Vision-based Road Safety Analysis

Nicolas Saunier, Tarek Sayed and Clark Lim

UBC Transportation Engineering Group

Bureau of Intelligent Transportation Systems and Freight Security (BITSAFS)

BITSAFS
Bureau of Intelligent Transportation Systems and Freight Security
Faculty of Applied Science

Outline

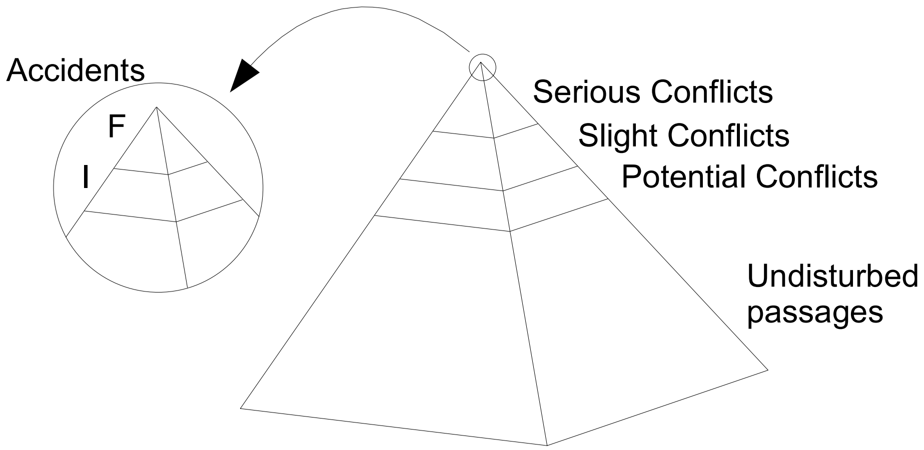
1. Road Safety in a Probabilistic Framework
2. Feature-based Vehicle Detection and Tracking
3. Learning Motion Patterns for Motion Prediction
4. Experimental Results in Road Safety

1. Road Safety

- Traditional road safety reactive approach, based on historical collision data.
- Pro-active approach: "Don't wait for accidents to happen".
- Need for surrogate safety measures that provide complementary information and are easy to collect (more frequent).
- Traffic conflicts (near-misses).

1. The Collision Probability

- The safety/severity hierarchy.
- For two interacting road users, there are various chain of events that can lead to a collision.

