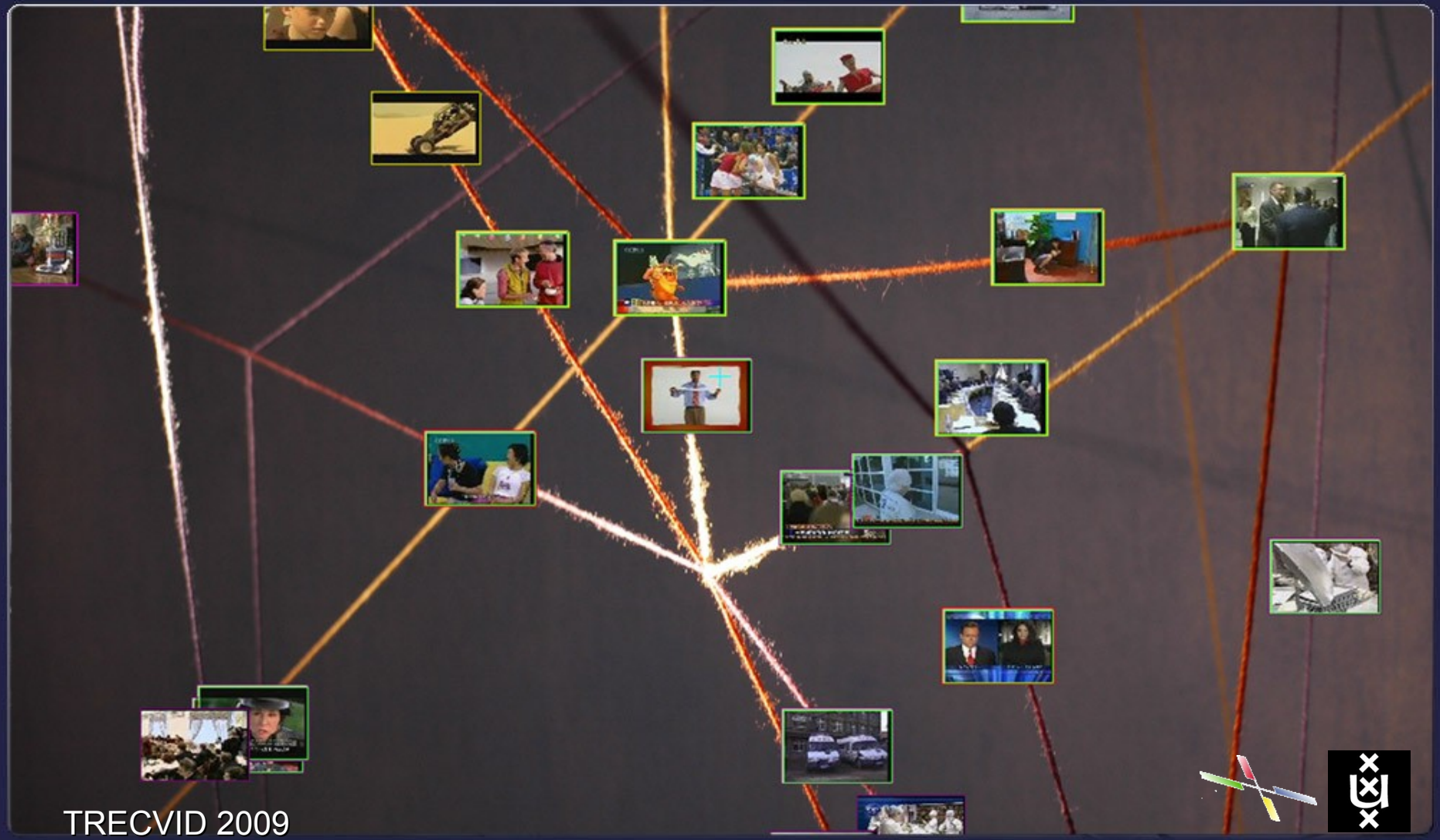
Learning From Video Browse Behavior

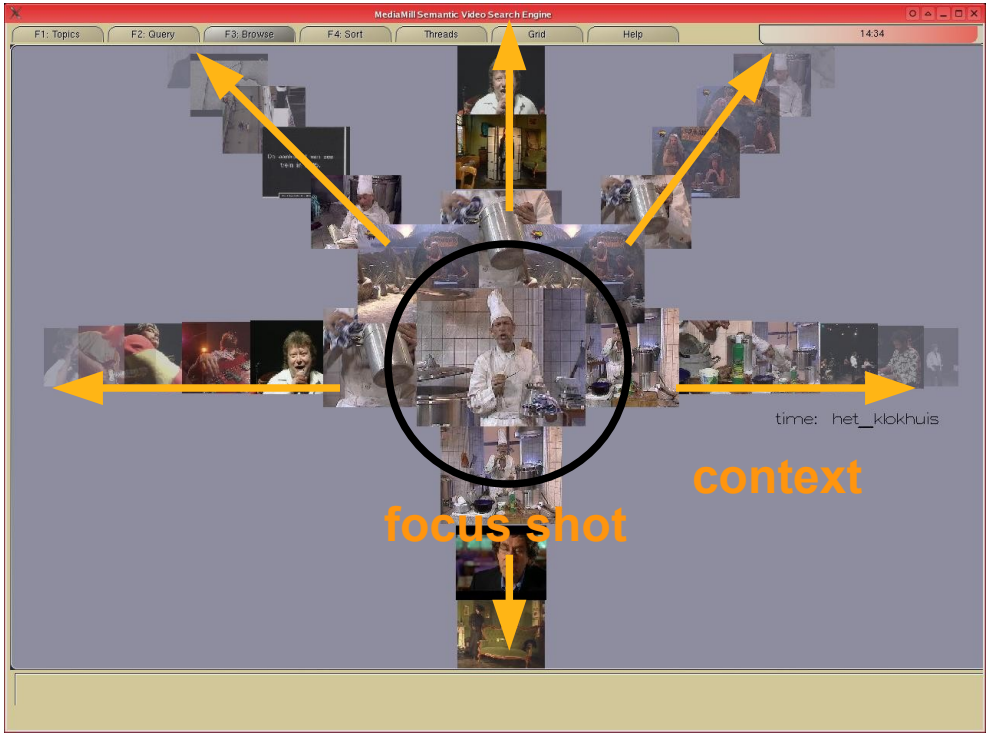


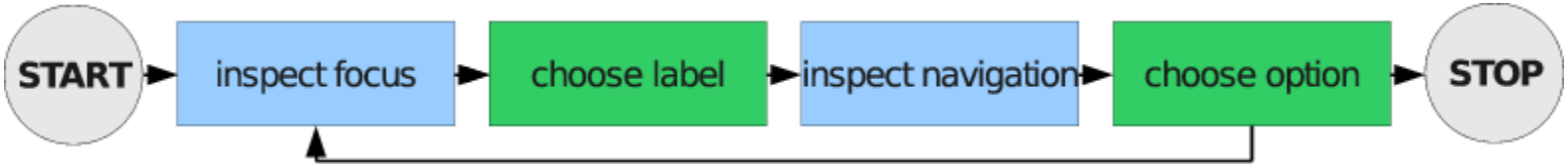
TRECVID 2009

Problem Statement

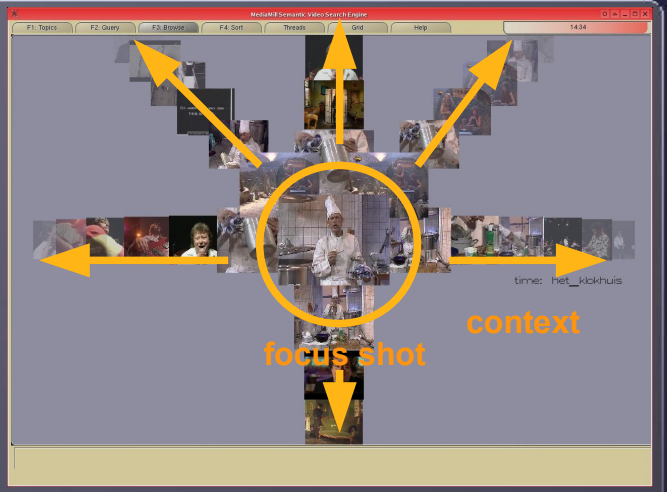
- Starting results relatively weak
  - Combination of query methods troublesome
- Possible solutions:
  - Optimize result selection
  - Visualize multiple query methods simultaneously
  - Analyze user browse behavior

