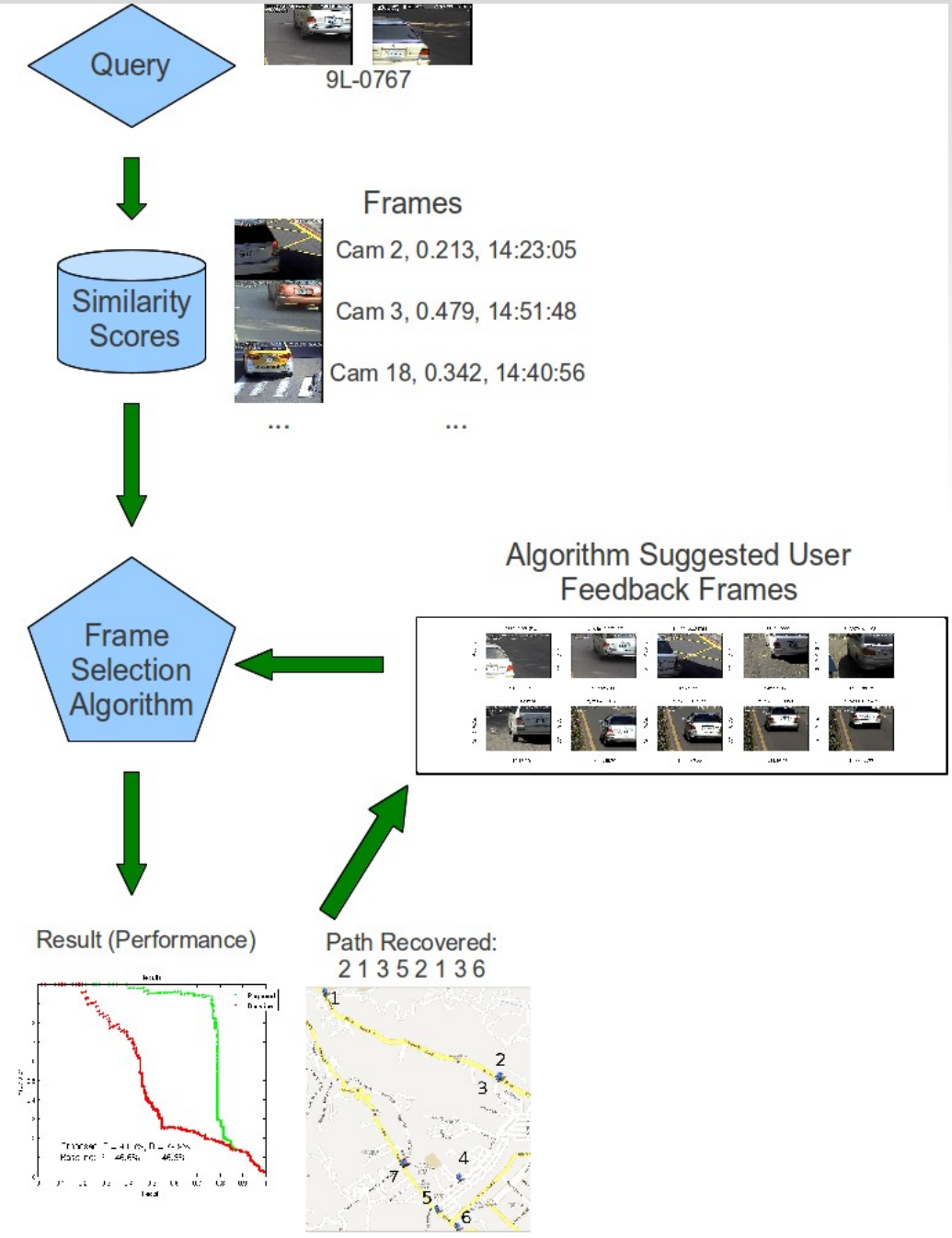
WAVS Presentation

Henry Shu
Feb 15, 2011

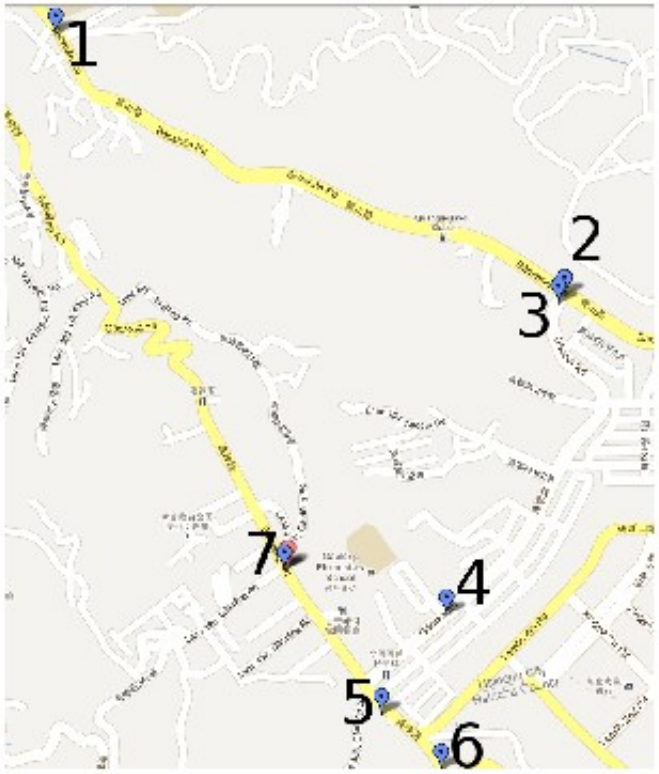
The Big Picture



Baseline Algorithm

- Select the top m most similar frames
  - Does not respect spatial-temporal realities

True: 2 1 3 5 2 1 3 6



Alg: 2 1 2 1 2 1 3 5 2 1 3 1 3 1 3 6

