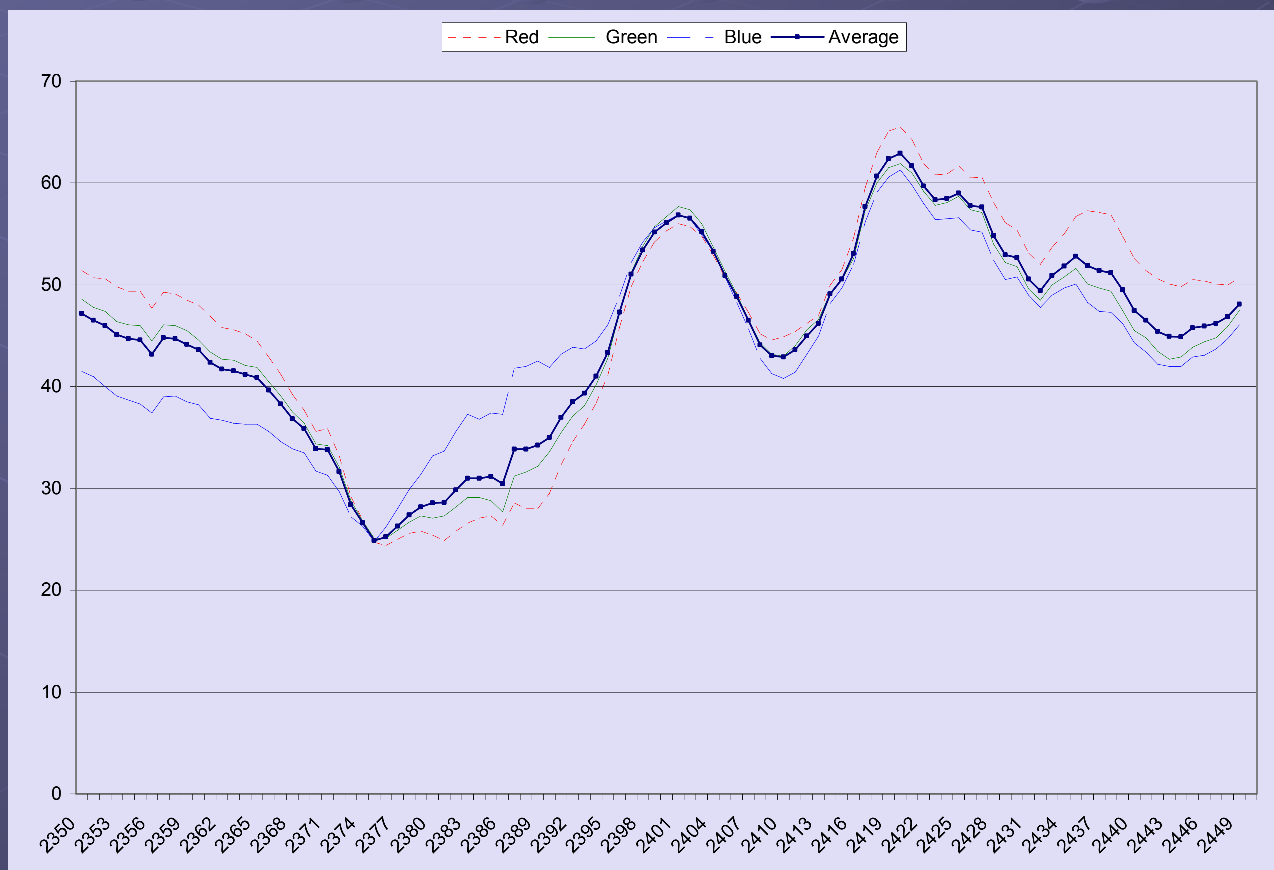
Dissolve Detection

- Detect parabolic shape in variance curve
- Problems
  - Parabolic shape may be highly distorted
  - Similar patterns are caused by motion and camera pans
- Solution
  - Detect minimum of the variance curve
  - Apply additional conditions to improve precision
- Truong proposes a set of four conditions on variance
  - Performance: recall and precision ~65%

Dissolve Detection



Dissolve Detection



Our Approach

- Observation
  - Color mean should change linearly during dissolve
- Method
  - Remove one of the conditions on variance
  - Added a condition on mean
- Result
  - Increased precision

Dissolve Detection Performance

Condition	Match	False Alarm	Missed	Recall	Precision	Utility
Minimum Variance	186	5786	3	98.4%	3.1%	50.76%
Minimum Length	185	3410	4	97.9%	5.1%	51.51%
Min Bottom Variance	184	3345	5	97.4%	5.2%	51.28%
Start/End Variance Diff	170	194	19	89.9%	46.7%	68.33%
Average Variance Diff	164	**95**	25	86.8%	63.3%	75.05%
Center Mean	158	**45**	31	**83.6%**	**77.8%**	**80.72%**

15%
improvement

Temporal Video Segmentation Conclusions

- Overall performance
  - Cut detection: recall 90%, precision 95%
  - Fade detection: recall 93%, precision 98%
  - Dissolve detection: recall 83%, precision 78%
- Future work
  - Dissolve detection leaves room for improvement
  - Special effect detection should be explored

Repeated Video Sequence Detection

Problem Definition

- Goal
  - Detect repetitions of video footage for purposes of story tracking
- Challenges
  - *Sequence Matching*
    - Handle partially matching sequences
  - *Repetition Detection*
    - There are over 20,000 shots in typical a 24-hour broadcast
    - All pairs of shots need to be considered
    - The process must be completed in real-time

Video Sequence Matching

- Develop Similarity Metrics corresponding to visual similarity
  - Frame similarity metric
  - Complete sequence similarity
  - Partial sequence similarity
- Establish similarity levels required for sequences to be considered matching

Related Work

- Semantic Video Retrieval
  - Determine if two video sequences have conceptually similar content
  - Cognitive gap – machines are currently unable to identify high level concepts
- Video Co-Derivative Detection
  - Determine if two video sequences have been derived from the same source
  - Received less attention in research community
  - Hoad and Zobel propose three methods of measuring co-derivative similarity: cut pattern, centroid position pattern, intra-frame color change
  - Cheung develops video signature based on random vectors in image feature space
  - **Partial sequence similarity has not been explored**

Frame Similarity Metric

$$V^x = \left\langle M^x(t,r), M^x(t,g), M^x(t,b), S^x(t,r), S^x(t,g), S^x(t,b), K^x(t,r), K^x(t,g), K^x(t,b) \right\rangle$$