Optimize Result selection?
Focus + Context browsing





Focus + Context browsing

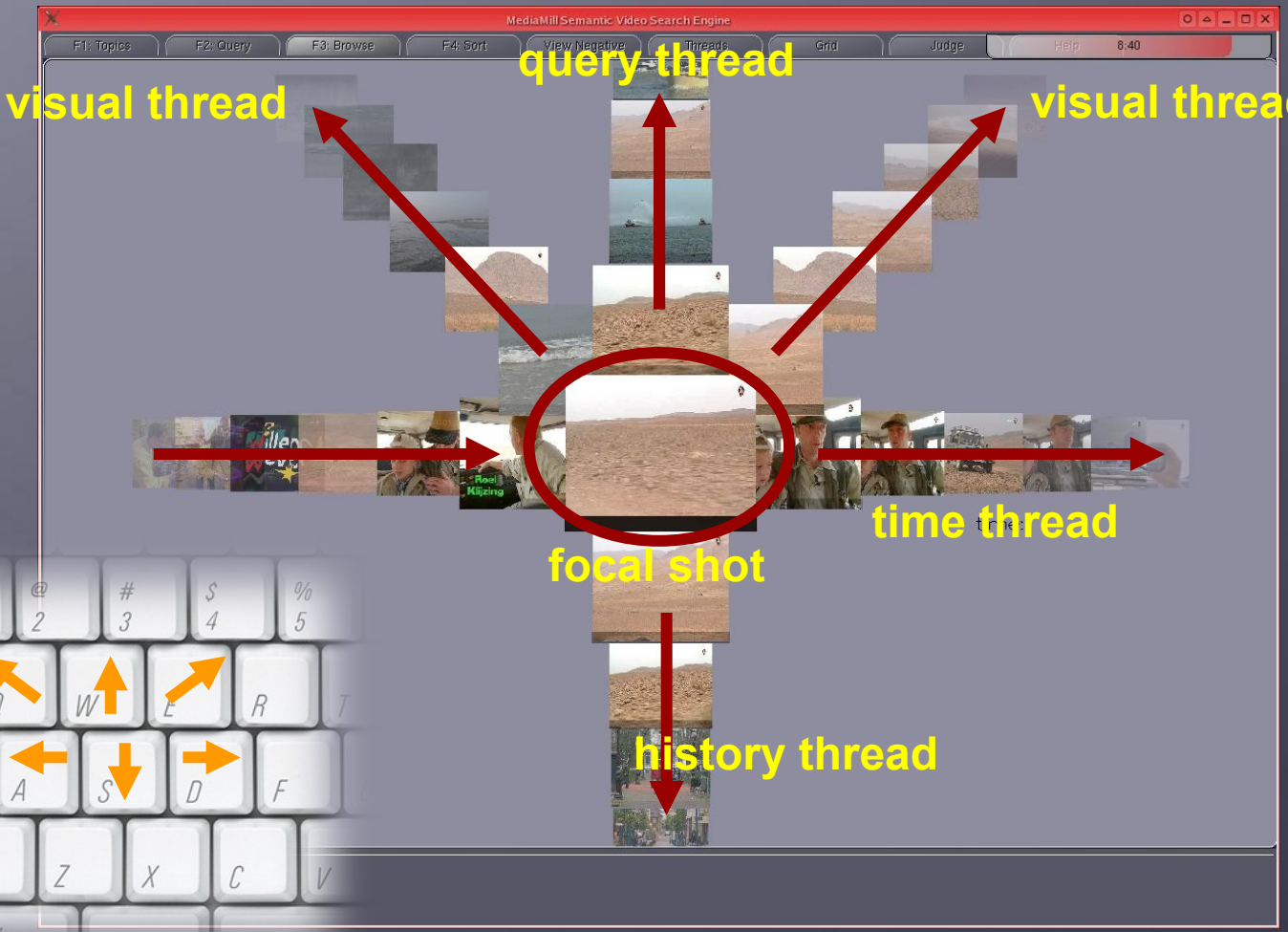


- Focus:
  - defined by the current focal shot
- Context:
  - defined by the rest of the interface
  - We use: multi thread browsing
    - *A thread is a linked sequence of shots in a specified order, based upon an aspect of their content*