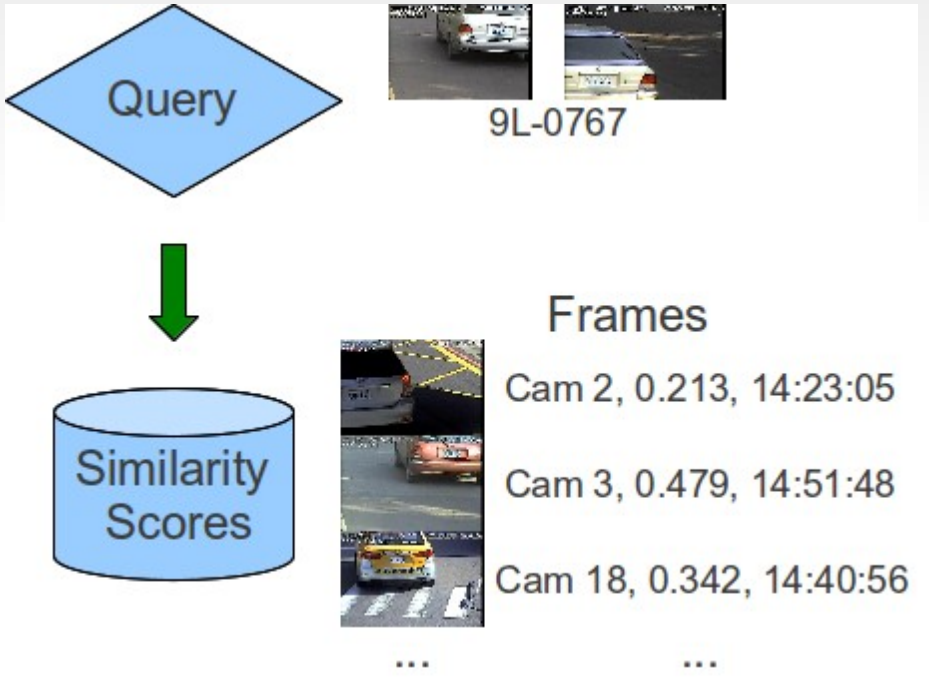
Related Works

- 3D camera site model around cameras
  - T. Kanade, 2001
- Camera transition probabilities
  - R. Zabih, CVPR 1999
- Path cover problem in a graph
  - M. Shah, ICCV 2003
- Content-based image retrieval
  - T. Kanade, ACM 2010

Our proposed approach...