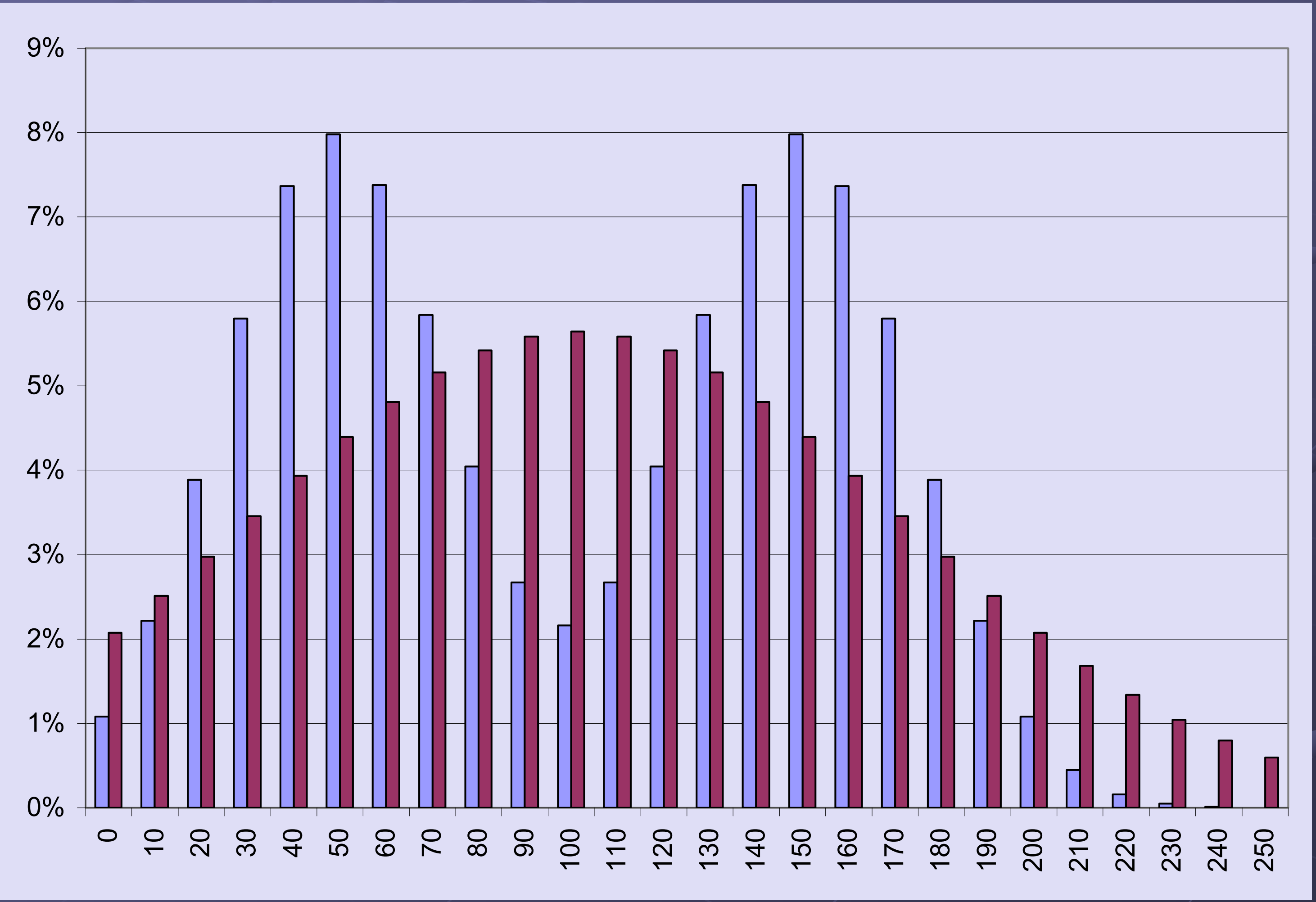
$$FrmSim\left(f^a, f^b\right) = 1 - FrameAvgMomentDiff\left(f^a, f^b\right)$$

$$FrameAvgMomentDiff\left(f^a, f^b\right) = \frac{1}{9}\left(\sum_{i=1}^{9} L_p\left(V_i^a, V_i^b\right)\right)$$

$$L_p\left(V_i^a, V_j^b\right) = \left[\left(\left|V_i^a(t,c) - V_i^b(t,c)\right|\right)^p\right]^{\frac{1}{p}}$$

$$f^a \approx f^b \Leftrightarrow FrmSim\left(f^a, f^b\right) \geq frameMatchThreshold$$

