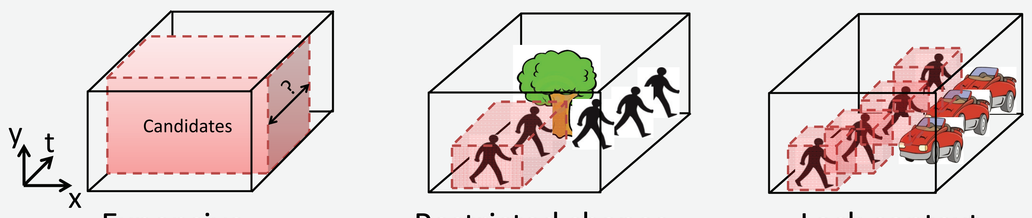
Efficient Activity Detection with Max-Subgraph Search

Chao-Yeh Chen and Kristen Grauman
The University of Texas at Austin

Problem

How to **detect** human activity in **continuous video**?

Status quo approaches:



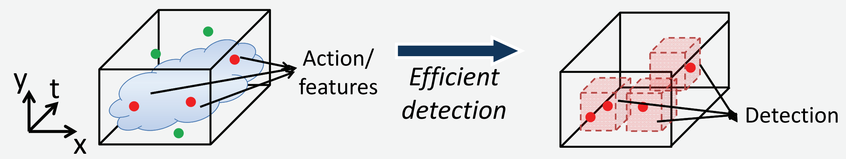
Expensive
Sliding window search

Restricted shapes
Cuboid detection

Lack context
Tracking humans

Main Idea

We pose activity detection as a *maximum-weight connected subgraph problem* over a learned space-time graph.



Highlights:

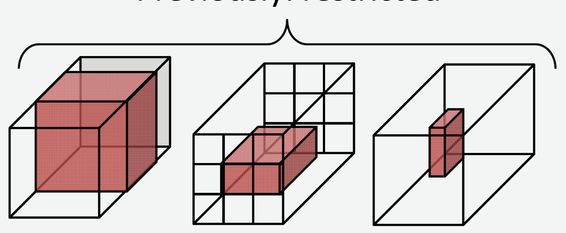
- Obtain exhaustive sliding window search result with much less time.
- Widen search scope to "non-cubic" detection volumes.
- Incorporate top-down knowledge of interactions of people/objects.

Related Work

- Sliding window over entire sequence
  e.g. [Ye et al. 2005, Duchenne et al. 2009, Satkin et al. 2010]
- Identify the detection subvolume with branch and bound method.
  e.g. [Yuan et al. 2009, Yu et al. 2011, Cao et al. 2010]
- Detection by tracking people, body parts, and nearby objects
  e.g. [Moore et al. 1999, Ramanan et al. 2003, Klaser et al. 2010]

Possible subvolume types

Previously: restricted



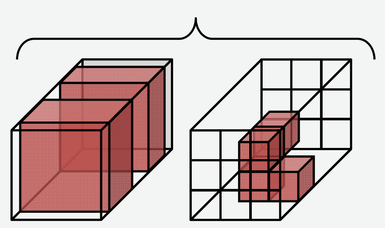
T-Sliding *or* T-Subgraph

ST-Cube-Sliding

ST-Cube-Subvolume
[Yuan et al. 2009]