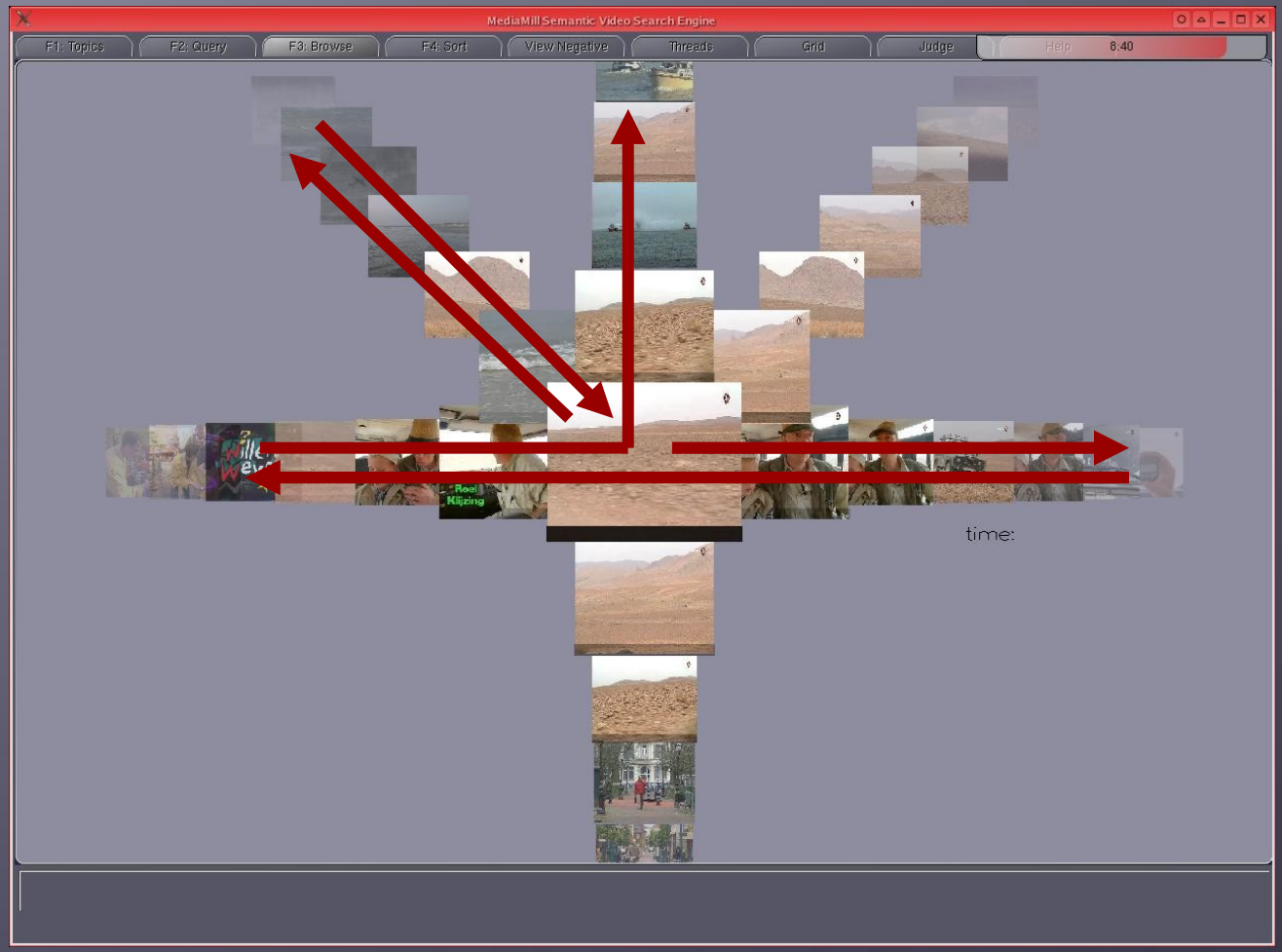
Threads used

- query threads
  - merged result of query-by-text and/or query-by-concept and/or query-by-example
- time threads
  - based on the shots in the video containing the focal shot
- visual threads
  - based on visual similarity of focal shot
- history thread
  - based on the previous user browse behavior

Multi Thread Browsing: ForkBrowser



Multi Thread Browsing: ForkBrowser



Problem Statement

- Starting results relatively weak
  - Combination of query methods troublesome
- Possible solutions:
  - Optimize result selection