Color Moments as Frame Representation



Complete Sequence Similarity Metrics

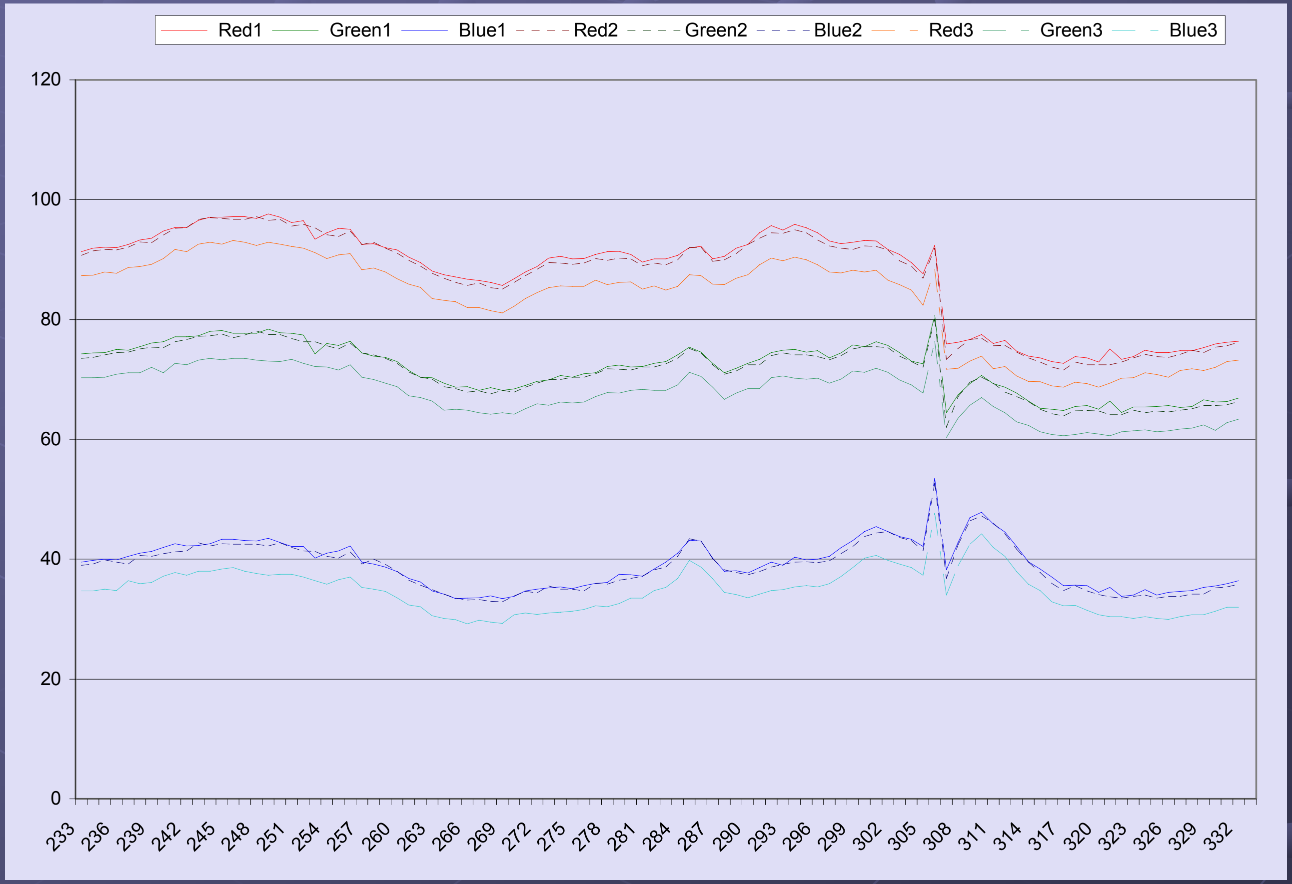
$$S_a = \left\langle f_1^a, f_2^a, \ldots, f_N^a \right\rangle \quad and \quad S_b = \left\langle f_1^b, f_2^b, \ldots, f_N^b \right\rangle$$

$$ClipSim(S_a, S_b) = \frac{1}{N} MatchingFrameCount(S_a, S_b) = \frac{1}{N} \sum_{i=1}^{N} frameMatch\left(f_i^a, f_i^b\right)$$

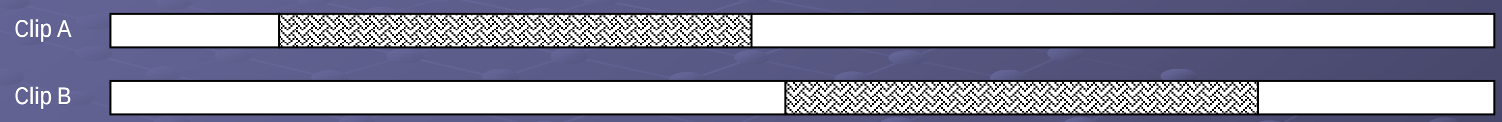
$$frameMatch\left(f_i^a, f_i^b\right) = \begin{cases} 1 \quad if \quad f_i^a \approx f_i^b \\ 0 \qquad Otherwise \end{cases}$$

$$S_a \approx S_b \Leftrightarrow ClipSim(S_a, S_b) \geq clipMatchThreshold$$

Color Moments as Sequence Representation



Partial Sequence Similarity Metric



$$PartialClipSim(S_a, S_b) = \max\left(\forall SS_a, SS_b : ClipSim(SS_a, SS_b)\right)$$

$$where \quad SS_x = \left\langle f_j^x, f_{j+1}^x, \ldots, f_{j+k}^x \right\rangle \; and \; 1 \le j < j+k \le N_x$$

$$and \quad k+1 \ge L$$

- *L* is the *significant length threshold*
  - Prevents accidental matching of very short subsequences

Partial Sequence Matching

- Optimal threshold values
  - *frameMatchThreshold* = 3.0
  - *L* = 30 frames
  - *clipMatchThreshold* = 0.50
- Determined experimentally
  - Using a 24-hour CNN News broadcast
  - Selected values producing best recall and precision

Other Observations

- Other metrics considered
  - Normalized color moment metric
  - Color moment difference metric
- Unsuitable for video news broadcasts
  - Work well for sequences with substantial motion
  - Do not work for static sequences, such as anchor persons, studios, interviews

Repetition Detection

- Develop methods of detecting repeated sequences in a live video broadcast
- Related Work
  - Gauch developed commercial detection system using color moments as frame feature
  - Pua used color moment hashing and filtering to detect repeated video sequences
  - Our research extended their work to handle partial repetition detection

Detection Methods

- Exhaustive sequence matching
  - Choose every pair of subsequences in the broadcast
  - Compute similarity metric value, i.e. compare frame by frame
- Exhaustive shot matching
  - Choose every pair of shots in the broadcast
  - Compute partial similarity metric
    - Align the shots in every way for which the overlap is at least ΔL
    - Compare overlapping sequences frame by frame
- Filtered shot matching
  - Determine which shots have a potential to match
  - Compute partial similarity metric only for the potentially matching shots

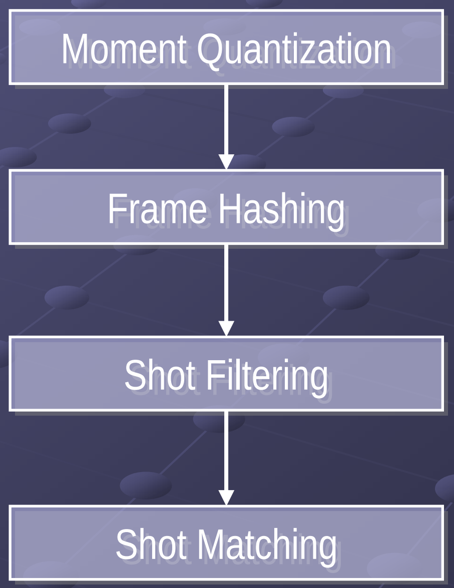
Time Complexity

- Let
  - *n* be the number of frames in the broadcast
    - In 24-hour broadcast at 30fps n = 2.9 million
  - *c* be the number of shots in the broadcast
    - In 24-hour broadcast *c* is approx. 20,000, *c* is proportional to *n*
  - *p* be the average shot length
    - *p* is independent of *n*, p=n/c ~ 150 frames
  - *f* be the fraction of potentially matching shots
- Exhaustive Sequence Matching
  - $O(n^4)$
- Exhaustive Shot Matching
  - $O(c^2 * p) = O(n^2/p)$
- Filtered Shot Matching
  - $O(c * c * f * p) = O(fn^2/p)$
  - The only viable alternative for real-time detection

