**Tracking**

**source**

video-dev/utilities/util-track.cpp

**description**

* traffic640x480.avi
* extract 200 x 200 region of interest
* post-process foreground blobs (see post-processing)
* function to display tracking results:   
  void printTrackUpdate(cv::Mat trackImage, std::list<Track>\* pDebugTracks)
* write frame image files to home/counter/segment-motion:  
  post processing result: debug\_<frame#>.png  
  tracking result: track\_<frame#>.png

|  |  |
| --- | --- |
| ***Error #1 (solved)* splitted blobs must be assigned to same track**  [../../../../Users/Holger/counter/360%20-%20392%20with%20trailer/track%20debug](file:///D:\Users\Holger\counter\360%20-%20392%20with%20trailer\track%20debug) | |
| debug\_372.png | track\_372.png |
| debug\_381.png | track\_381.png |
| **possible Solution (372, 381):**   * for each new rectangle (rc\_new) * calc area of rectangle intersection (rc\_new && rc\_old) * if area\_intersect >= 75% \*area\_new | |

|  |  |
| --- | --- |
| debug\_161.png | track\_161.png |
| debug\_162.png | track\_162.png |
| debug\_163.png | track\_163.png |
| **possible solution (161, 163):**   * reduce confidence to max 2 🡪 old tracks will be deleted faster * combine existing tracks, if they move into same direction and their areas intersect (> 20%) * assign track with shorter history to longer track * keep track with longest length | |

|  |  |
| --- | --- |
| ***Error #2 (solved)* two cars moving in opposite direction must be better separated**  [..\..\..\..\Users\Holger\counter\054 - 067 two cars opposite\track debug](../../../../Users/Holger/counter/054%20-%20067%20two%20cars%20opposite/track%20debug) | |
| debug\_61.png | track\_61.png |
| debug\_62.png | track\_62.png |
|  | track\_63.png |
|  | track\_64.png |
|  | track\_65.png |

|  |  |
| --- | --- |
| debug\_384.png | track\_384.png |
| debug\_385.png | track\_385.png |

**possible solution:**

* maintain size and speed of rc1
* detect new vehicle rc2 (opposite) before separation

|  |  |
| --- | --- |
| ***Error #3 (solved)***  **new blobs must be assigned to spatially close track (error in 392)**  [..\..\..\..\Users\Holger\counter\360 - 392 with trailer](../../../../Users/Holger/counter/360%20-%20392%20with%20trailer) | |
| debug\_391.png | track\_391.png |
| debug\_392.png | track\_392.png |

***Requirement #4 (solved)***

**update only with newBlob, not with intersection of newBlob and lastTrackEntry**

|  |  |  |
| --- | --- | --- |
|  | **wrong**  \improve\_intersect | **right**  \wrong\_separation |
| debug\_61.png | track\_61.png | track\_61.png |
| debug\_62.png | track\_62.png | track\_62.png |

***Requirement #5 (solved)***

if **multiple newBlobs fit into lastTrackEntry** (see example below), then assign

* rightmost blob, if track moves to right
* leftmost blob, if track moves to left

Parameter: intersection area: 0.5 (does not work well with 0.75)

|  |  |  |
| --- | --- | --- |
|  | **wrong**  \ 2018-12-8\_16h\_1m\_18s | **right**  \2018-12-8\_16h-9m-27s |
| debug\_385.png | track\_385.png | track\_385.png |
|  |  |  |

***Requirement #6 (solved)***

if **track has left the scene** partially, then assign

* rightmost blob, if scene has been left on the right
* leftmost blob, if scene has been left on the left

|  |  |  |
| --- | --- | --- |
|  | **wrong**  \2018-12-8\_19h-50m-57s | **right**  \ 2018-12-9\_15h\_55m\_45s |
| debug\_62.png | track\_62.png | track\_62.png |
|  |  | **right**  \ 2018-12-9\_16h\_3m\_52s |
| debug\_385.png |  | track\_385.png |

***Implementation #6***

**Track::checkLeavingRoi**

* private enum: leavingRoiTo {none, left, right}
* trackHistory >= 4
* moving direction: right -> right edge of lastTrackEntry >= 95% \* roiWidth
* moving direction: left -> left edge of lastTrackEntry <= 5% \* roiWidth

***Requirement #7 (closed)***

track leaving roi AND inversing direction must be assigned as new track

intersection of two cars at the edge of roi can only be determined by velocity inversion

|  |  |  |
| --- | --- | --- |
|  | **wrong**  \2018-12-9\_17h\_1m\_36s | **right**  \ 2018-12-15\_16h\_21m\_34s |
| debug\_542.png | track\_542.png | track\_542.png |
| debug\_545.png | track\_545.png | track\_545.png |
| debug\_546.png | track\_546.png | track\_546.png |

