An Efficient Index Structure for Large-scale Geo-tagged Video Databases

Ying Lu†, Cyrus Shahabi†, Seon Ho Kim†

†Integrated Media Systems Center, University of Southern California

MediaQ ✓

Introduction

- **Motivations:**
  - **Mobile videos are prevalent**
    - *YouTobe statistics: ~20% mobile videos, ~3hours/min upload*
  - **Increasingly geo-tagged**
  - **Spatial-temporal video queries are demanding, e.g.,**

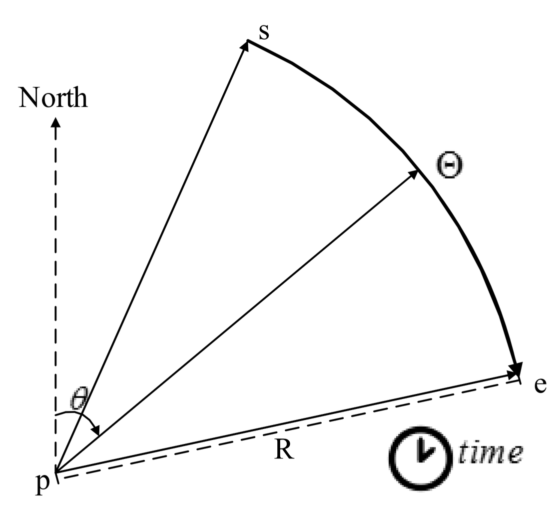
    "Find videos recorded in front of Tommy Trojan on the 2013 USC-UCLA football game day"

- **Challenge:**
  - **The fusion of location, time and direction**
  - **Existing indexes are not efficient**

- **How to efficiently index and search the large-scale videos by using the geo-metadata?**

Video Frame Model

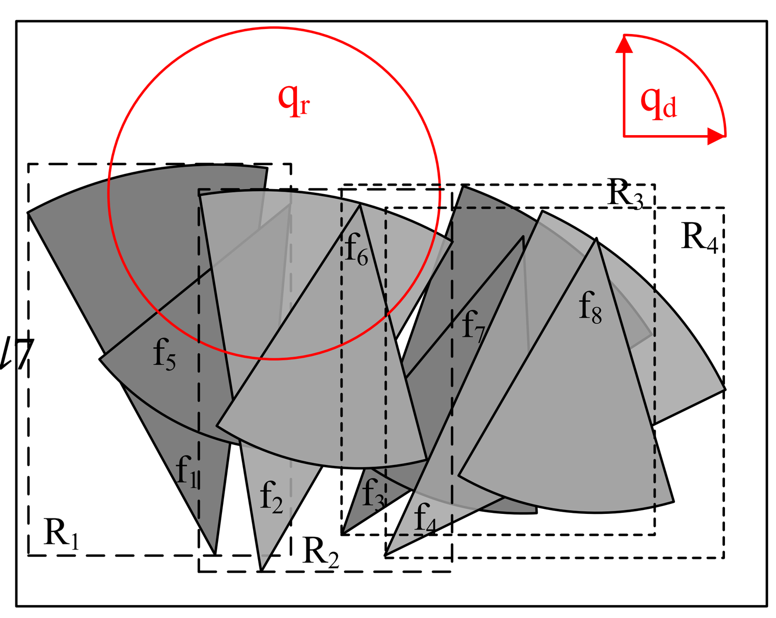
- **Model a video frame *f* in form of (*p*,*θ*, *R*, *time*)**
  - *p*: camera location
  - Θ: view orientation
  - *R*: maximum viewable distance
  - *time*: timestamp



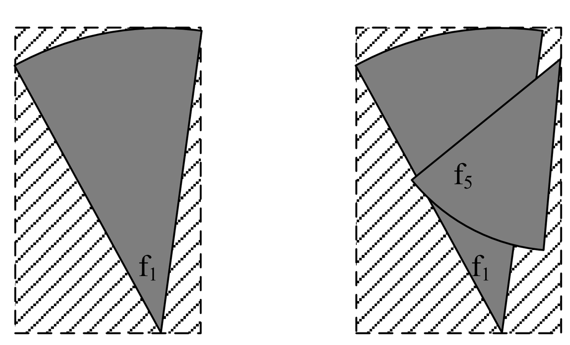
Field Of View (FOV) model

Existing Indexes

- **R-tree [1]**
  - **Range query q_r**
    - *results: f_1, f_2, f_5, f_6*
    - *visit: node#=4; FOV#=7*
    - *unnecessary visit: R_3, R_4, f_3, f_4, f_7*
  - **Directional query q_d**
    - *results: f_1, f_2, f_3, f_4*
    - *visit: node#=4; FOV#=8*
    - *unnecessary visit: f_5, f_6, f_7, f_8*
  - **Drawbacks**
    - *Large "dead space"*
    - *Large "overlap"* →
    - *No directional info in index nodes*
    - *Only based on area optimization criteria*