Filtered Shot Matching Algorithm

- Moment Quantization
  - Assign each frame to a hyper-cube of color moment space
  - Uniformly quantize color moments
    - $qV_i = floor(V_i / qStep)$
    - $qStep = 6.0$
- Frame Hashing
  - Compute hash value for every frame
  - Place each frame in a hash table

$$hv = \prod_{i=1}^{9} i \cdot (qV_i + 1) \bmod hashTableSize$$



Filtered Shot Matching Algorithm

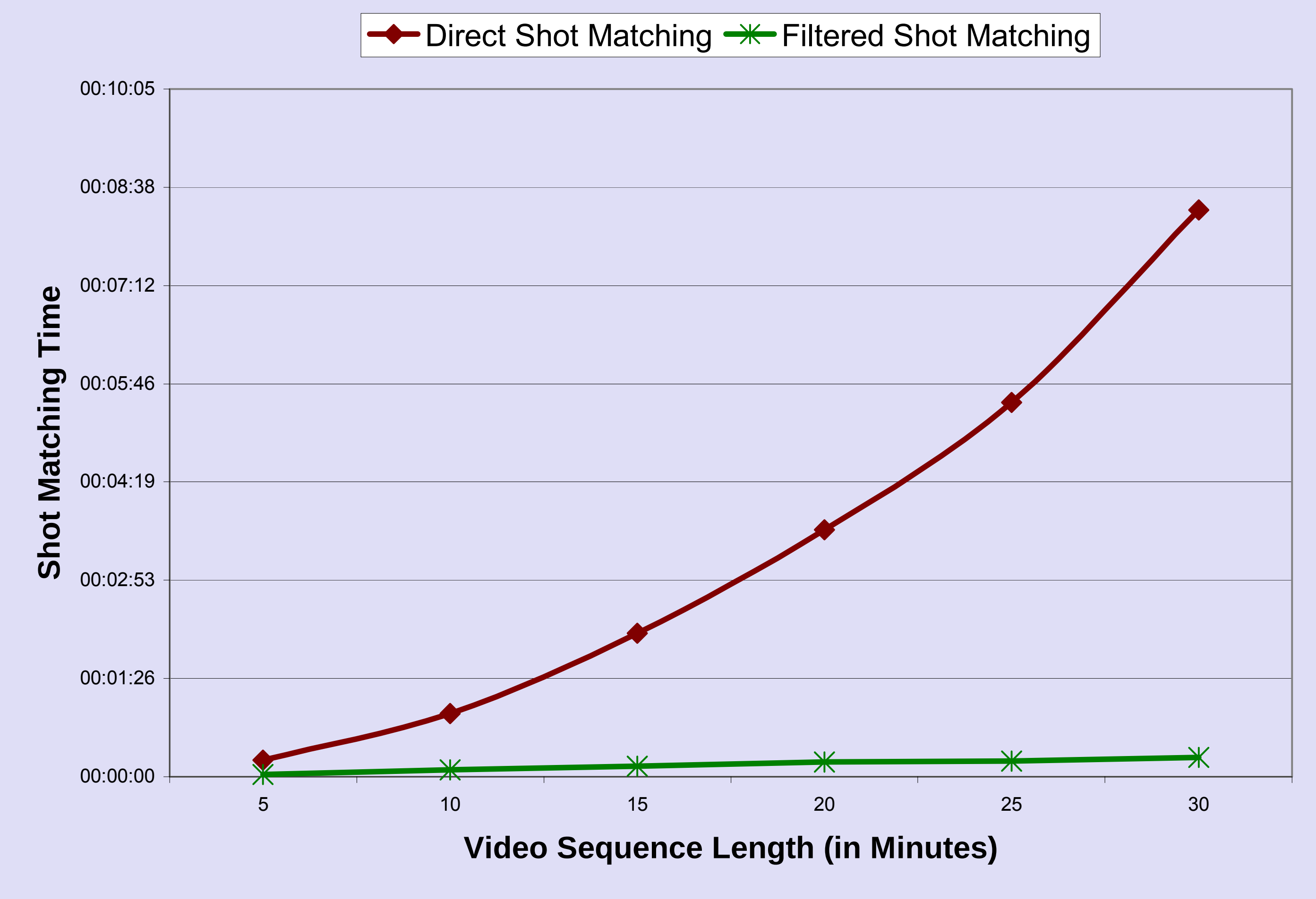
- Shot Filtering
  - For a given shot ***s*** find potentially matching shots
  - Consider every frame in ***s***
  - Find all other frames with the same quantized moments
    - Retrieve from hash table
  - Compute q-similarity for every shot ***v***
    - Number of frames in ***v*** and in ***s*** whose quantized moments are equal
  - Chose shots with q-similarity > *qSimThreshold*
    - *qSimThresh* = 10 frames
- Shot Matching
  - Compute partial similarity metrics for every pair of potentially matching shots

Shot Matching Performance

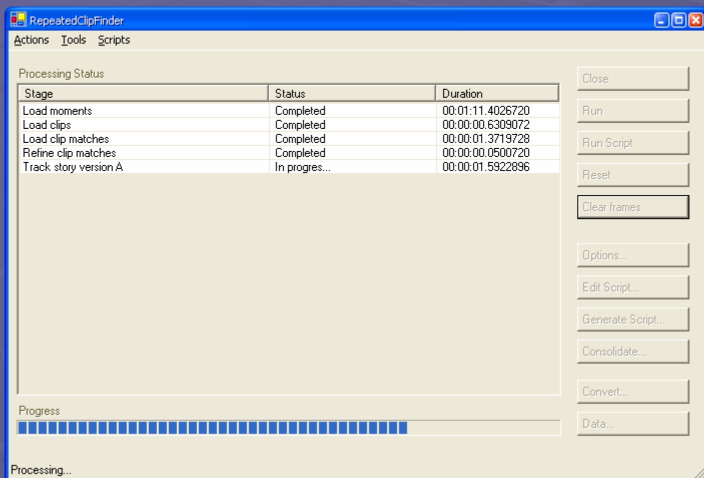
Shot No.	No. of Frames	True Matches	Detected Matches	True Positives	False Positives	False Negatives	Recall	Precision
5925	553	2	2	2	0	0	100%	100%
7611	266	6	8	5	3	1	83%	63%
7612	360	6	7	6	1	0	100%	86%
7613	1017	3	4	2	2	1	67%	50%
9509	457	5	5	5	0	0	100%	100%
9514	76	3	2	2	0	1	67%	100%
9524	167	4	4	4	0	0	100%	100%
11490	321	6	5	5	0	1	83%	100%
18323	309	3	3	3	0	0	100%	100%
19750	776	4	6	3	3	1	75%	50%
Overall							**86%**	**91%**