VS

Now: broader



T-Jump-Subgraph

ST-Subgraph

Classifier training for feature weights

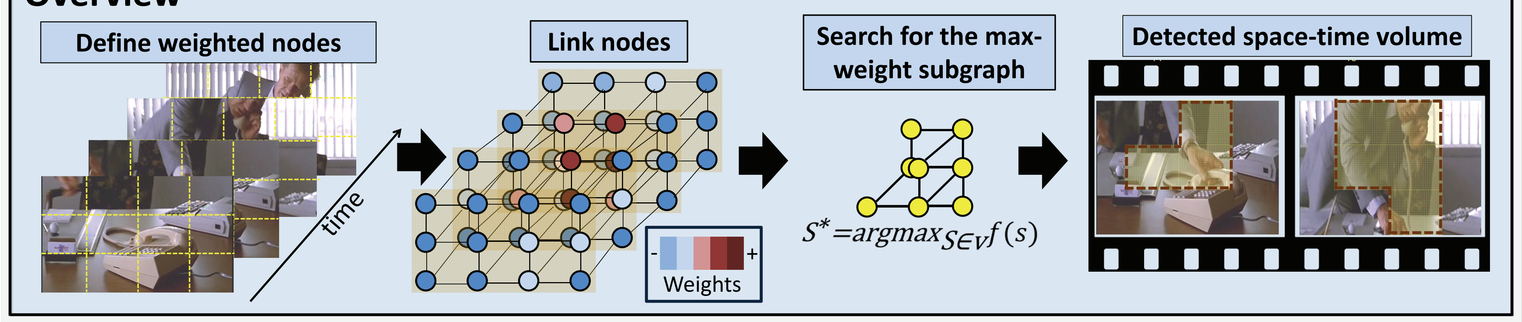
Classifier must be defined such that features computed within local space-time region *S* can be combined *additively.* Suitable classifiers include boosting, Naïve Bayes, or linear SVM:

Candidate subvolume; SVM weight for j^{th} word; Num occurrences of j^{th} word; SVM weight for i^{th} feature's word

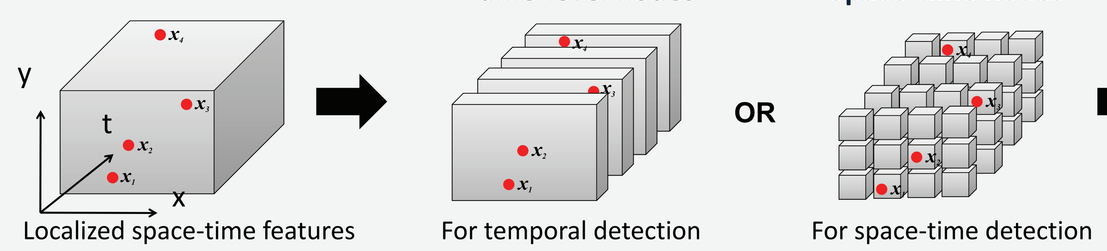
$$f(S) = \beta + \sum_{j=1}^{\#words} \omega^j h^j(S) = \beta + \sum_{i=1}^{\#features\ in\ S} \omega^{c_i}$$