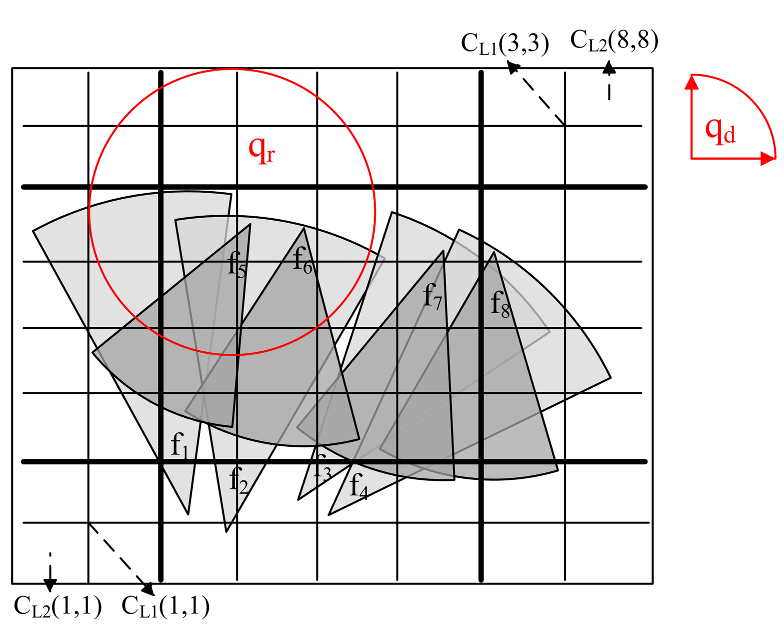


An R-tree indexing a set of FOVs



"Dead space" of a FOV and R-tree node

- **Grid based index [2]**
  - **Drawbacks**
    - *Need prior knowledge: cell size*
    - *Store direction info 3rd level only*
    - *Unnecessary visit: f_5, f_6, f_7, f_8*



C_{L1}

$C_{L1}(2,1)$	$f_1, f_2, f_3, f_4, f_7, f_8$
$C_{L1}(3,1)$	f_8
$C_{L1}(1,2)$	f_1, f_5
$C_{L1}(2,2)$	$f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8$
$C_{L1}(3,2)$	f_3, f_4, f_8

C_{L2}

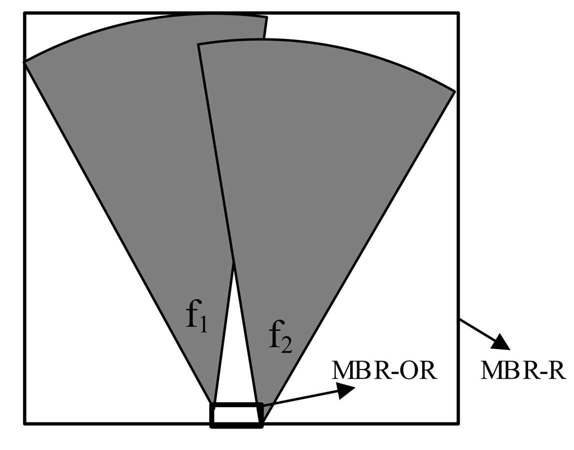
$C_{L2}(3,1)$	f_2
$C_{L1}(3,2)$	f_8
$C_{L1}(4,2)$	f_3
$C_{L1}(5,2)$	f_4
$C_{L1}(4,6)$	f_5, f_6
$C_{L1}(6,6)$	f_7
$C_{L1}(7,6)$	f_8