Costs of In/Excluding Frames

- Similarity score s, $0 < s < 1$, 0 is similar
- $p = \exp(-\lambda s)$, $\lambda > 0$
- Including cost = -log p
- Excluding cost = -log(1 - p)



Speed Violation

2, 3005, 0.26168

GT: 0, ALG: 0



14:10:27.00

1, 1976, 0.25316

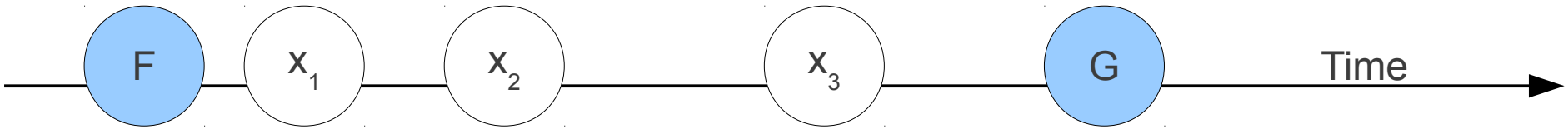
GT: 0, ALG: 0



14:10:28.03

Temporal Discontinuity

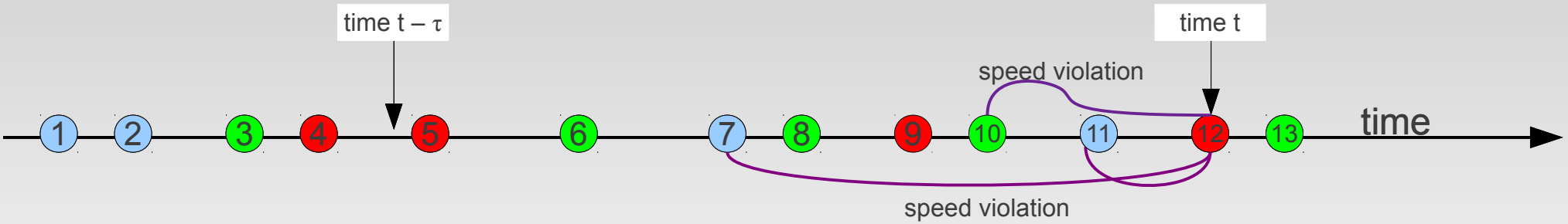
- Frames F and G selected
- time(G) – time(F) > τ
- No frame x temporally between F and G are selected



Our Proposed Approach

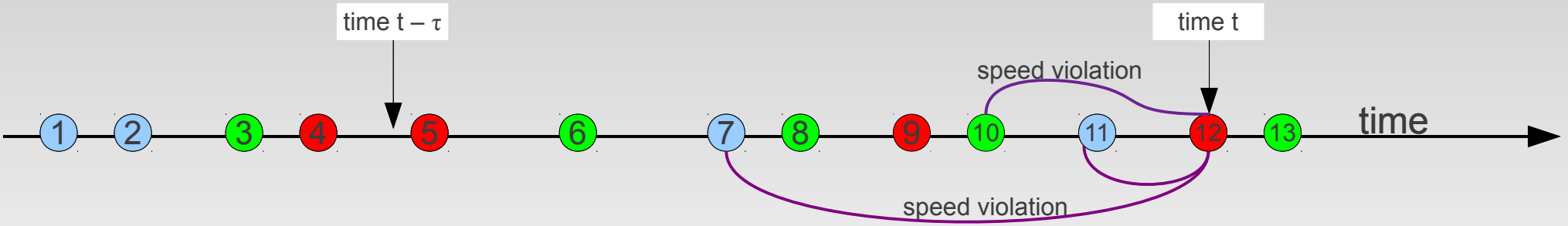
- Select frames with minimum cost, subject to:
  - Constraint 1: No two selected frames induce a **speed violation**
  - Constraint 2: The selected frames altogether induce R or less **temporal discontinuities**
- Developed an algorithm solving above **exactly** (global optimum)
  - < 10s with +6000 frames