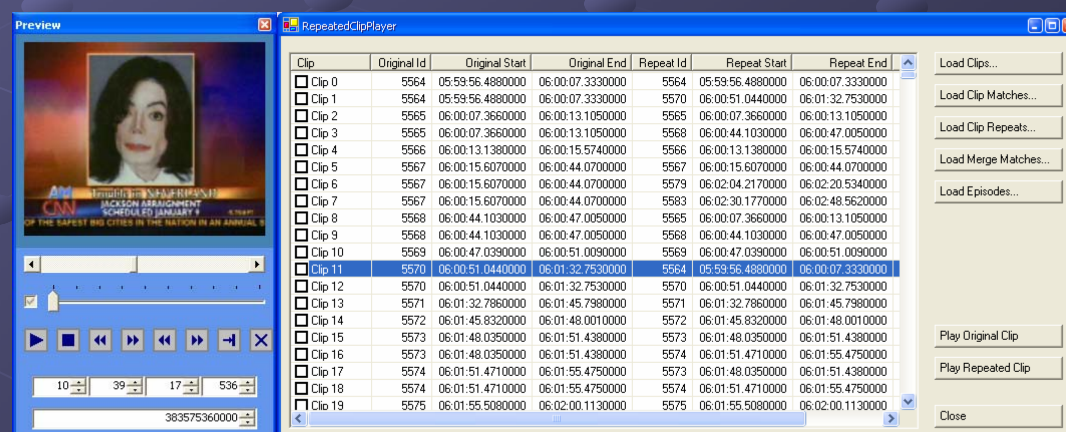
- Performance equivalent to exhaustive shot matching
- Substantially faster

Shot Matching Execution Time



Shot Matching Demo





Repeated Sequence Detection Conclusions

- Results
  - Successfully detected partially repeated video sequences in live news broadcast
    - Recall 88%, Precision 85%
  - Adapted shot filtering to partial matching
- Future Work
  - Development of similarity metrics which can handle
    - Changes in brightness
    - Slow motion repetitions
  - Creation of automatic methods for
    - Detection of picture-in-picture mode
    - Removal of on-screen captions

Story Tracking

Story Tracking

- Goal
  - Given information about user's interest in a certain news story, follow and report the development of the story over time.
- Related Work
  - Story tracking was first proposed as a problem of textual information retrieval
  - Became one of the tasks of the Topic Detection and Tracking
  - Pioneering work was done by Allan *et al.*
- Visual story tracking is a novel approach

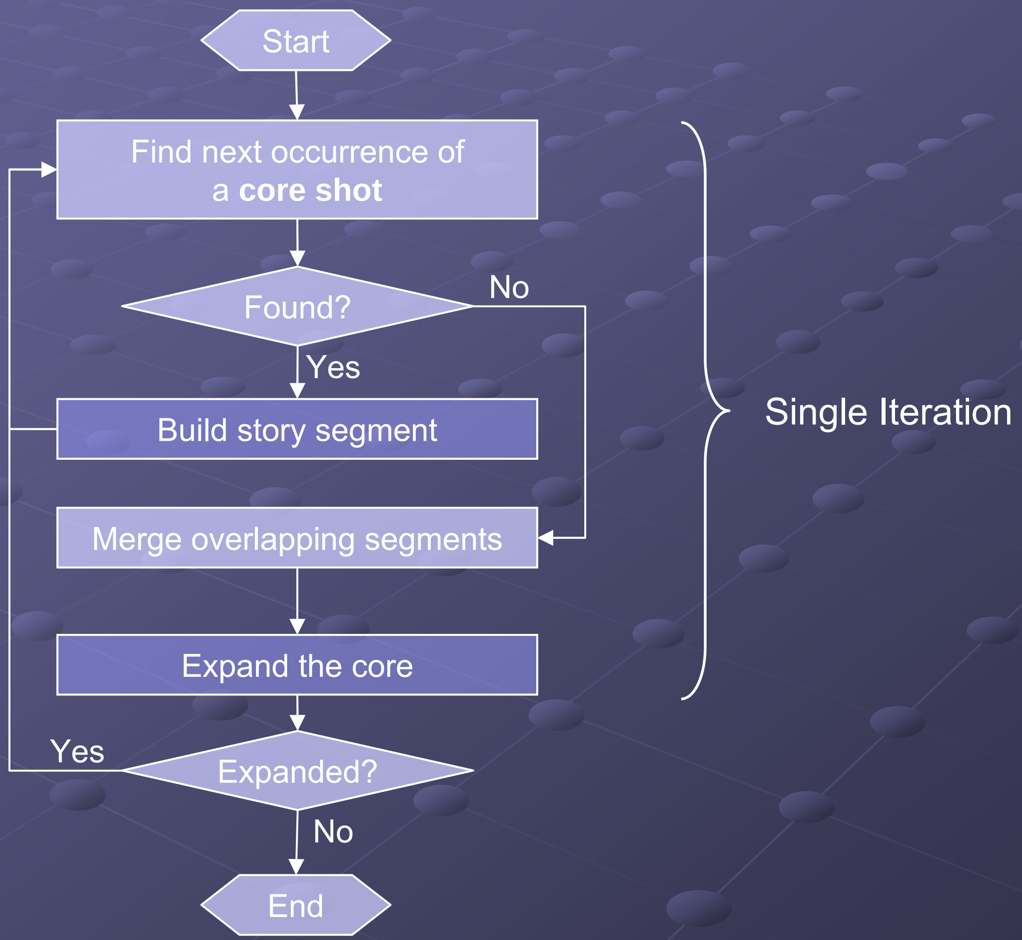
Overview

- Visual Story Tracking
  - **News Story**: event or set of events which are reported in the news
  - **Story:** a set of all shots in a video broadcast which are relevant to the *news story* of interest
  - **Task:** Given a set of query shots relevant to a news story, detect the **story**

Approach

- Approach
  - Define the story core as the set of query shots
  - Detect occurrences of the core shots
  - Build story segments around them
  - Identify other relevant shots and add them to the core
    - As the story evolves and new footage becomes available its subsequent repetitions are detected by the algorithm

