Leveraging High-Level Expectations for Activity Recognition

David Minnen, Irfan Essa, Thad Starner
GVU Center / Georgia Institute of Technology
{dminn,irfan,thad}@cc.gatech.edu
http://www.cc.gatech.edu/cpl/projects/expectationGrammars

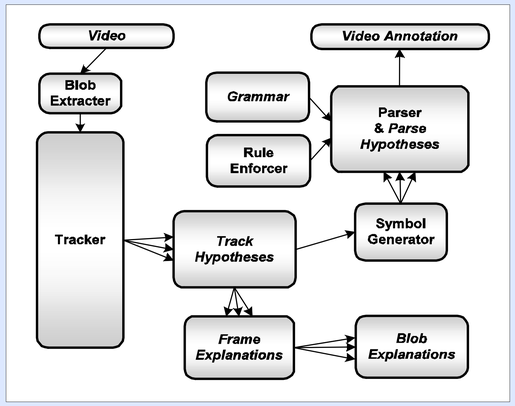
Overview & Goals

Automatically detect and decompose an activity from video

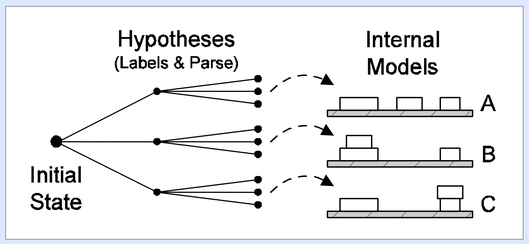
- Activity manually specified as a parameterized stochastic grammar
- Feedback from high-level parse influences interpretation of object identities
- Experiments use the Towers of Hanoi task to focus on activity recognition rather than visual appearance modeling

Approach

- Multiple track hypotheses maintained for each possible blob-object mapping
- Parse driven by blob interaction events
  - Domain-general events are detected (e.g., *merge, split, enter, exit*)
  - Events transformed into activity-specific terminals of the grammar
- Abstract model used to detect and prune semantically inconsistent interpretations
- Likelihood of parse computed from production and symbol probabilities
- Activity recognized by selecting the *most likely, semantically consistent hypothesis*



Data Flow



Recognition of Towers of Hanoi with Occlusion

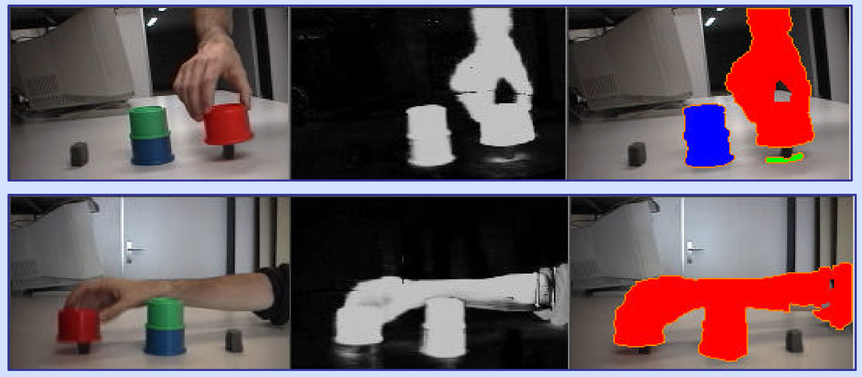
Limitations & Future Work

- Difficult to represent complex temporal constraints such as concurrency and quantitative temporal relationships
- Inefficient deletion error recovery (exponential growth)
- Requires activities with strong temporal relationships to disambiguate object identities and actions without relying on appearance models
- Comparison with other temporal models (e.g., n-gram models, HMMs, DBNs)

Stochastic Grammar for Towers of Hanoi Task

```
ToH -> Setup, enter(hand), Solve, exit(hand);
Setup -> TowerPlaced, exit(hand);
TowerPlaced -> enter(hand, block_A, block_B, block_C),
                  Put_1(block_A, block_B, block_C);
Solve -> state(Tower = TowerStart), MakeMoves, state(Tower = TowerGoal);
MakeMoves -> Move(block) [0.1]  |  Move(block), MakeMoves [0.9];

Move -> Move_1-2 | Move_1-3 |  Move_2-1 | Move_2-3  |  Move_3-1 | Move_3-2;

Move_1-2 -> Grab_1, Put_2;  Move_1-3 -> Grab_1, Put_3;
Move_2-1 -> Grab_2, Put_1;  Move_2-3 -> Grab_2, Put_3;
Move_3-1 -> Grab_3, Put_1;  Move_3-2 -> Grab_3, Put_2;

Grab_1 -> touch_1, remove_1(hand, ~)  |  touch_1(~), remove_last_1(~);
Grab_2 -> touch_2, remove_2(hand, ~)  |  touch_2(~), remove_last_2(~);
Grab_3 -> touch_3, remove_3(hand, ~)  |  touch_3(~), remove_last_3(~);

Put_1 -> release_1(~) | touch_1, release_1;
Put_2 -> release_2(~) | touch_2, release_2;
Put_3 -> release_3(~) | touch_3, release_3;
```

Identifying Objects by Action Instead of Appearance



Partial Parse of ToH Task

```
[113-739 (1-29) ToH . ]
 [113-739 (1-29) Setup, Solve, exit(hand) . ]
  [113-234 (1-4) TowerPlaced, exit(hand), enter(hand) . ]
   [113-169 (1-2) enter(hand, red, green, blue), Put_1(red, green, blue) . ]
    [169-169 (2-2) release_1 . ]
  [248-733 (5-28) state(Tower=TowerStart), MakeMoves, state(Tower=TowerGoal) . ]
   [248-733 (5-28) Move(block), MakeMoves . ]
    [248-315 (5-7) Move_1-3 . ]
     [248-315 (5-7) Grab_1, Put_3 . ]
      [248-270 (5-6) touch_1, remove_1(hand) . ]
      [315-315 (7-7) release_3 . ]
    [337-733 (8-28) Move(block), MakeMoves . ]
     [337-385 (8-10) Move_1-2 . ]
      [337-385 (8-10) Grab_1, Put_2 . ]
       [337-350 (8-9) touch_1, remove_1(hand) . ]
       [385-385 (10-10) release_2 . ]
          ...
```