Algorithm: Some Definitions



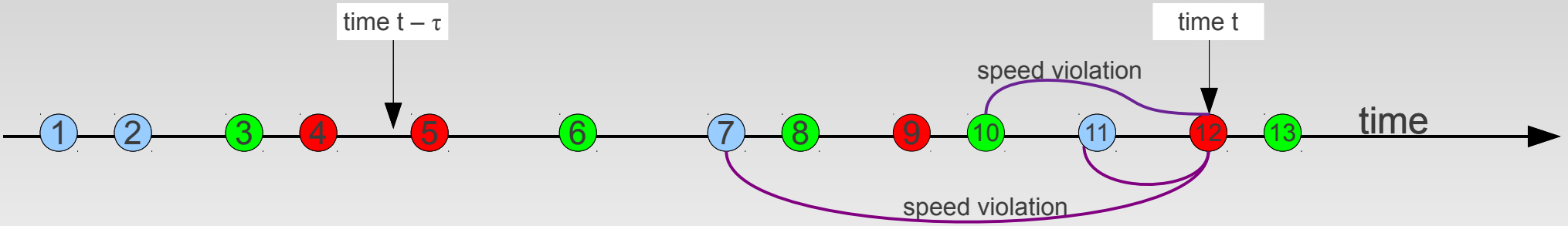
- Shown 13 frames (purple lines mean speed-violation frame pairs)
  - For simplicity, only speed violations relative to frame 12 are shown
- Frame color: Camera that took the frame
- Selecting (not selecting) frame i costs s_i (d_i)
- Definition: subp(u, k, c) means the subproblem in which we pretend that
  - there are only frames 1, 2, …, u
  - these u frames induce exactly k temporal discontinuities
  - the last selected frame (could be u itself or not) is from camera c
- Obviously, the original problem is $\text{best}_{k,c}$ subp(13, k, c)
- Definition: N(u, k) is the optimal cost of subp(u, k, cam of u) with the additional requirement that u is selected.

Algorithm: Optimal Cost



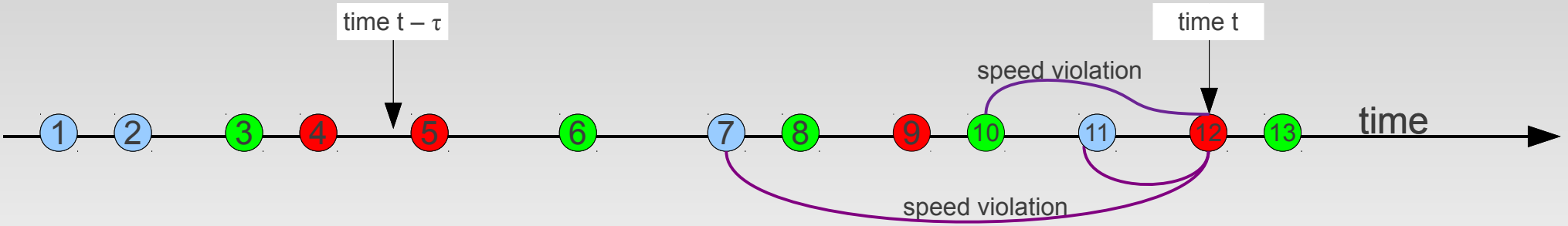
- Definition: Let M(u, k, c) be the optimal cost of subp(u, k, c).
- Example of u = 12:
  - M(12, k, blue) = M(11, k, blue) + d_{12}.
  - M(12, k, green) = M(11, k, green) + d_{12}.
  - M(12, k, red) = ?
    - Case Not selecting frame 12: M(11, k, red) + d_{12}.
    - Case Selecting frame 12: Two sub-cases
      - Sub-case Selecting frame 12 creates a temporal discontinuity (“Discont”)
      - Sub-case Selecting frame 12 does not create a temporal discontinuity (“Smooth”)

Algorithm: Optimal Cost



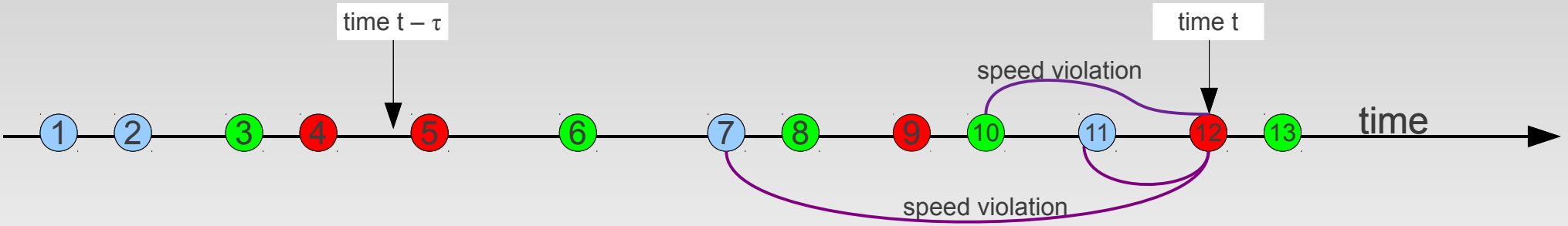
- ("Discont") Sub-case Selecting frame 12 creates a temporal discontinuity:
  - M(12, k, red) = minimum of...
    - $M(4, k - 1, \text{red}) + (d_5 + d_6 + \ldots + d_{11}) + s_{12}$
    - $M(3, k - 1, \text{green}) + (d_4 + d_5 + \ldots + d_{11}) + s_{12}$
    - $M(2, k - 1, \text{blue}) + (d_3 + d_4 + \ldots + d_{11}) + s_{12}$
  - Remember, M(2, k – 1, blue) does not necessarily mean frame 2 was selected. It just means "up to frame 2, the last selected frame being a blue one".