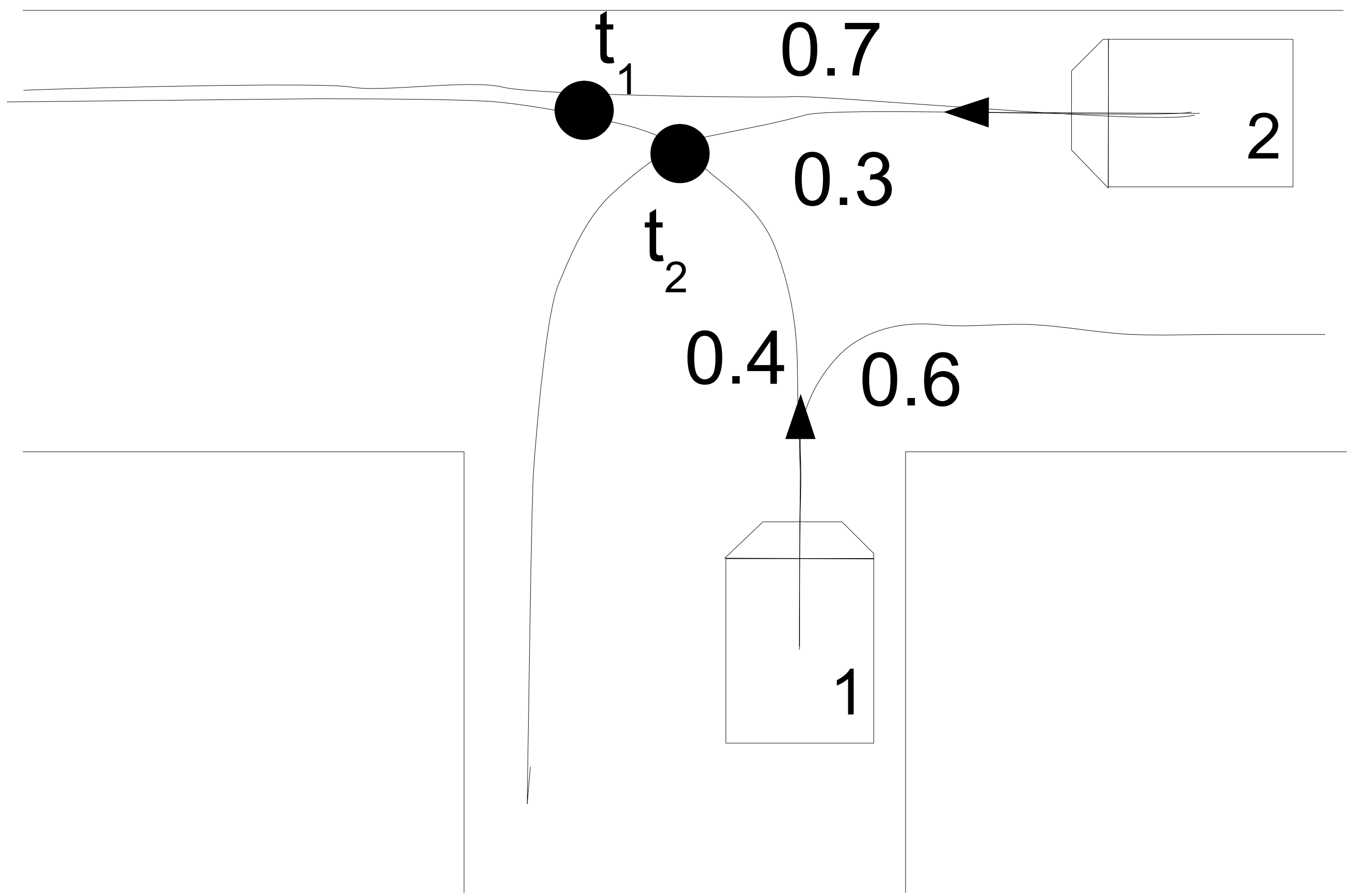


- Given extrapolation hypotheses for road users,

$$P(Collision(A_1, A_2)|H_i, H_j) = e^{-\frac{\Delta_{i,j}^2}{2\sigma^2}}$$

$$P(Collision(A_1, A_2)|Q_{1,t\leq t_0}, Q_{2,t\leq t_0}) = \sum_{i,j} P(H_i|Q_{1,t\leq t_0})P(H_j|Q_{2,t\leq t_0})\ e^{-\frac{\Delta_{i,j}^2}{2\sigma^2}}$$

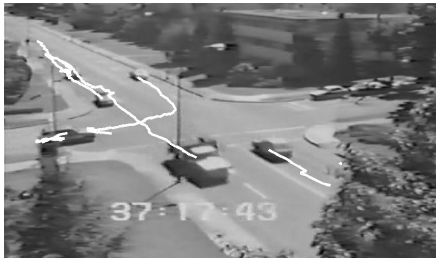
1. Simple Example



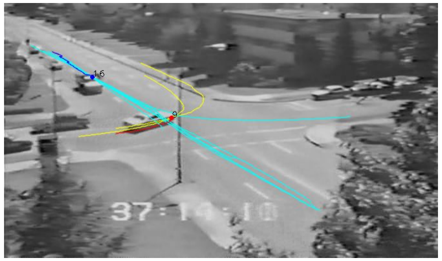
$$P(Collision) = 0.4 \times 0.7 \times e^{-\frac{(t_1 - t_0)^2}{2\sigma^2}} + 0.4 \times 0.3 \times e^{-\frac{(t_2 - t_0)^2}{2\sigma^2}}$$

1. A Modular System

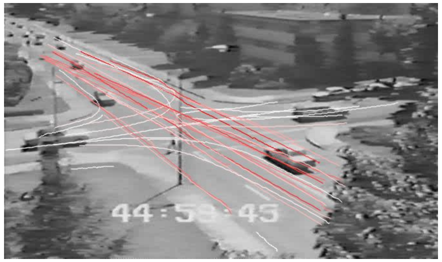
Image Sequence



Trajectory Database



Interaction Database



- Motion Patterns
- Volume, Origin-Destination Counts
- Driver Behavior...