We propose:
Focus + Context

  - Visualize multiple query methods simultaneously

We propose:
Multi Thread Browsing

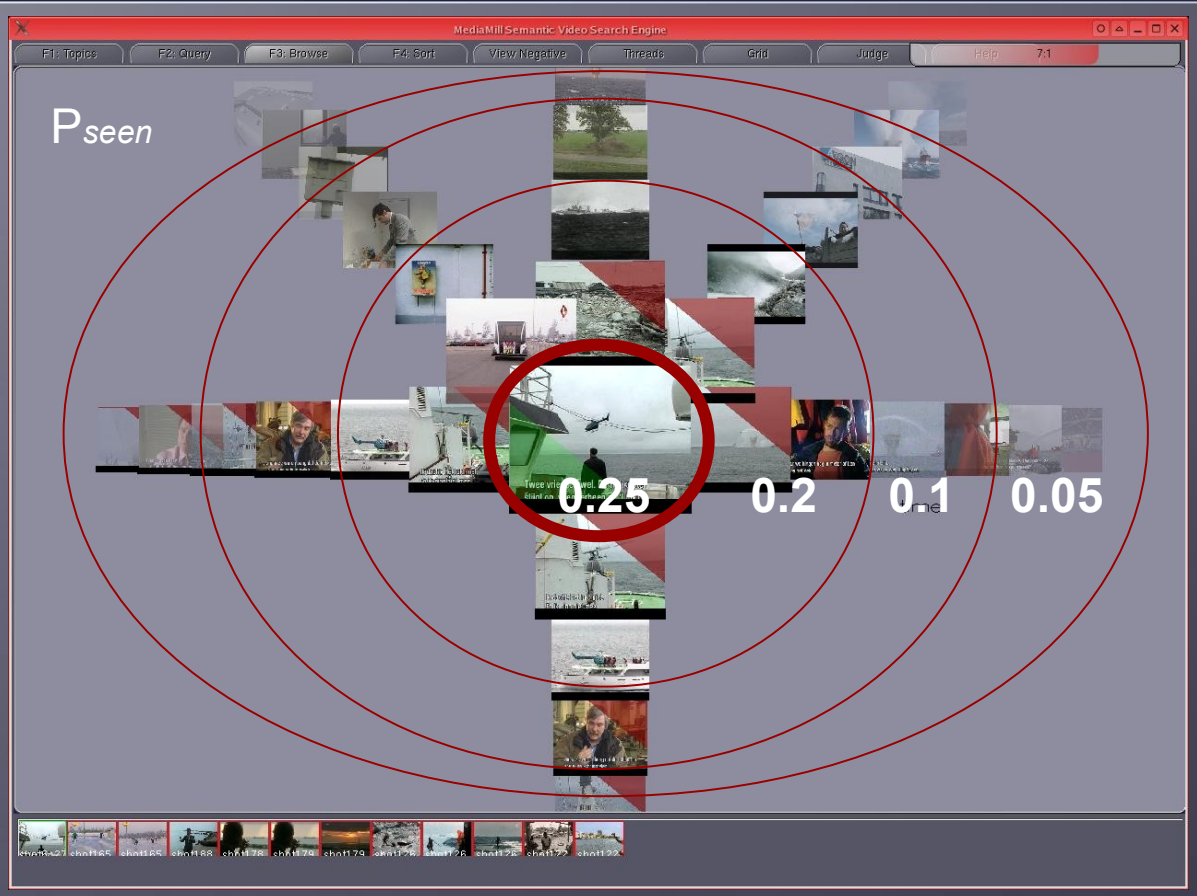
  - Analyze user browse behavior

We propose:
Relevance Feedback
based on context

Relevance Feedback based on Context

- Based on online SVM learning
  - User provides positive annotations
  - System gathers negative annotations based on user browse behavior
    - using displayed context
- User switches query thread when current results seem exhausted

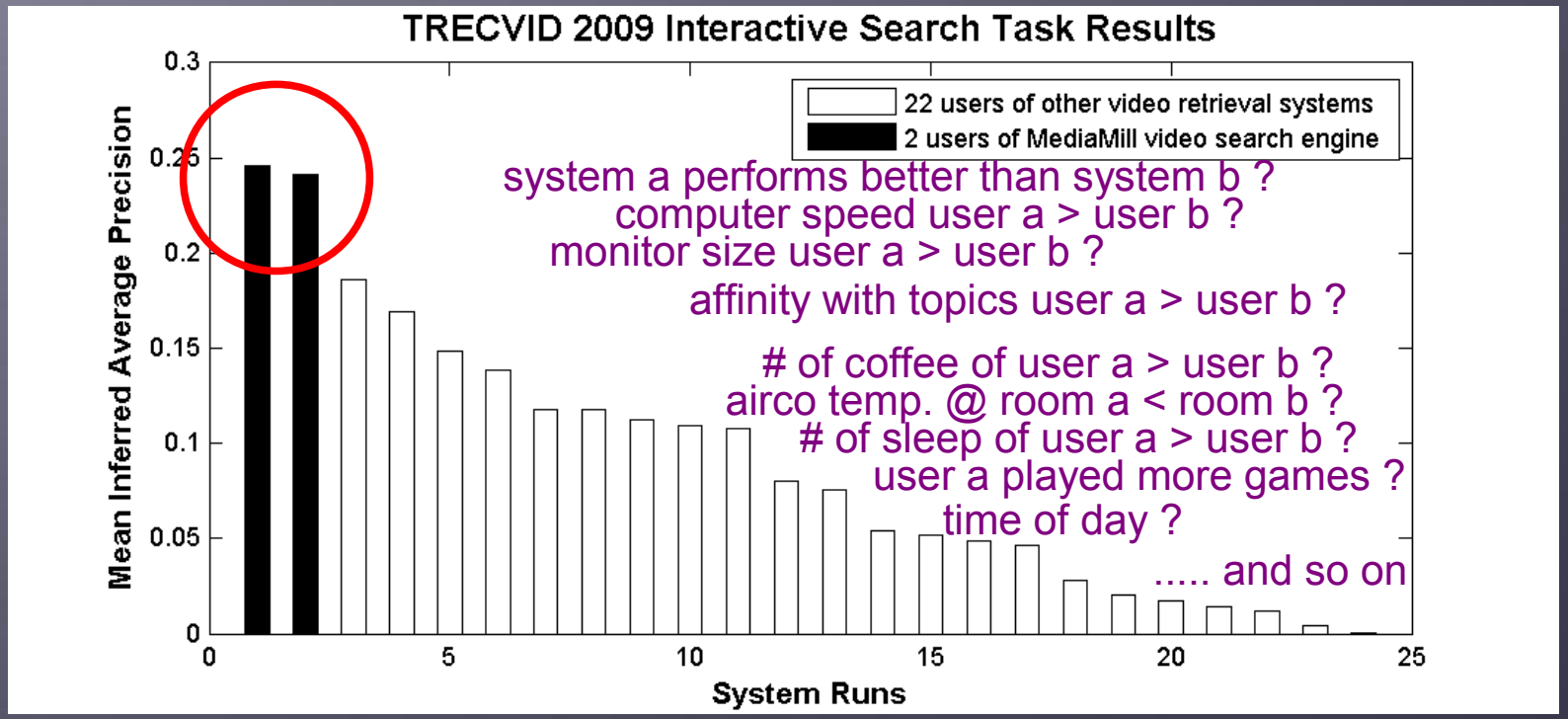
Relevance Feedback based on Context



- All displayed shots accumulate a score to have been seen by the user
- When a shot reaches a threshold that shot is used as a negative example