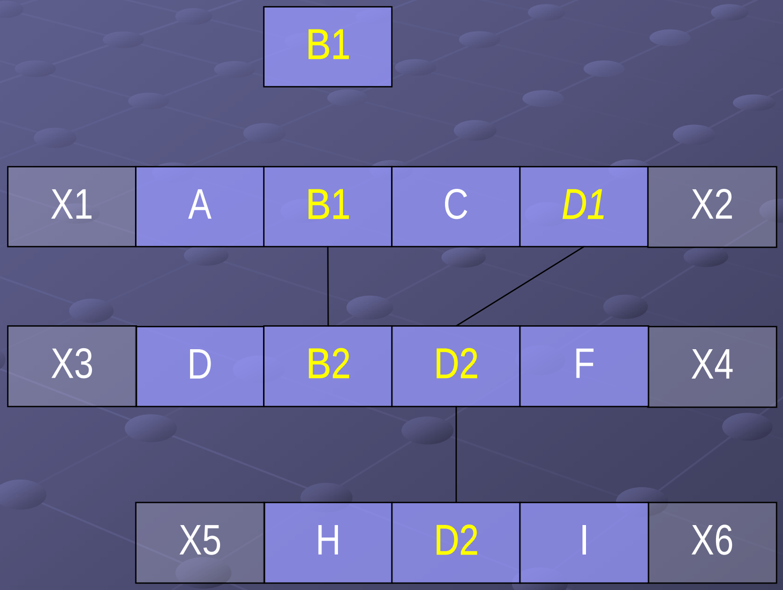
Story Tracking Algorithm



Important Phases

- Segment Building
  - Define story segment as a sequence of shots around the core shot
  - Sequence length is determined by the **neighborhood size** (*w*) given in minutes
- Core Expansion
  - Every modified segment is checked for potential new core shots
  - A shot is added to the core if it occurs at least a given number of times in the segments of the story
  - Required number of occurrences is determined by the **co-occurrence threshold** (*tc*)

Graphical Story Representation



Formal Story Representation

Story Board

Story Core
Subset of Σ containing shots whose repetitions are detected

Partition induced on Σ by the shot matching equivalence relation

$$SB_{\Phi} = \left\langle \Sigma, \Omega, \mathrm{P}(\Sigma), \delta, \gamma \right\rangle$$

Set of shots belonging to the story

Co-Occurrence Function
assigns no-zero values to shots in the same segment

Shot Classification Function
labels shots as anchors, commercials, etc.

Experimental Data

- Video Source
  - 18-hour broadcast of CNN News channel
  - Recorded on Nov 4, 2003
  - Format: Windows Media Video, 160x120 pixels, 30 fps
  - Size: ~30GB
- Story
  - Regarding Michael Jackson's arrest in connection with child abuse charges
  - 16 segments of various lengths
    - From 30 seconds to almost 10 minutes
  - 17 repeating shots
  - The entire broadcast was viewed by a human observer, and all segments of the story were manually detected to establish the ground truth

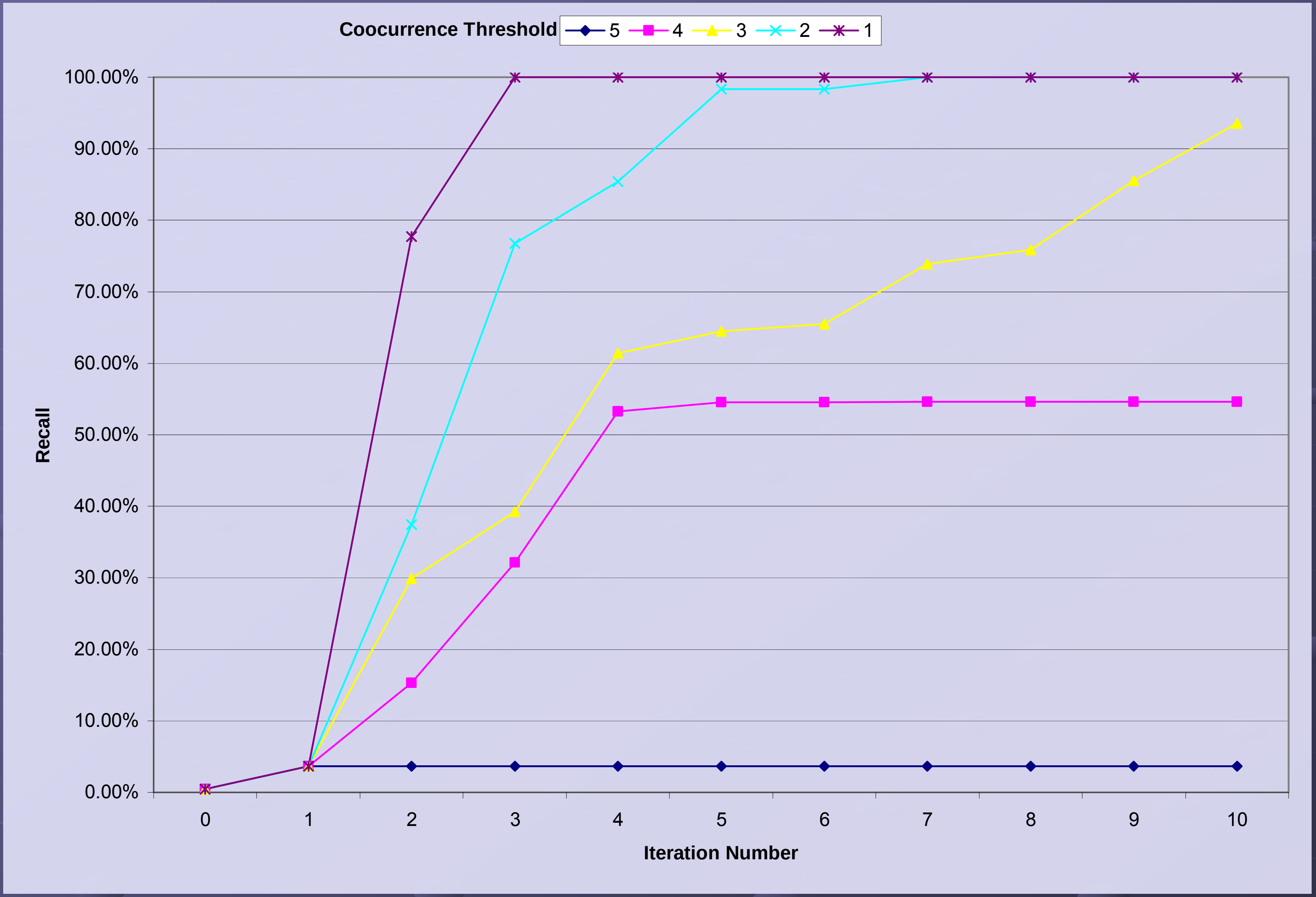
Ground Truth for Story Tracking

Experiments

- Queries
  - Three queries corresponding to three segments of the story
  - Different duration and number of query shots
- Parameters
  - Range of neighborhood sizes
  - Range of co-occurrence thresholds