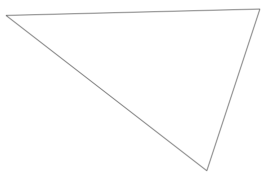
- Traffic Conflict Detection
- Exposure Measures
- Interacting Behavior...

Interpretation Modules

2. Vehicle Detection and Tracking

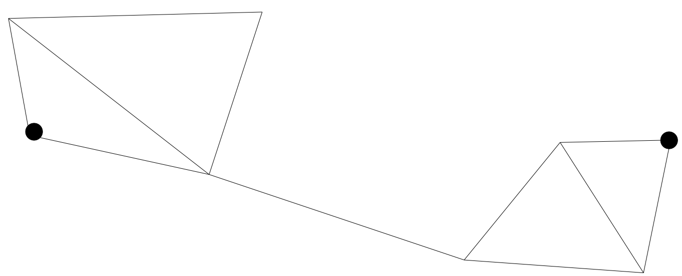
- Feature-based tracking was chosen since
  - it is the most readily available method (Kanade Lucas Tomasi tracker implemented by Stan Birchfield and in Intel OpenCV Library),
  - it is robust to partial occlusion, variable lighting conditions, and requires no special initialization.
- Simple constraints to filter out noise and irrelevant motion.
- Extension of the feature-based tracking algorithm by Beymer et al. (1997) to intersections (CRV 06).

2. Grouping Algorithm

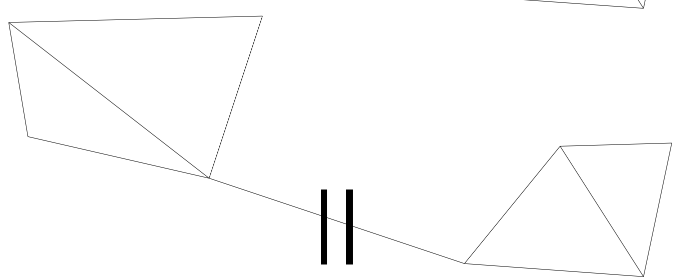


vertex: feature track
edge: grouping relationship
$d_{ij}(t)$: distance between feature i and j at time t

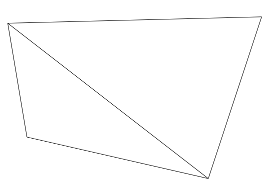
For each image at time t,

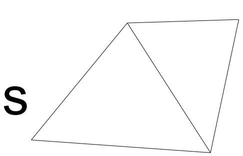


1. connection of recently detected features within distance $D_{connection}$



2. feature disconnection if relative motion is too large
$\max d_{ij}(t) - \min d_{ij}(t) > D_{segmentation}$



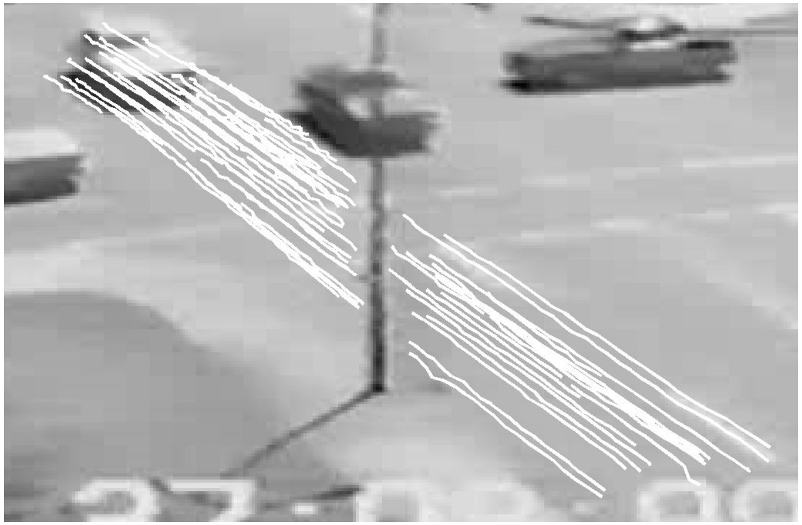


vehicle hypotheses

3. identification of the graph connected components and vehicle hypothesis generation if the features are not tracked

2. Feature-based Vehicle Detection and Tracking

- Accuracy between 84.7 % and 94.4 % on 3 sets of sequences.