***Implementation #7***

**Track::isReversingX**

* true if signBit(avgVelocity) != signBit(prevAvgVelocity)

**SceneTracker::deleteReversingTracks**

* check tracks that are leavingRoiTo left or right for reversing direction
* delete those tracks --> blobs will be assigned to new track in next update step

***Requirement #8 (closed)***

* **reversing tracks must be assigned to new track**
* track leaving roi AND inversing direction must be assigned as new track
* intersection of two cars at the edge of roi can only be determined by velocity reversion
* at low frame rate (10fps) tracks could jump in other direction, inversing direction cannot be determined by reversed velocity (as moving average is calculated over 5 steps)

|  |  |  |
| --- | --- | --- |
|  | **wrong**  /2018-12-19\_20h\_56m\_53s | **right**  /2018-12-23\_13h\_23m\_17s |
| debug\_10.png | track\_10.png | track\_10.png |
| debug\_11.png | track\_11.png | track\_11.png |
| debug\_12.png | track\_12.png | track\_12.png |
| debug\_13.png | track\_13.png | track\_13.png |

***Implementation #8***

**Track::updateAverageVelocity**

* reduce window 5 🡪 3

**Track::isReversingX**

* true, if track reverses in x-direction

**SceneTracker::deleteReversingTracks**

* delete reversing tracks (even if they don't leave roi)
* TODO create new track for deleted one immediately

***Requirement #9 (closed)***

* if track covers entire roi (e.g. occurs for busses, trucks): keep velocity before covering entire roi
* otherwise velocity drops below zero and second track might be introduced

|  |  |  |
| --- | --- | --- |
|  | **wrong**  /2018-12-23\_13h\_54m\_54s | **right**  /2018-12-23\_15h\_55m\_5s |
| debug\_663.png | track\_663.png | track\_663.png |
| debug\_666.png | track\_666.png | track\_666.png |
| debug\_667.png | track\_667.png | track\_667.png |
| debug\_668.png | track\_668.png | track\_668.png |

***Implementation #9***

**Track::updateAverageVelocity**

* if trackEntry.leftEdge < 5% \* roi.width  
  or trackEntry.rightEdge > 95% \* roi.width  
  🡪 avgVelocity = avgVelocity

***Requirement #10 (closed)***

* **reversing tracks (outside backlash) must be assigned to new track**
* velocity difference must be significant in order to avoid re-assigning tracks of stand still motion, e.g. waving leaves

|  |  |  |
| --- | --- | --- |
| 2017-09-18/opposite/2 - 40 cars multi 7m52s | **wrong**  / 2018-12-26\_14h\_50m\_22s | **right**  / 2018-12-26\_15h\_50m\_54s |
| debug\_5.png | track\_5.png | track\_5.png |
| debug\_6.png | track\_6.png | track\_6.png |
| debug\_7.png | track\_7.png | track\_7.png |
| debug\_10.png | track\_10.png | track\_10.png |

***Implementation #10***

**Track::isReversingX**

* true, if track reverses in x-direction AND at least one abs velocity outside backlash

***Requirement #11 (open)***

* **assign matching blobs to track with highest confidence**
* velocity difference must be significant in order to avoid re-assigning tracks of stand still motion, e.g. waving leaves

|  |  |  |
| --- | --- | --- |
| 2017-09-18/opposite/2 - 40 845 - 890 van 3m17s | **wrong**  /2018-12-26\_16h\_14m\_24s | **right**  / |
| debug\_855.png | track\_855.png | track\_5.png |
| debug\_856.png | track\_856.png | track\_6.png |
| debug\_857.png | track\_857.png | track\_7.png |
| debug\_858.png | track\_858.png | track\_10.png |
| debug\_859.png | track\_858.png |  |
| debug\_860.png | track\_860.png |  |

***Implementation #11***

**Track::isReversingX**

* true, if track reverses in x-direction AND at least one abs velocity outside backlash

***Requirement #12 (open)***

* **track occluded objects**

|  |
| --- |
| 2017-09-18/opposite/2 - 40 845 - 890 van 3m17s |
|  |

***Implementation #12***

**SceneTracker::checkOcclusion**

* set isOccluded flag for each occluded track
* tracks moving in opposite direction?
* calc dist
* isNextUpdateOccluded
* calc remainingUpdatesInOcclusion
* calc occlusionRect (SceneTracker property)

**SceneTracker::updateTracksIntersect**

* if isOccluded
  + addSubstitute for occluded tracks, keep confidence
  + discard blobs within occlusionRect (80% or more intersection)
  + decrement remainingUpdatesInOcclusion
  + if remainingUpdates <= 0 -> reset isOccluded flag
  + update all remaining tracks (non-occluded)
* else
  + update all tracks