Algorithm: Optimal Cost



- ("Smooth") Sub-case Selecting frame 12 does not create a temporal discontinuity:
  - M(12, k, red) = minimum of...
    - $N(9, k) + (d_{10} + d_{11}) + s_{12}$
    - $N(8, k) + (d_{9} + d_{10} + d_{11}) + s_{12}$
    - $N(6, k) + (d_{7} + d_{8} + \ldots + d_{11}) + s_{12}$
    - $N(5, k) + (d_{6} + d_{7} + \ldots + d_{11}) + s_{12}$

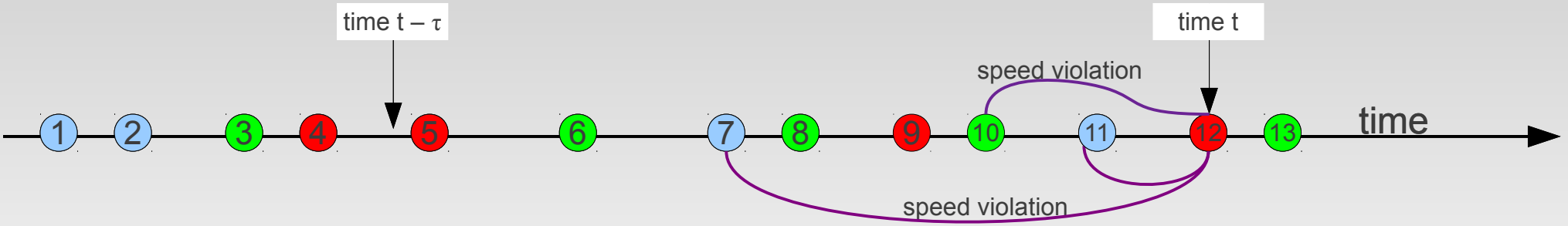
Algorithm: Optimal Cost



- Take the minimum of slides 11, 12, and 13 to get the final M(12, k, red).
- We have to compute N(12, k) as well. It might be used later, too.
  - N(12, k) is the minimum of M(12, k, red) from "Discont" (slide 12) and "Smooth" (slide 13).
- For all u and c, the base case k = 0 is $M(u, 0, c) = d_1 + d_2 + \ldots + d_u$.

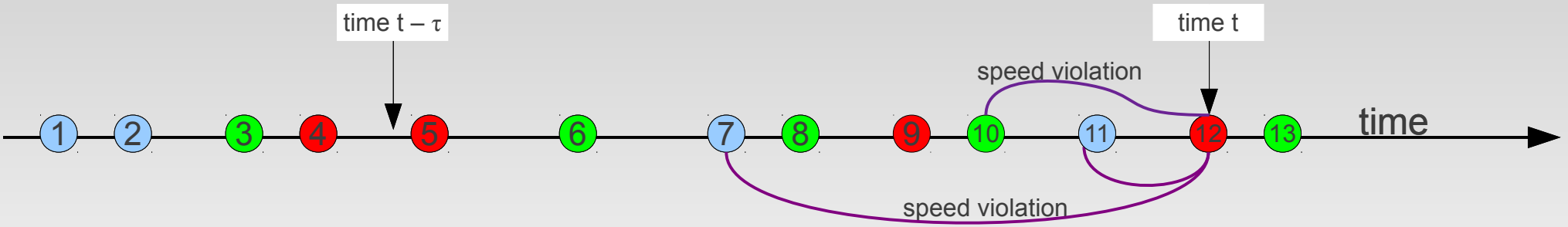
  That is, not selecting any frames.
- Now we have M(u, k, c) for all u, k, c, how can we recover the frames that are selected in $best_c$ subp(13, R, c)?