C_{L3}

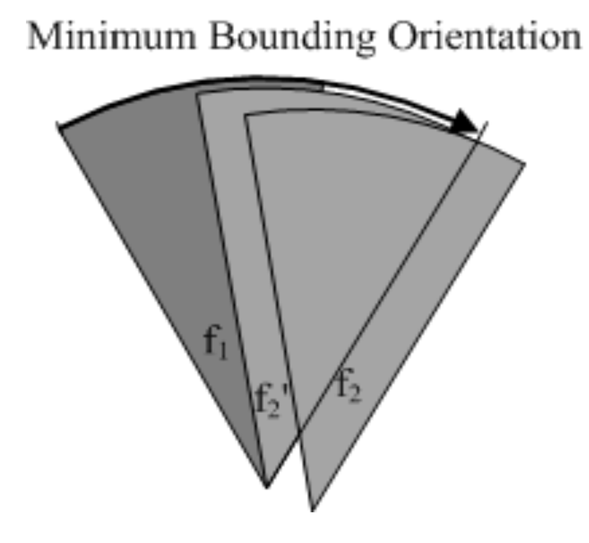
0° – 45°	f_1, f_2, f_3, f_4
46° – 90°	f_3, f_4
91° – 180°	f_6, f_7, f_8
181° – 225°	f_5, f_6, f_7, f_8
226° – 270°	f_5
315° – 360°	f_1, f_2

A New Index: Orientated R-tree (OR-tree)

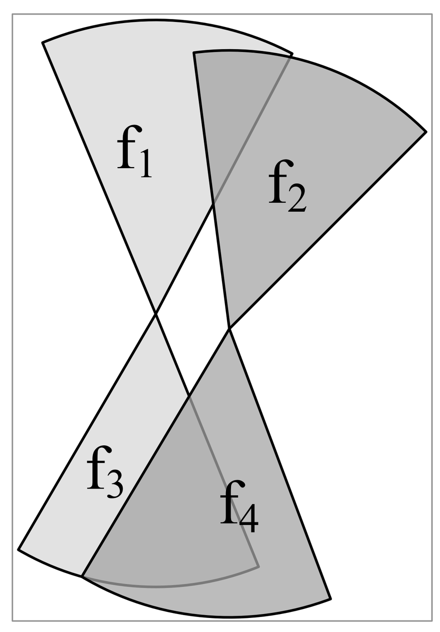
- Store smaller MBRs
- Incorporate orientations into internal index nodes
- Incorporate combined optimization of
  - *Area of camera locations*
  - ***O**rientation*
  - *View distance **R***