Approach

Overview



Define weighted nodes

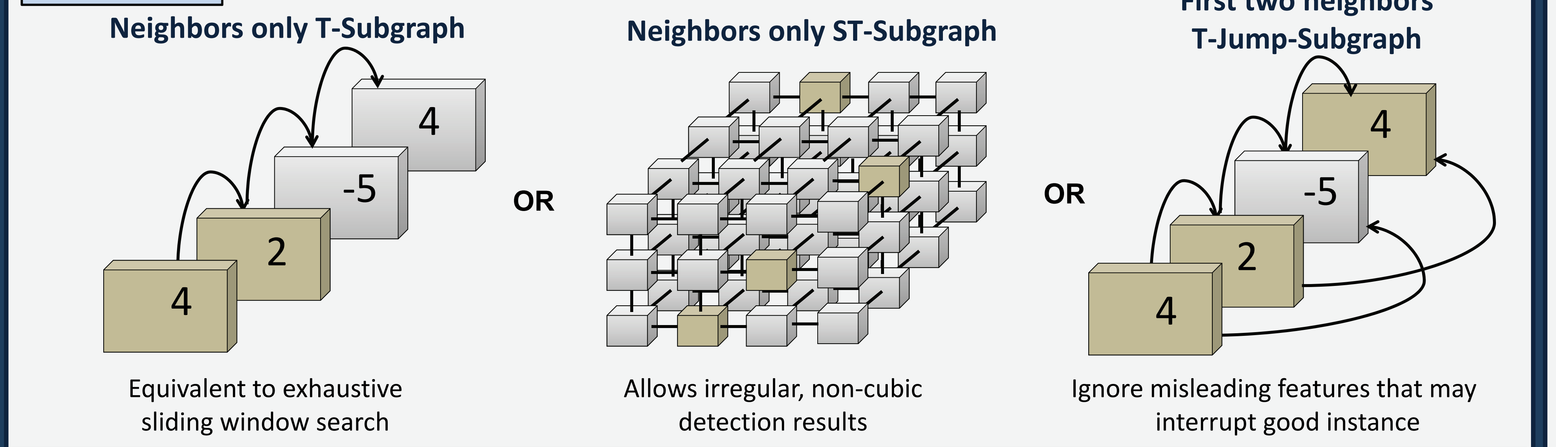


Localized space-time features

For temporal detection

For space-time detection

Link nodes



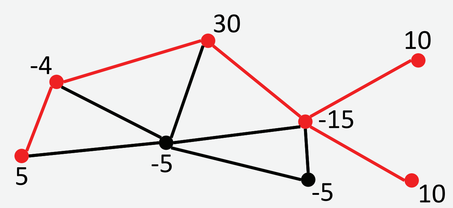
Equivalent to exhaustive sliding window search

Allows irregular, non-cubic detection results

Ignore misleading features that may interrupt good instance

Search for the max-weight subgraph

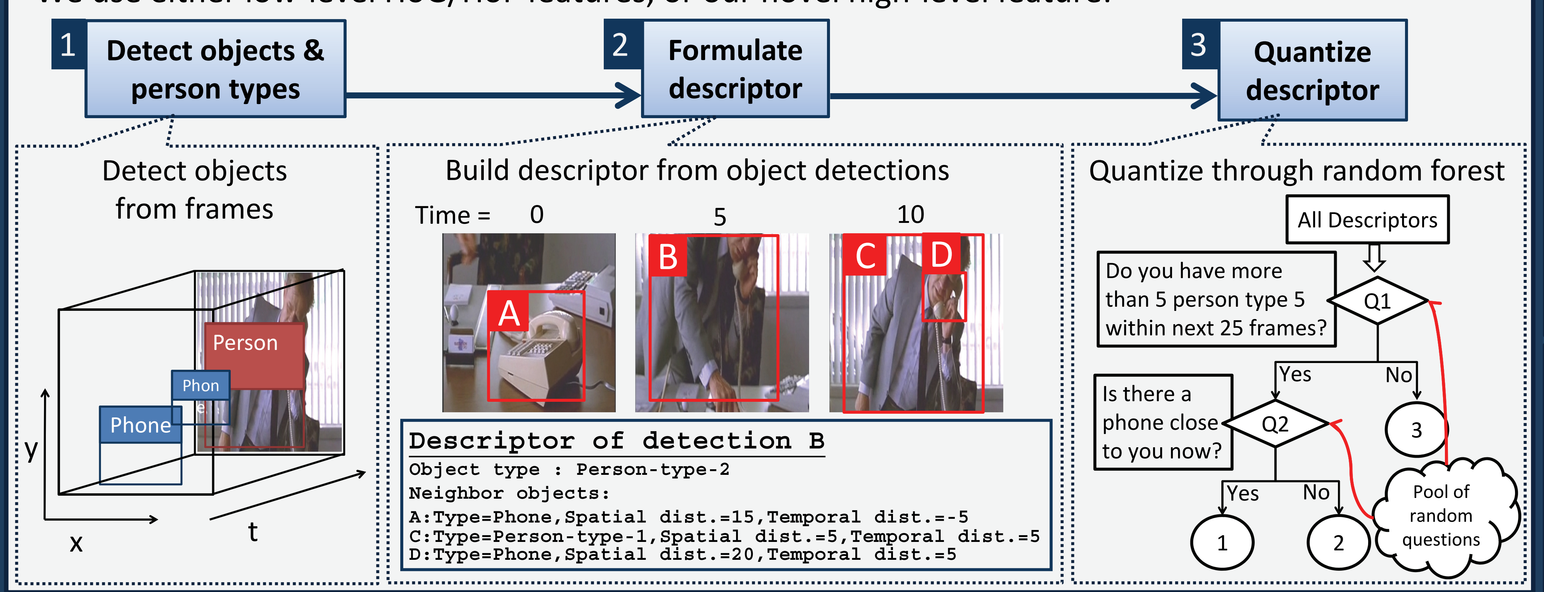
- Transform max-weight subgraph problem into a *prize-collecting Steiner tree* problem.
- Solve efficiently with branch and cut method from [Ljubic et al. 2006].