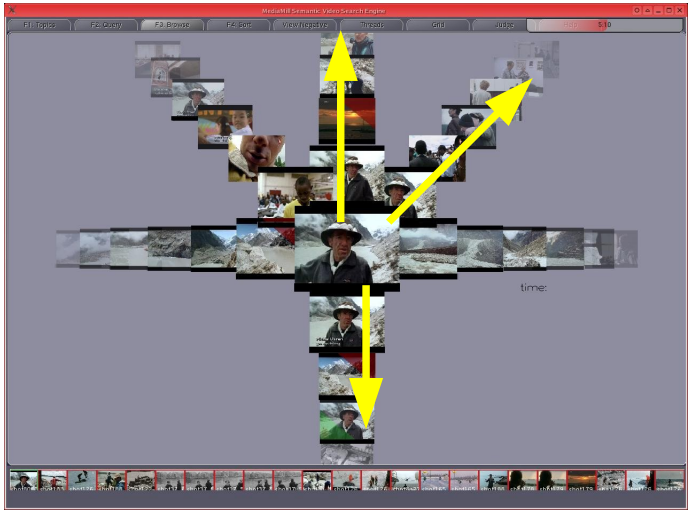
How to evaluate performance?

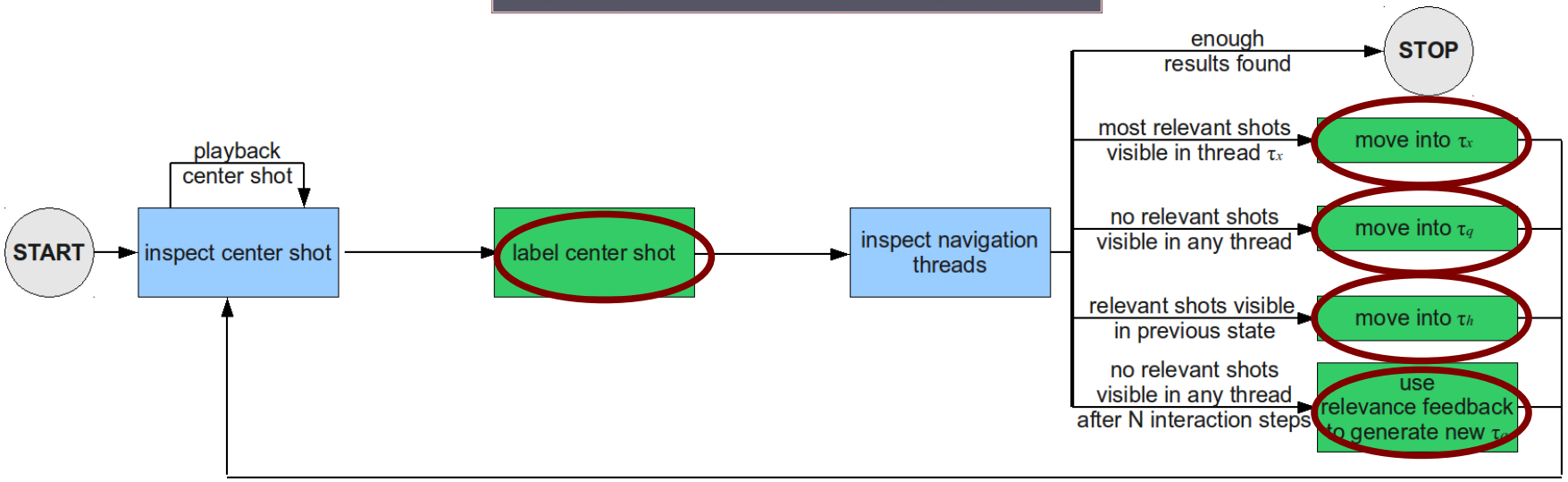
- Problem with measuring real world users