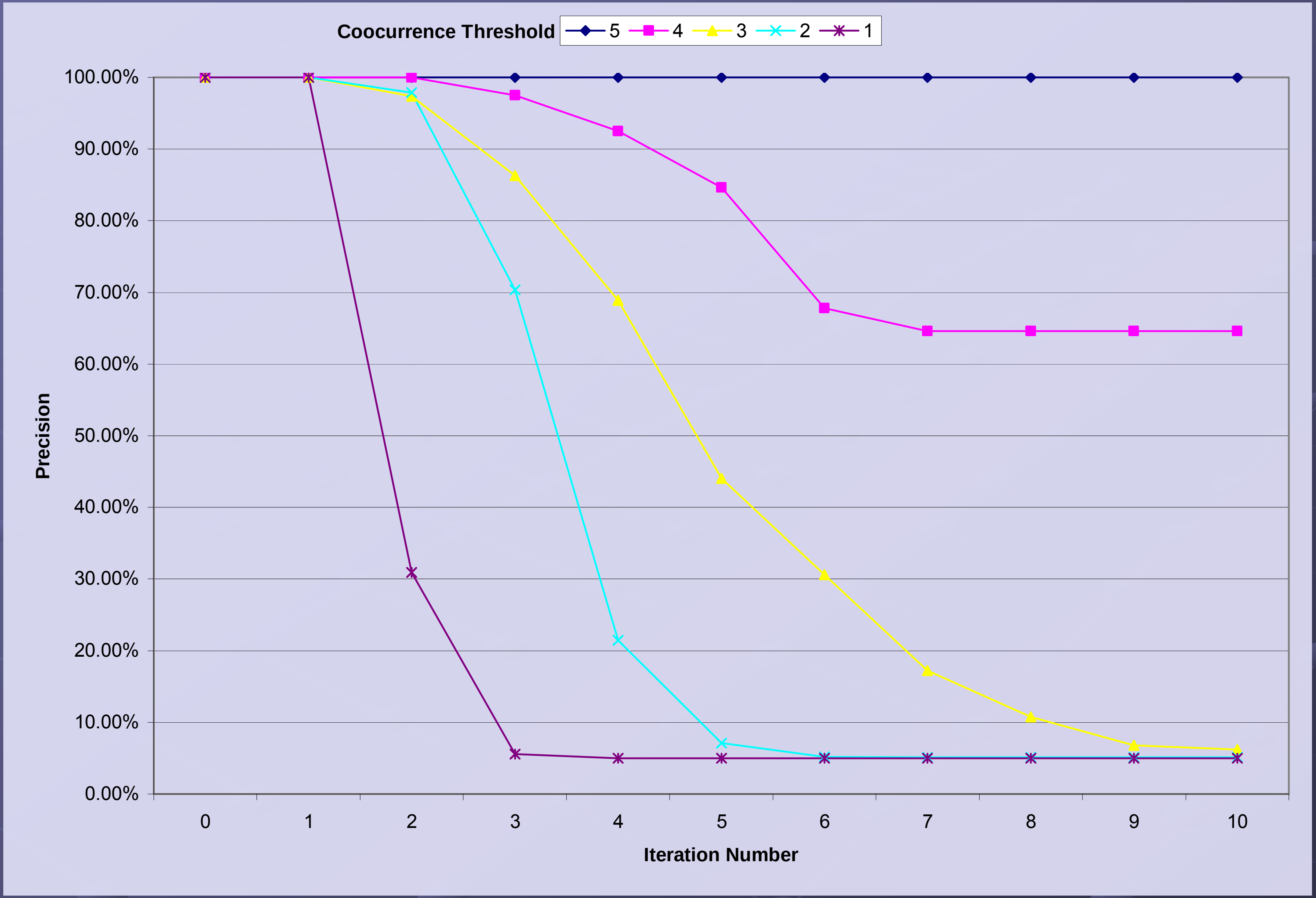
Segment No.	Segment Duration	Query Size (shots)
3	0:35	1
5	0:21	3
6	4:22	6