Demo

3. Motion Patterns

- How to predict road users' future positions to compute the collision probability ?
- Road users do not move randomly. Typical road users movements, traffic motion patterns, can be learnt from the observation of traffic data.
- Incremental learning of trajectory prototypes.

3. Algorithm Ingredients

- choose a suitable data representation of motion patterns, → *trajectory prototypes*
- define a distance or similarity measure between trajectories or between trajectories and motion patterns, → *LCSS*
- define a method to update the motion patterns. → *keep longer trajectories*

3. Longest Common Subsequence Similarity

Let $Head(T_i)$ be the sequence $\{t_{i,1}, ...t_{i,n-1}\}$. Given a real number $0 < \epsilon < 1$, the LCSS similarity of two trajectories T_i and T_j of respective lengths m and n is defined as

$$LCSS_\epsilon(T_i, T_j) = \begin{cases} 0 & if\ m = 0 \\ 0 & if\ n = 0 \\ 1 + LCSS_\epsilon(Head(T_i), Head(T_j)) & if \text{ the points match} \\ max(LCSS_\epsilon(Head(T_i), T_j), LCSS_\epsilon(T_i, Head(T_j))) & otherwise \end{cases}$$

Two points t_{i,k_1} and t_{j,k_2} match if $|x_{i,k_1} - x_{j,k_2}| < \epsilon$ and $|y_{i,k_1} - y_{j,k_2}| < \epsilon$.

- Distance DLCSS = 1 - LCSS/min(n,m)
- The LCSS can be computed by a dynamic programming algorithm in O(nm).
- This is costly but flexible.

3. Learning Algorithm

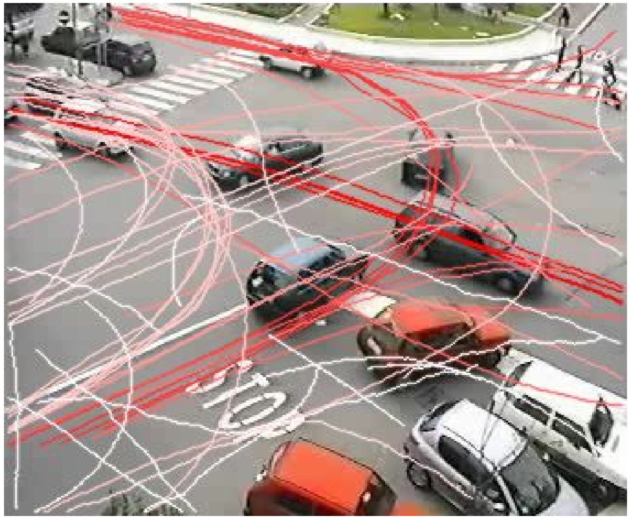
Input: A set of trajectories $Q = \{Q_i\}$, the allowed matching distance ϵ in the LCSS similarity definition, and the maximum LCSS distance δ for two trajectories to match ($0 \leq \delta \leq 1$).
Output: A set of prototype trajectories $P = \{P_j\}$.
for all Trajectory Q_i **do**
  for all Prototype P_j in P **do**
    Compute $DLCSS_\epsilon(Q_i, P_j)$.
    if $DLCSS_\epsilon(Q_i, P_j) < \delta$ AND P_j is shorter than Q_i **then**
      P_j is removed from P.
    if Q_i didn't match any prototype OR Q_i matched at least one shorter prototype **then**
      Q_i is added to P.

- No need to set the number of patterns.