Algorithm: Some Definitions



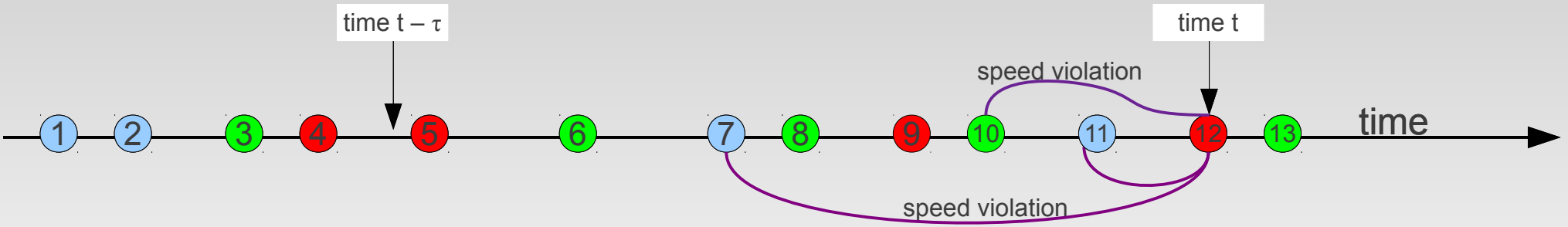
- Definition: T(u, k) is the last selected frame, could be u itself or not, of the solution of subp(u, k, color of u). This solution must induce exactly k temporal discontinuities.
- Definition: L(u, k) is the latest selected frame prior to u in a solution that selects u. This solution must induce exactly k temporal discontinuities.

Algorithm: Frame Selection



- Back to the example where u = 12, and compute T(12, k) and L(12, k)
- If M(12, k, red) came from not selecting frame 12 in slide 11:
  - T(12, k) = T(9, k)
- If M(12, k, red) came from selecting frame 12 in “Discont” or “Smooth”
  - T(12, k) = 12
- L(12, k) is one of 5, 6, 8, 9 (see slide 13) or one of T(2, k – 1), T(3, k – 1), T(4, k – 1) (see slide 12), depending on which has the smallest cost.

Algorithm: Frame Selection



- Now we can recover the selected frames from subp(13, k, c') (here c' = arg $\min_c$ M(13, k, c)) as follows
  - Set u ← T(the very last frame of color c', k)
  - Start with the current selected frame u.
  - The previous selected frame is L(u, k).
  - Update k to k – 1 if frames u and L(u, k) are more than τ apart in time.
  - Update u to L(u, k)
  - Continue until k becomes 0
- Note: We want to pick some k ≤ R (see slide 9) such that $\min_c$ M(13, k, c) is minimum.

No presentation material.