Recall



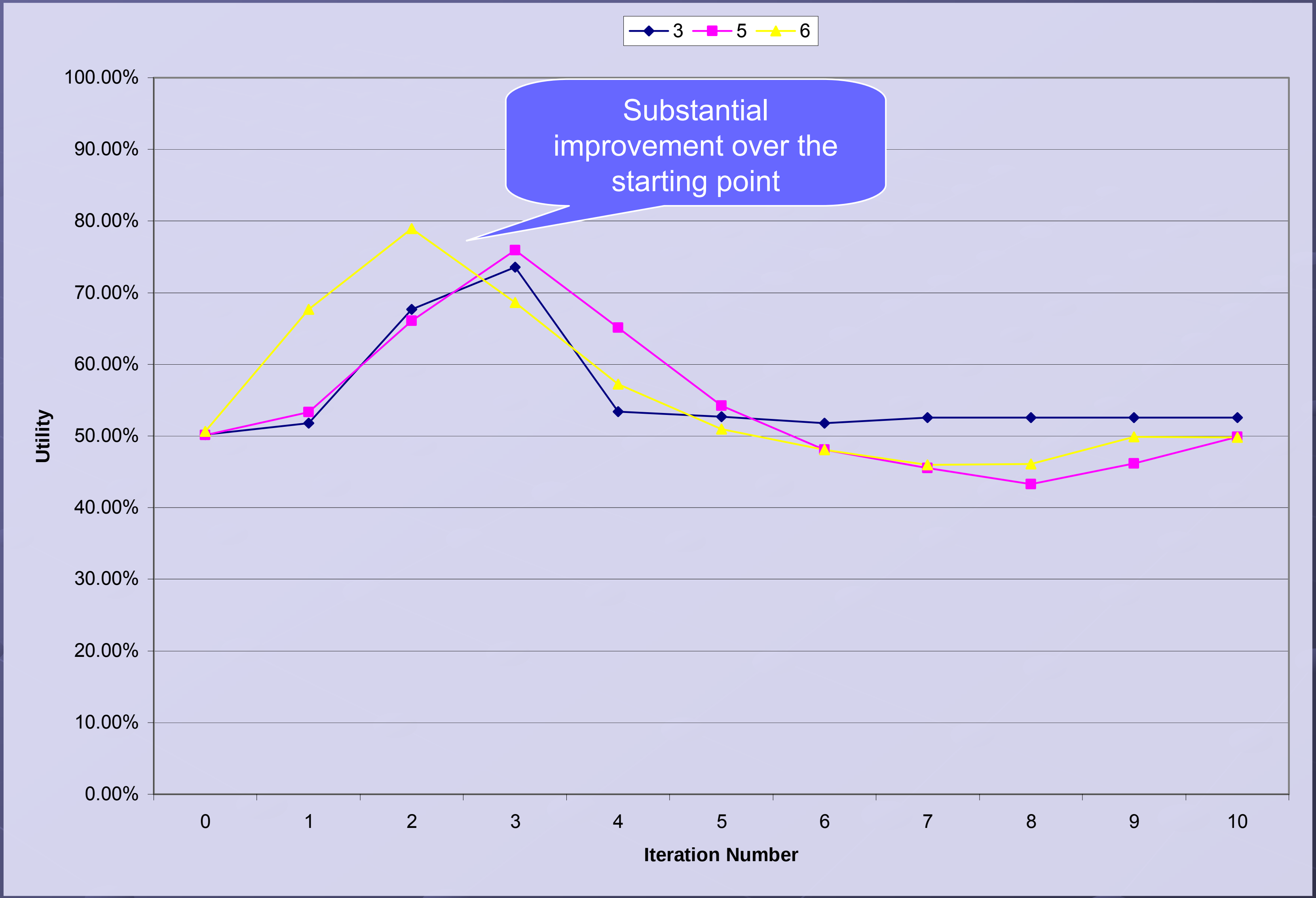
Coocurrence Threshold
5
4
3
2
1
100.00%
90.00%
80.00%
70.00%
60.00%
50.00%
40.00%
30.00%
20.00%
10.00%
0.00%
Recall
0
1
2
3
4
5
6
7
8
9
10
Iteration Number

Precision



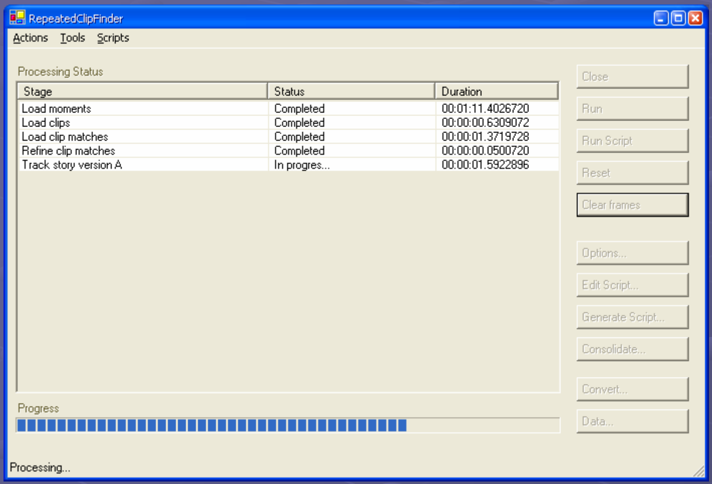
Coocurrence Threshold
5
4
3
2
1
Precision
100.00%
90.00%
80.00%
70.00%
60.00%
50.00%
40.00%
30.00%
20.00%
10.00%
0.00%
0
1
2
3
4
5
6
7
8
9
10
Iteration Number

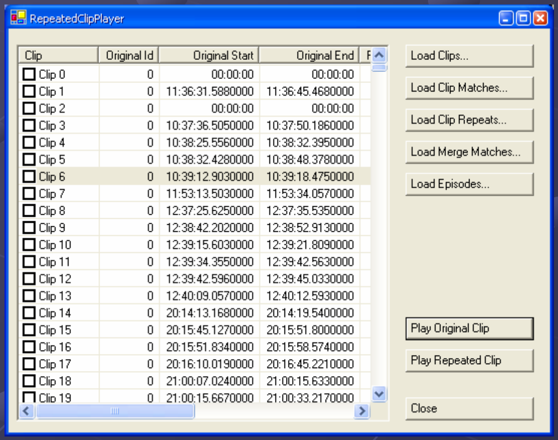
Utility



3
5
6
Substantial improvement over the starting point
100.00%
90.00%
80.00%
70.00%
60.00%
50.00%
40.00%
30.00%
20.00%
10.00%
0.00%
Utility
0
1
2
3
4
5
6
7
8
9
10
Iteration Number

Story Tracking Demo





Performance Analysis

- Segment Building
  - Segments built by the algorithm are often extended past the end of actual segments
- Core Expansion
  - Commercials
    - Repeat frequently throughout the broadcast
    - Are often erroneously added to the core
    - Cause the story to grow out of control
  - Anchor persons
    - Detected as matching by the shot matching algorithm
    - If included in the core, produce the same effect as commercials

Story Tracking Conclusions

- Overall Performance
  - Recall and Precision approx. 75%
  - Small number of iterations is optimal
  - Story tracking works well even for very small queries
- Future Work
  - News shot classification techniques can improve performance
    - Commercial detection
    - Anchor person shot identification

Conclusion

Story tracking in news video broadcasts can be effectively performed based on detection of repeated video footage.

Primary Contribution

- Development of cut, fade, and dissolve detection technique using color moments
  - Compact representation
  - Performance equivalent to other methods
  - Substantial improvement (15%) of dissolve detection performance for news video
- Creation of method for partial video sequence repetition detection in live broadcasts
  - Partial sequence similarity metric
  - Adaptation of shot filtering methods for partial matching
- Invention of a novel story tracking technique

Future Work

- Temporal Segmentation
  - Further improvement of dissolve detection methods
  - Exploration of techniques for identification of computer effects
- Repeated Sequence Detection
  - Similarity metrics capable of dealing with global sequence changes
  - Detection methods for picture-in-picture content
  - Automatic on-screen caption removal
- Story Tracking
  - Automated new shot classification methods
  - Multimodal story tracking techniques
    - Textual and visual story tracking methods could be combined to fully realize the merits of both means of conveying information

Thank You

Questions

?