3. Use of Prototype Trajectories

- The number of matched trajectories are counted to provide probabilities of movements.
- The input to the algorithm are feature trajectories (available in abundance), instead of noisy reconstituted object trajectories.
- At prediction time, the feature trajectories are matched against all prototype trajectories.

4. Motion Patterns

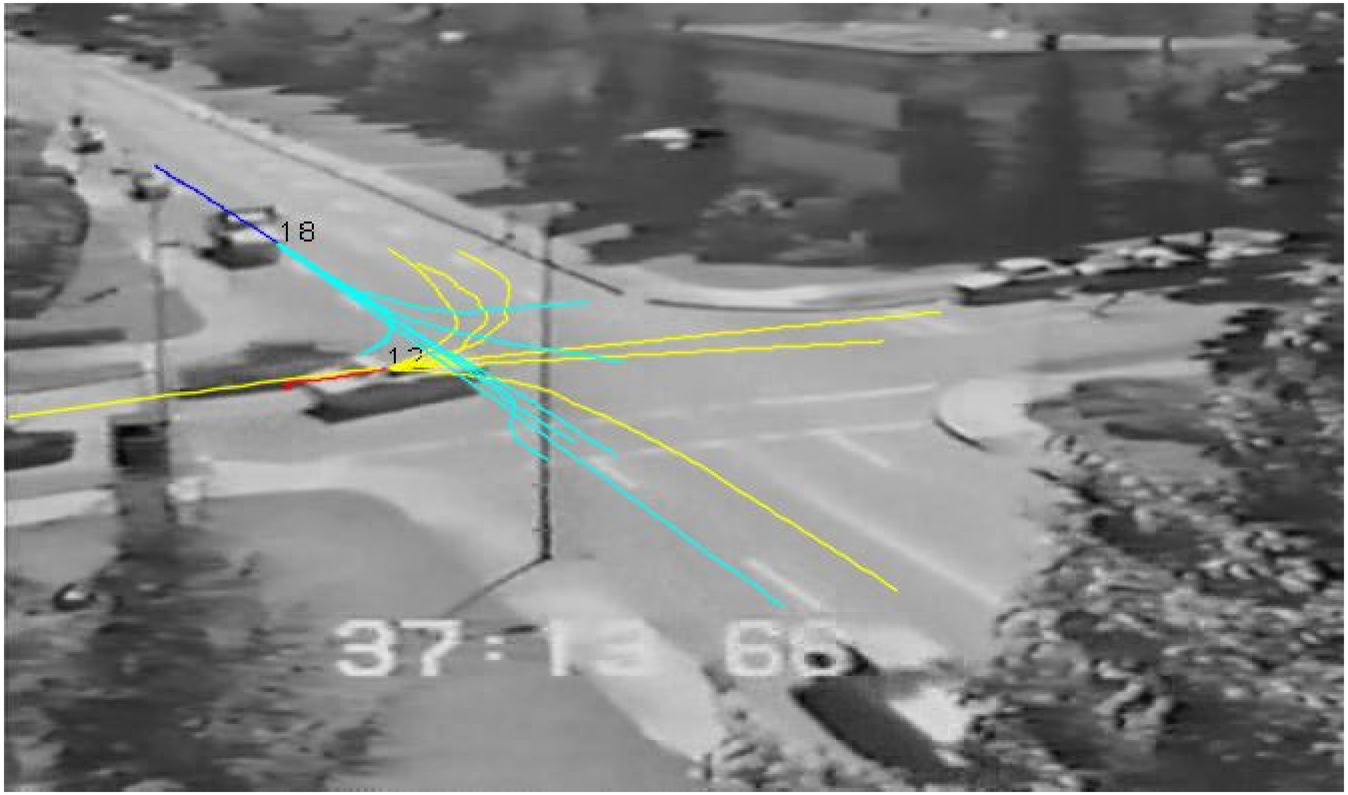


58 prototype trajectories
(138009 trajectories)

128 prototype trajectories
(88255 trajectories)