Demo

- Query: 9L-0767





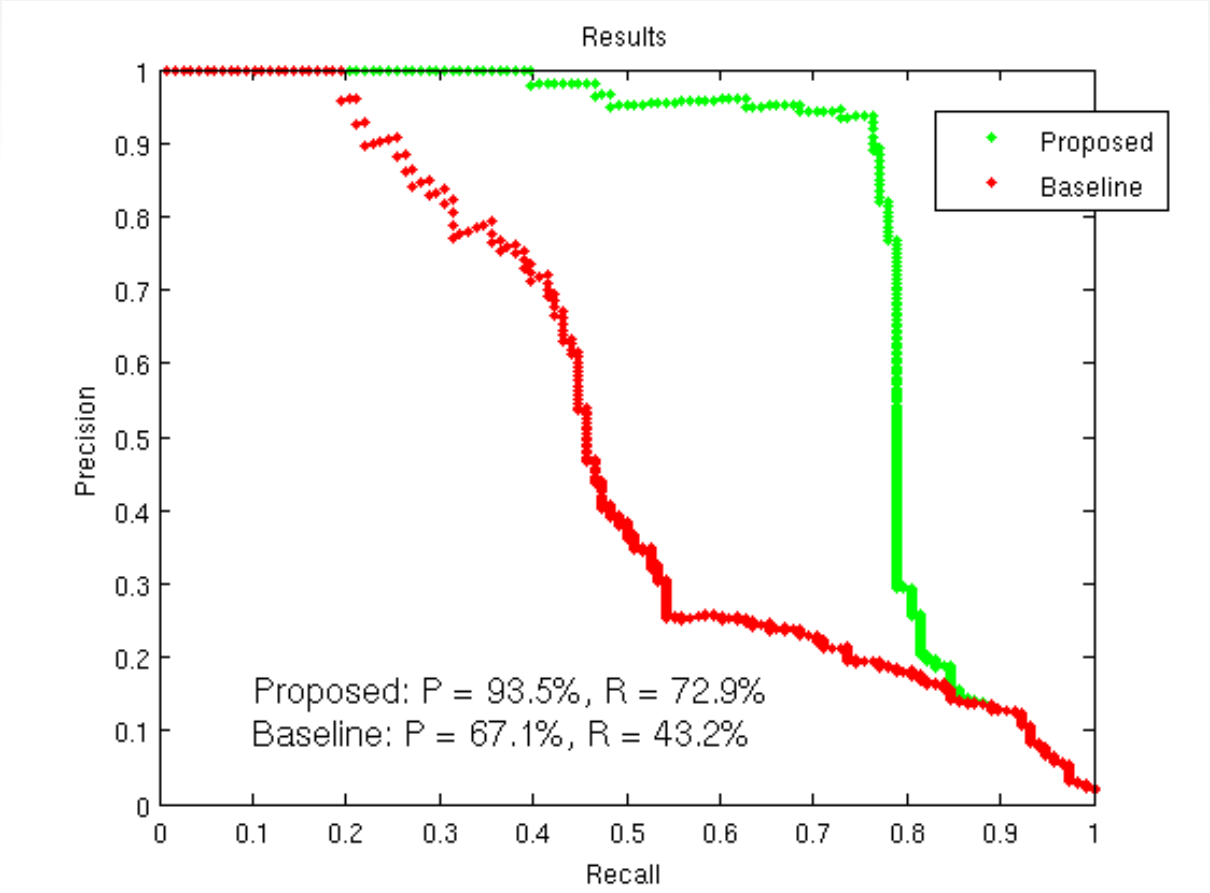






Result (No User Feedback)

- Performance metric
  - Precision: # correct frames selected / # frames selected
  - Recall: # correct frames selected / # truly villain's car frames



PP Path (No User Feedback)

2, 2986, 0.17932, 14:10:16.13



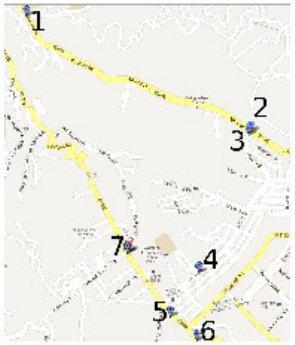
Cam, Frame ID, Sim Score, Time

2, 3000, 0.13838, 14:10:16.63



Cam, Frame ID, Sim Score, Time

True: 2 1 3 5 2 1 3 6



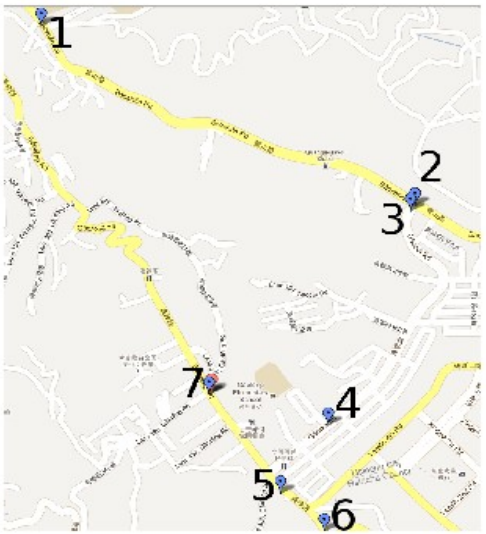
Note that the recovered path is exactly correct:

Alg: 2 1 3 5 2 1 3 6

BL Path (No User Feedback)