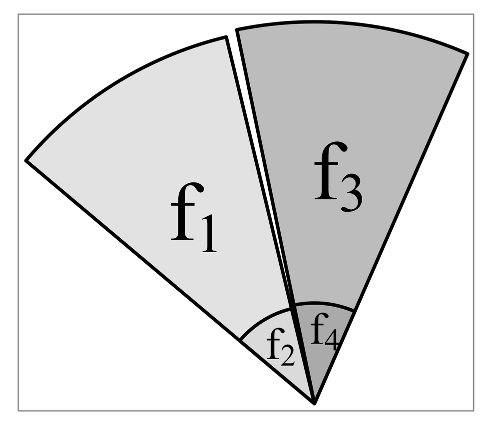
MBR-OR << MBR-R



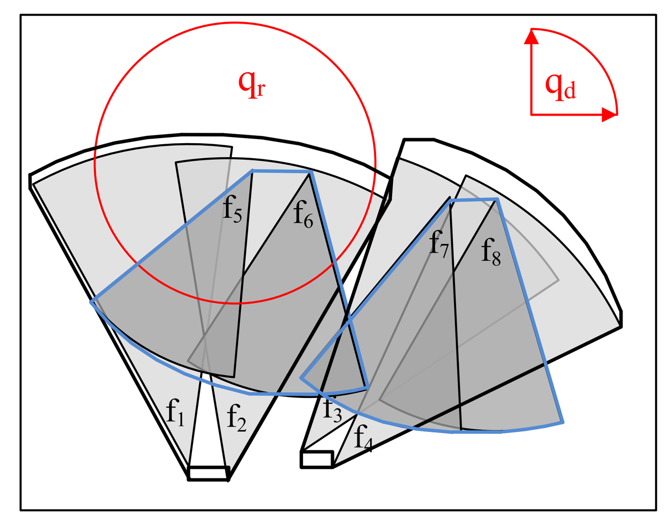
Store MBO in internal nodes



(a) Based on A



(b) Based on A+O



(c) Based on A+O+R

Optimization Criteria

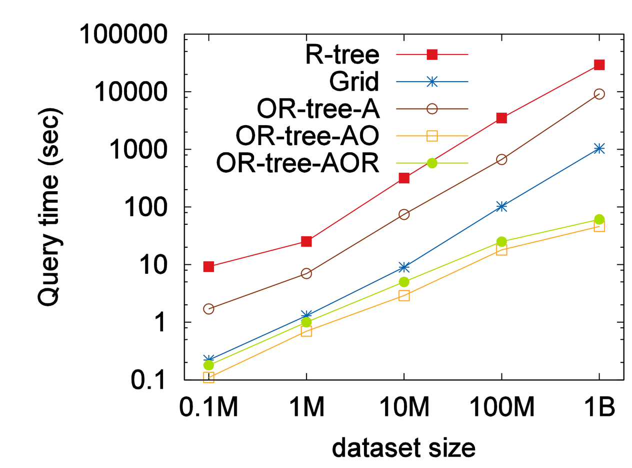
Experimental Results

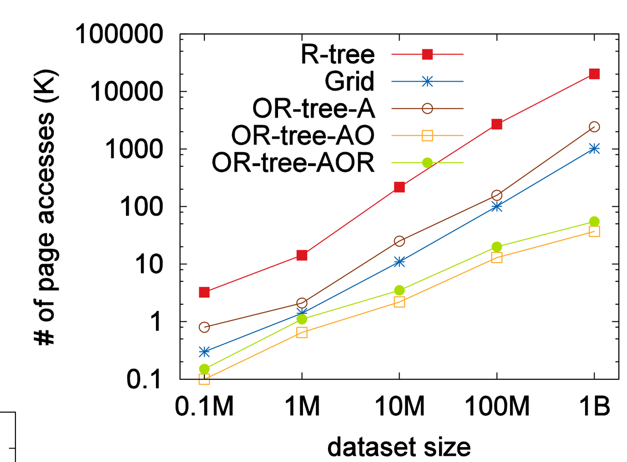
Table 2: Datasets for the experiments

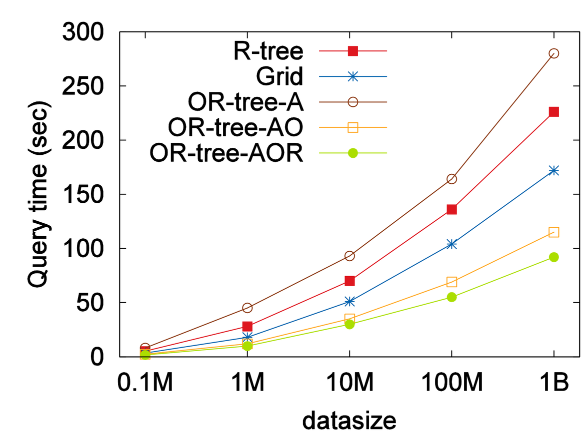
Statistics	*RW*	*Gen*
total # of FOVs	0.2M	0.1M∼1B
total # of videos	1276	100∼1M
FOV# per second	1	1
average time per videos	3mins	0.28hours
total time	62 hours	27.8hours∼31.71years
average camera moving speed (meters/s)	1.25	7.50
average camera rotation speed (degrees/s)	30	30
maximum viewable distance R (meters)	100	250
average viewable cover angle α (degrees)	51	51

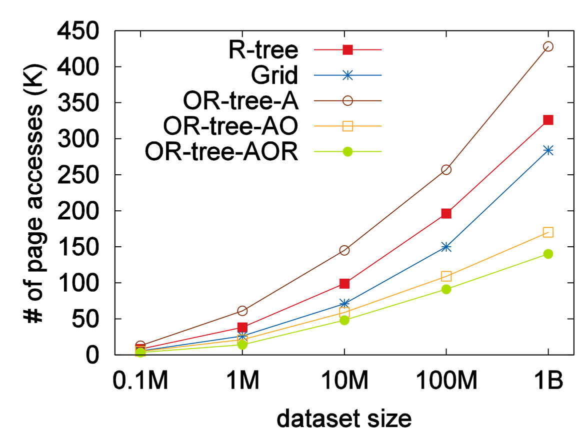
Table 3: Generated Dataset

total FOV#	0.1M	1M	10M	100M	1B
total video#	100	1K	10K	100K	1M
total time	27.8hours	11.6days	115.7days	3.2years	31.7years









Related Work

1. Antonin Guttman. R-Trees: a dynamic index structure for spatial searching. In ACM SIGMOD, pages 47-57, 1984.
2. He Ma, Sakire Arslan Ay, Roger Zimmermann, and Seon Ho Kim. Large-scale Geo-tagged video indexing and queries. In GeoInformatica, pages 0362-5915, 2013.

Conclusion and Future Work

- We proposed a new index called OR-tree to index FOVs for efficient video search.
- Our future direction is to index video considering time of videos
- Highly frequent update also need to be supported.

USC Viterbi School of Engineering
National Institute of Justice — The Research, Development, and Evaluation Agency of the U.S. Department of Justice
NORTHROP GRUMMAN
Google
IMSC Integrated Media Systems Center