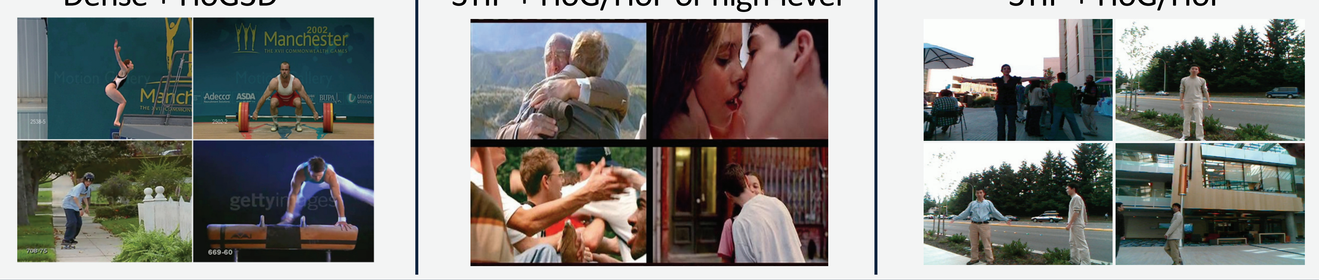
Red = max-weight subgraph

Localized Space-Time Features

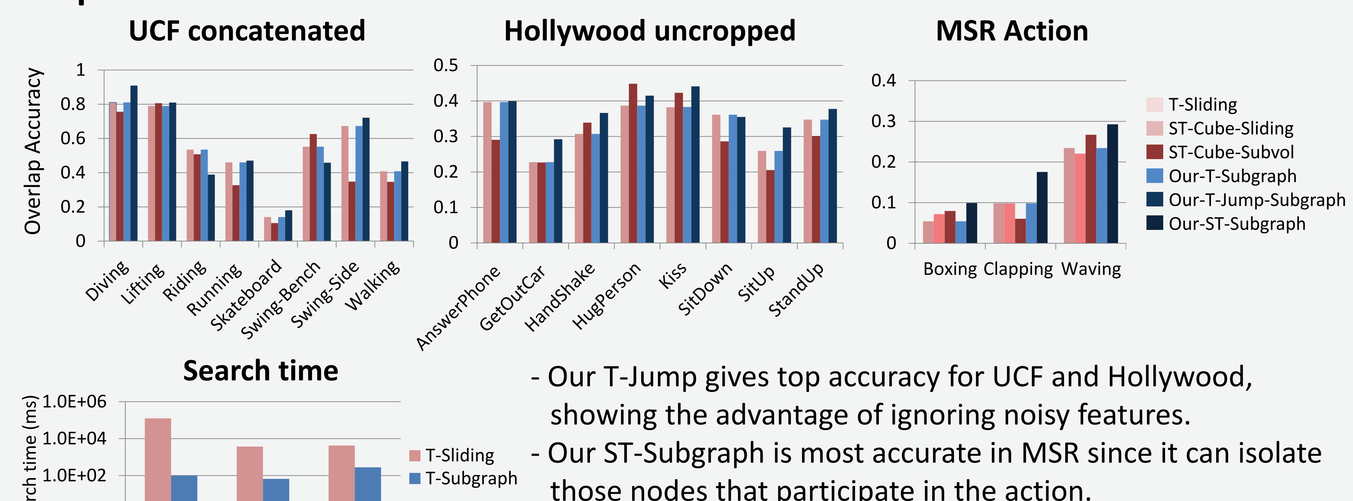
We use either low-level HoG/HoF features, or our novel high-level feature:

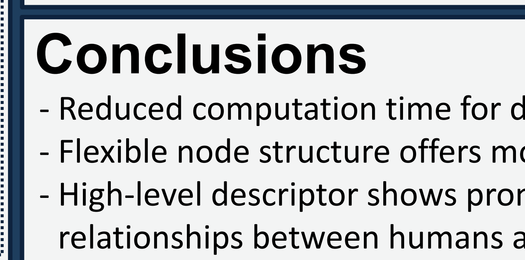


Results



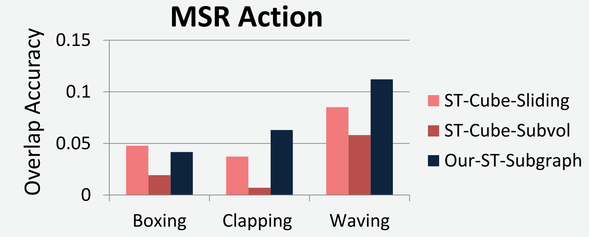
Temporal detection

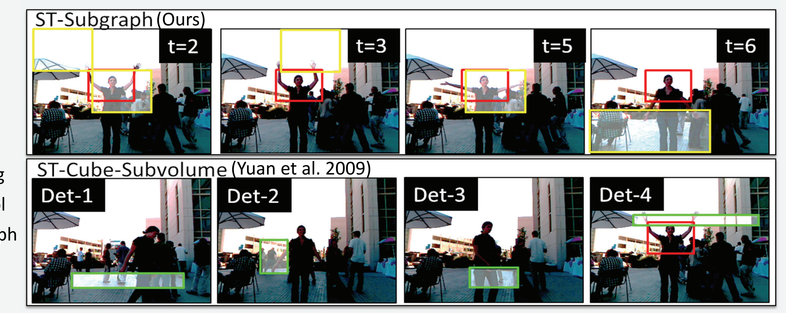




- Our T-Jump gives top accuracy for UCF and Hollywood, showing the advantage of ignoring noisy features.
- Our ST-Subgraph is most accurate in MSR since it can isolate those nodes that participate in the action.
- Our T-Subgraph is orders of mag. faster than T-Sliding.

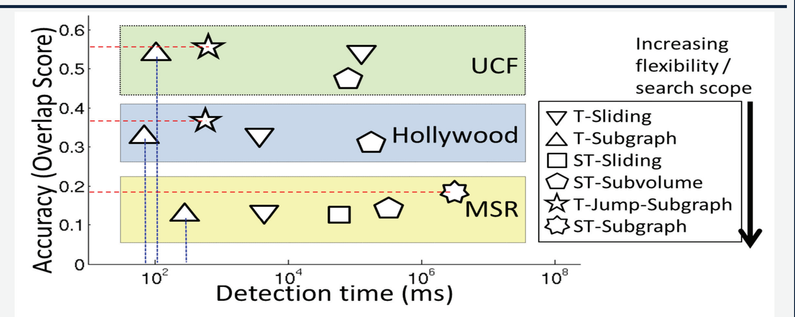
Space-time detection





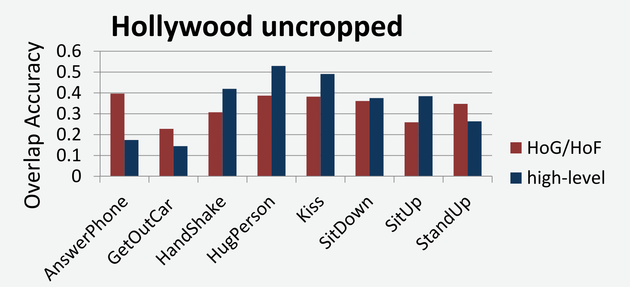
Speed and accuracy trade off

- Increased search scope boosts accuracy, though costs more.
- Flexibility of proposed method allows the best speed/accuracy.



Subgraph search with high-level features

T-Subgraph detection on Hollywood using proposed high-level features



Conclusions

- Reduced computation time for detection vs. sliding window search.
- Flexible node structure offers more robust detection in noisy backgrounds.
- High-level descriptor shows promise for complex activities by incorporating semantic relationships between humans and objects in video.