- Component level evaluation requires user simulation

User Simulation with a State Machine



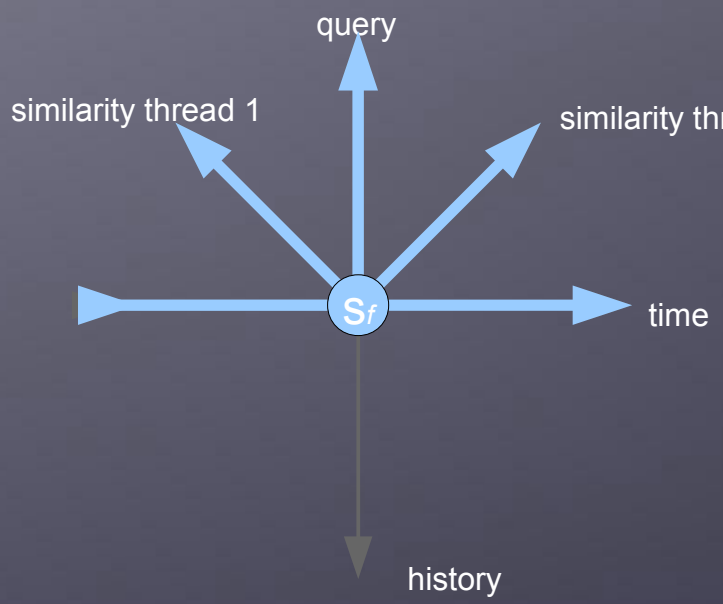


Experimental Setup

- TRECVID 2008 dataset
  - 200 hours of video
  - 48 topics, with (incomplete) annotations
  - 57 semantic concepts (21 of '08, 37 of '07)
  - best concepts taken as optimal starting query
- Experiment A:
  What is the benefit of having multiple threads?
- Experiment B:
  When should a user switch to relevance feedback results?

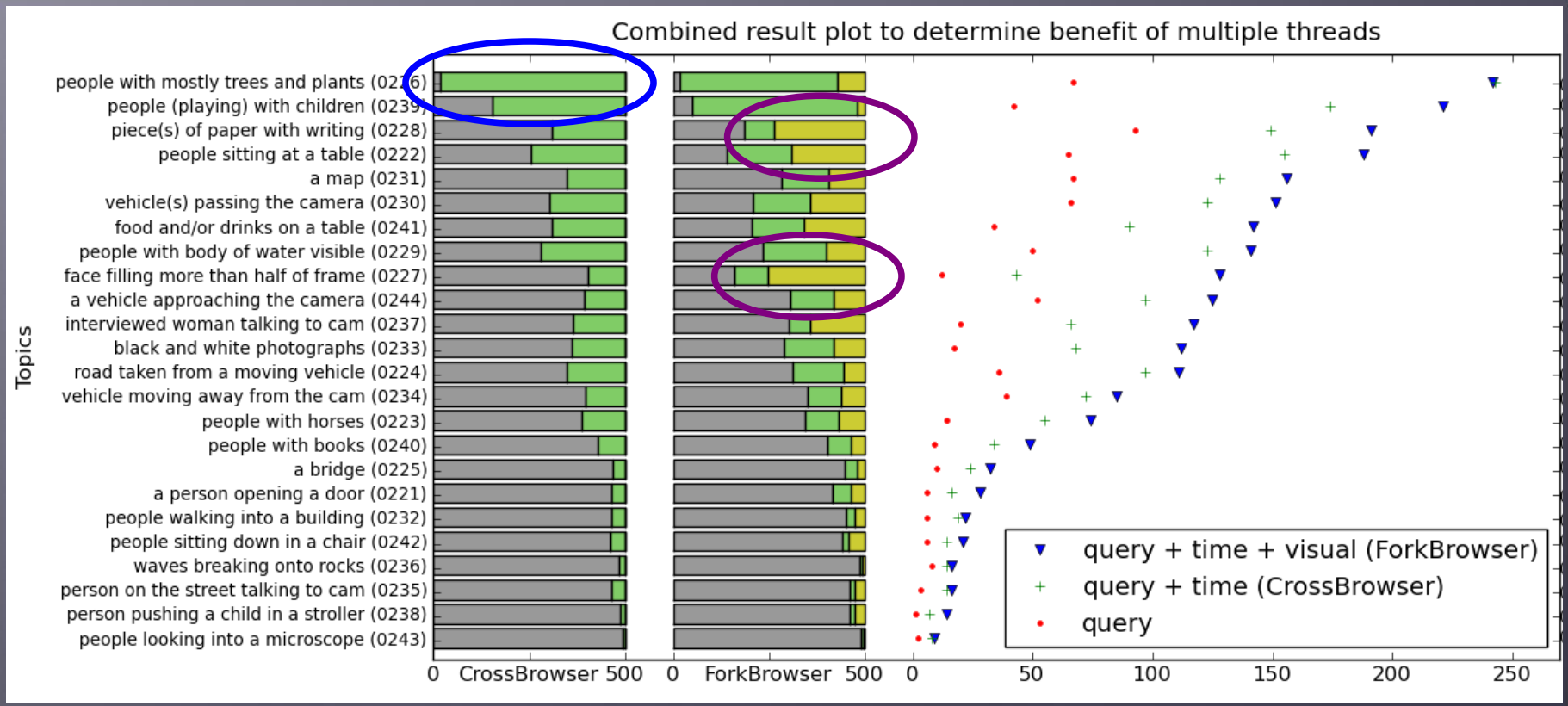
Experiment A

- ***What is the benefit of having multiple threads?***
- Measure
  - retrieval performance vs number of shown threads
  - number of positives after 500 actions, repeat for:



- query only
- query + time (CrossBrowser)
- query + time + visual similarity (ForkBrowser)

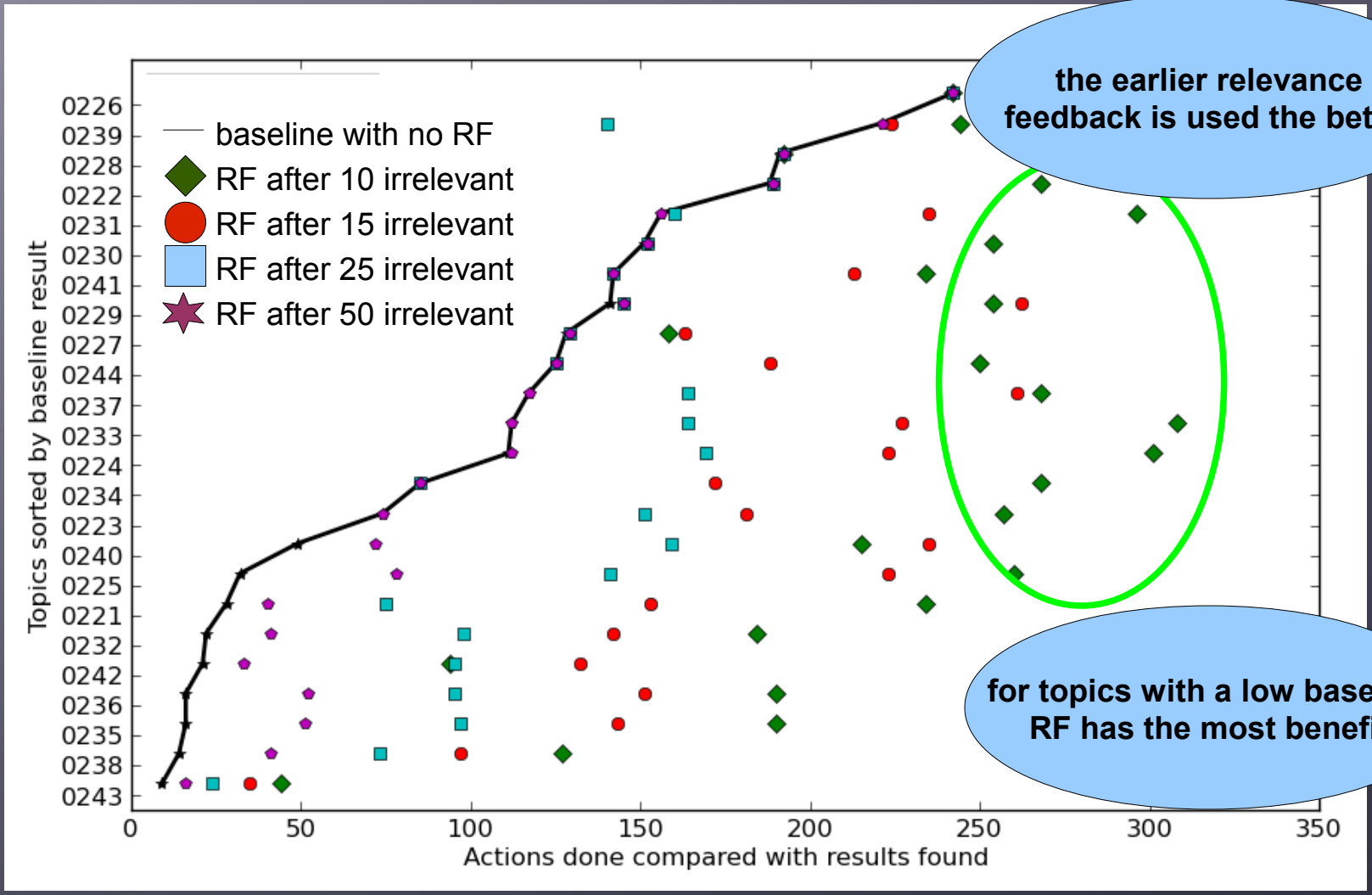
Experiment A



Experiment B

- ***When should a user switch to relevance feedback results?***
- Measured
  - optimal # of actions without results before using relevance feedback

Experiment B



TRECVID 2009 results

concept detectors
visual threads
relevance feedback

Conclusions

- Results indicate:
  - showing multiple threads yield better performance
    - also increases the time to perceive results for real world humans
  - We found a inverse correlation between # of threads shown and importance of initial query
  - Relevance Feedback yields greatest benefit for topics which would otherwise have limited results.
- ForkBrowser Focus + Context browsing paradigm, together with good initial concepts, consistently performs well

Any questions?