True: 2 1 3 5 2 1 3 6



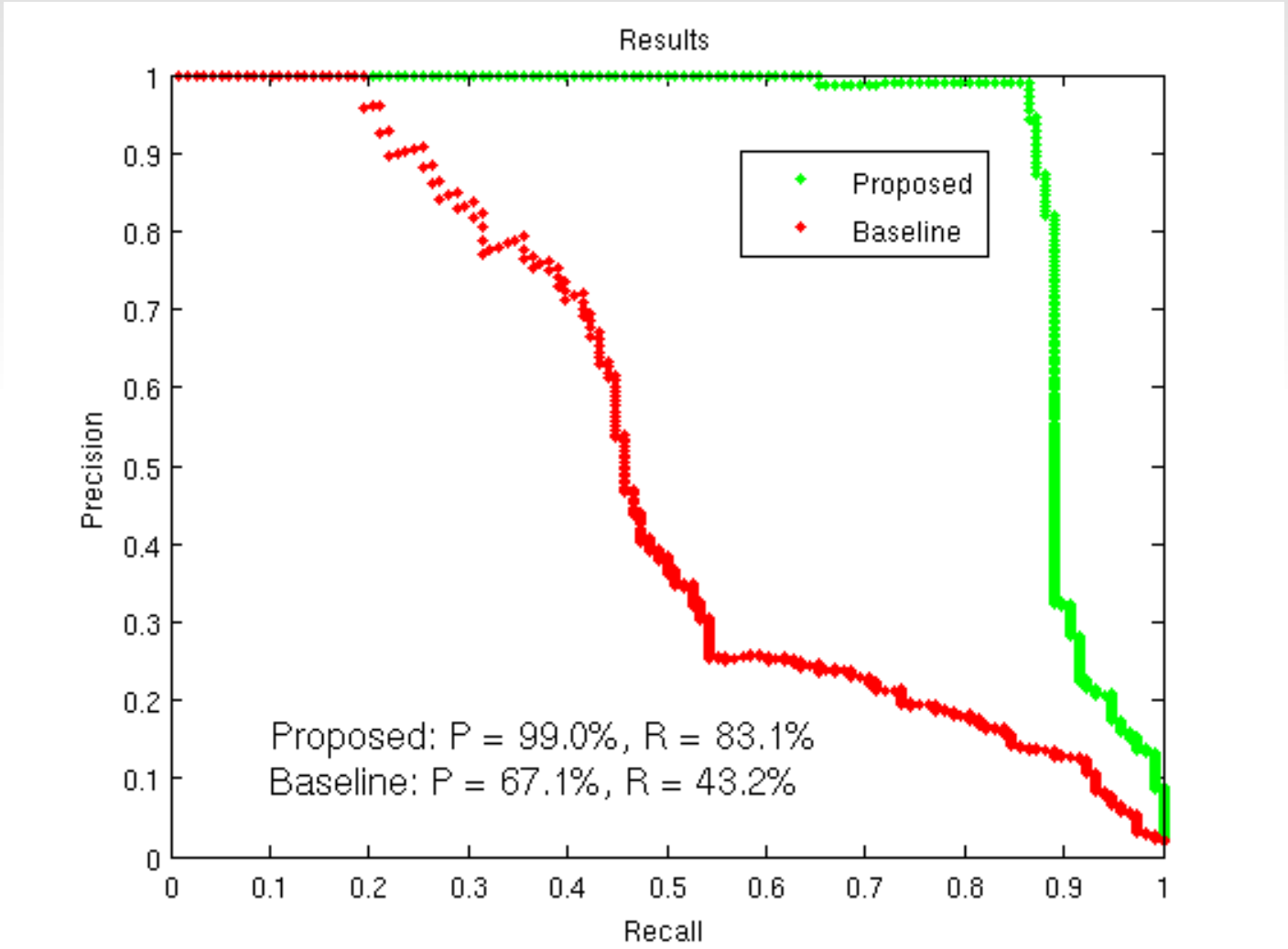
Alg: 2 1 2 1 2 1 3 5 2 1 3 1 3 1 3 6

Demo (Algorithm Suggested Feedbacks)

Of all the frames that the proposed algorithm did not select, these are the frames that the algorithm thinks might also be the villain's car.



Result (After 1 Feedback)



Conclusion

- Our algorithm can recover the path exactly right, without user feedbacks
- Our algorithm already greatly outperforms the baseline without user feedbacks
- Our algorithm can propose very relevant vehicles for the user to provide feedbacks.