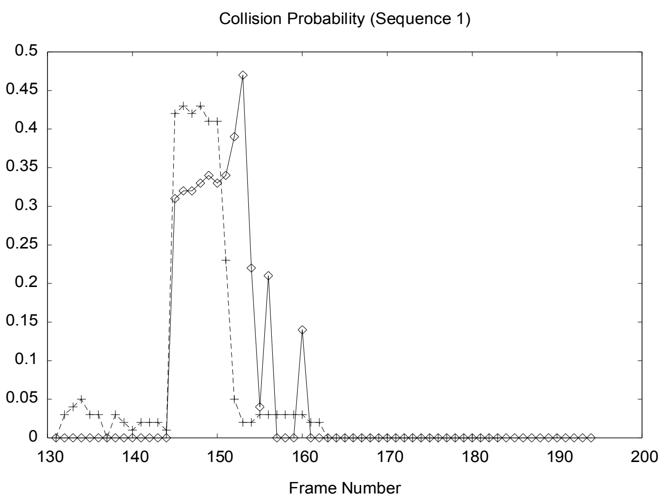
58 prototype trajectories
(2941 trajectories)

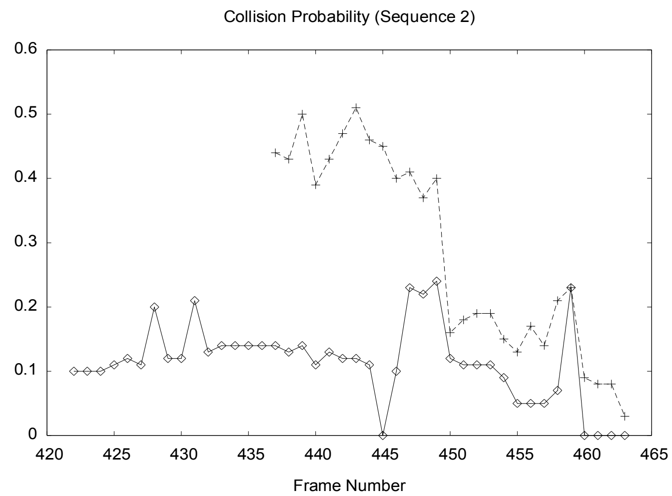
4. Traffic Conflicts

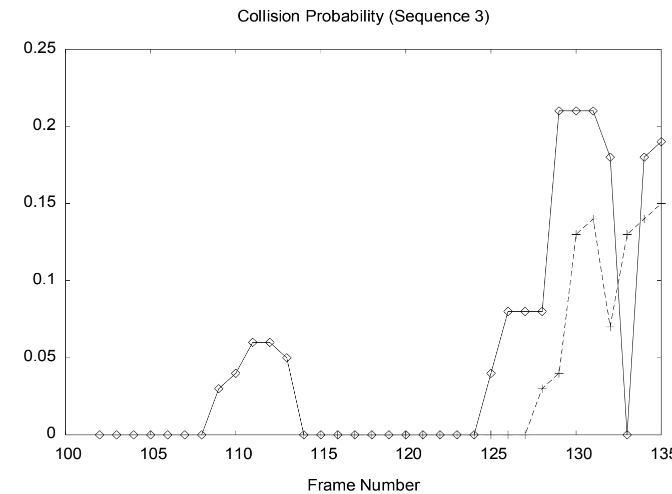


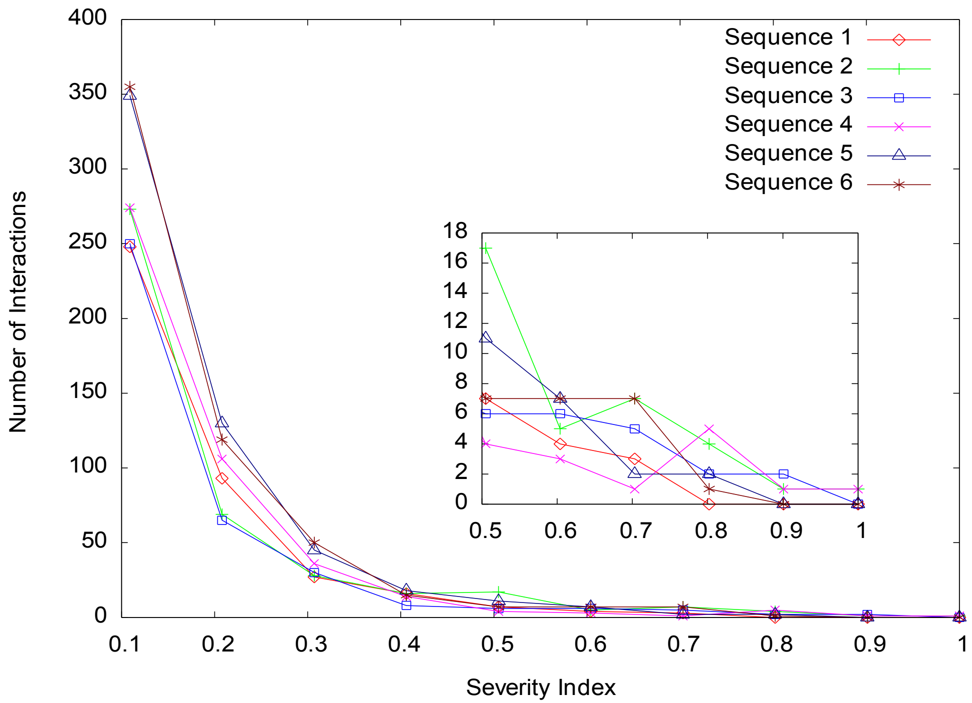
Demo

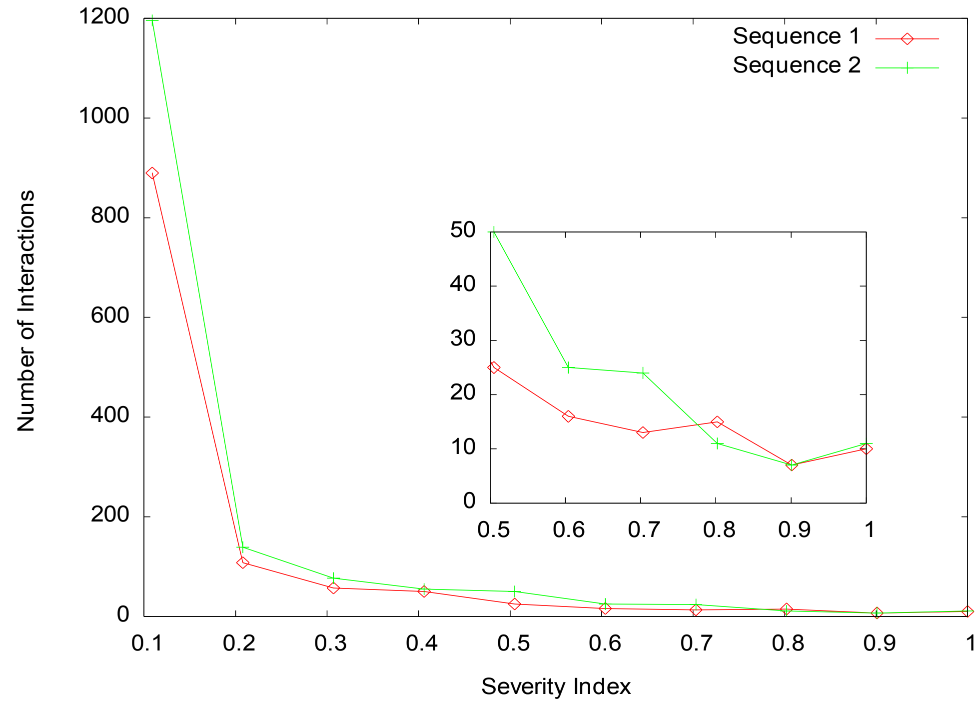
4. Collision Probability











Conclusion

- Probabilistic framework for automated road safety analysis.
- Complete system for automated traffic data collection (traffic intelligence).
- Robustness and versatility of feature tracking.

Future Work

- Improve vehicle detection and tracking: detect shadows, estimate vehicle size.
- Extensions:
  - Road user identification: trucks, buses, vehicles, two-wheels and pedestrians.
  - Pedestrian tracking. Demo

THANK YOU !