**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 (fixed)* 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 (fixed)* 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 (fixed)***  **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 (implemented)***

**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 (fixed)***

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 (implemented)***

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 (implemented)***

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 (implemented)***

* **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 (implemented)***

* 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 (implemented)***

* **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 (implemented)***

* **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

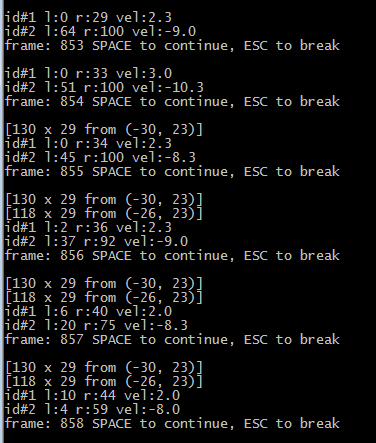
***Refactor velocity calculation***

1. calc velocity for each track update (privat variable in TrackEntry)
2. Track constructor: use ID as parameter only
3. separate addingTrackEntry
4. addTrackEntry
   1. check, if object touches left or right border, calculate velocity based on
      1. left or right edge of object
      2. centroid otherwise
5. UpdateAverageVelocity: use history elements only

***Error#13 (fixed)***

1. **suppress second occlusion introduced in frame 856 (done 13.1)**

* probable cause: if tracks are occluded, nextUpdateOccluded does not have to be checked anymore



**Implementation #13**

1. ***Redesign OcclusionIdList::assignBlobs()***

***Error#14 (fixed)***

1. **occlusion area has been left too early (done 14.1)**

* fix: in occlusion rect:
  + if there is one blob: match right (or left) edge with edge of substitute
  + if two blobs: assign to track, even if update steps have not been reached

**Implementation #14**

1. ***Occlusion::assignBlobs()***
   * 3 cases considered: 0, 1, 2 or more blobs
   * zero blob: calc substitute (using average velocity) and decrease confidence
   * one blob: calc substitute (using average velocity) and decrease keep confidence
   * edges are not adjusted in current version
   * two or more blobs: normal assignment process

***Error#15 (fixed)***

1. **combination of tracks and deletion of reversing tracks only for tracks that are not occluded (done 15.1)**

* see below track\_857.png: velocity will drop below zero in next update step, as velocity of substitute in occlusion is not calculated correctly at ROI border
* track\_858.png: both tracks have been combined into one left moving track (keep#1, discard#2)
* deleteReversingTracks erases #1 and creates new track #2

|  |  |  |
| --- | --- | --- |
| 2017-09-18/opposite/ 845 - 890 | **wrong**  /2019-1-20\_12h\_5m\_57s\_#15 | **right**  /2019-1-20\_19h\_51m\_15s\_#15\_fixed |
| debug\_857.png | track\_857.png | track\_857.png |
| debug\_858.png | track\_858.png | track\_857.png |
| debug\_864.png | track\_864.png | track\_864.png |
| debug\_865.png | track\_865.png | track\_865.png |
| debug\_866.png | track\_866.png | track\_866.png |

**Implementation #15**

1. ***SceneTracker::combineTracks() and SceneTracker::deleteReversingTracks()***
   * combineTracks and deleteReversingTracks for un-occluded tracks only
   * execute deleteReversingTracks before combineTracks
   * fix calculation of substitute values that touch roi border
2. ***adjustSubstPos() not used in this version, keep it for later use***

***Error#16 (fixed)***

1. update occlusion area as blobs move -> faster detection of new blobs moving in the opposite direction (done 16.1)
2. delete occlusion, if one of the tracks has been deleted (done 16.2)
3. delete occlusion, if tracks have passed occluded area (done 16.3)

|  |  |  |
| --- | --- | --- |
| 2017-09-18/opposite/ 2 - 40 | **wrong**  /2019-1-20\_20h\_4m\_56s\_#16 | **right**  / |
| debug\_17.png | track\_17.png | track\_17.png |
| debug\_18.png | track\_18.png |  |
| debug\_19.png | track\_19.png |  |
|  |  |  |
| debug\_33.png | track\_33.png |  |
| debug\_34.png | track\_34.png |  |

***Implementation #16***

1. ***Occlusion::updateRect()***
   * updates occlusion rect based on associated tracks (movingLeft, movingRight)
2. ***SceneTracker::deleteOcclusionsWithMarkedTracks()***
   * erase occlusion from list, if one of the associated tracks (movingLeft, movingRight) has been marked for deletion
3. ***Occlusion::assignBlobs()***
   * set hasPassed bit, if tracks are outside occlusion area
   * occlusion will be deleted in next update step of OcclusionIdList.assignBlobs()

***Error#17 (fixed)***

1. **track entering from roi border and running into occlusion shortly after is not able to extend to full object size, as blobs are cut at roi border**

* in occlusion area substitute track entries are used to represent occluded tracks
* row 2: green track entering scene does not represent full object size -> yellow track is introduced as left blob is not assigned to green track (intersection between track and blob not large enough)

|  |
| --- |
| before blob assignment after blob assignment after deletion of tracks |
|  |
|  |
|  |

**Implementation #17**

1. ***Occlusion::updateRect()***
   * enlarge occlusion rect to roi border from which blob is entering

|  |
| --- |
|  |
|  |
|  |

***Error#18 (fixed)***

1. **between blob assignment and finishing Scene::updateTracks() track IDs are scrambled**

* two red tracks with same ID

|  |
| --- |
|  |

**Implementation #18**

1. ***SceneTracker::setOcclusion()***
   * change data type for movesRight, movesLefty: Track& 🡪 IterTrackconst in order to avoid mis-assignment of tracks

|  |
| --- |
|  |

***Error#19 (fixed)***

1. **after blobs (substitute blobs) went through occlusion, new track is introduced large blob**

* happens for tracks entering scene, when width of substitute track entry cannot be calculated correctly
* single blob (white) should be assigned to right moving track (green) instead of introducing new track (yellow)

|  |
| --- |
| rightSize(60,20); leftSize(30,20); collisionX(20); gapStartX(20) |
|  |
|  |

**Implementation #19**

1. ***Occlusion::assignBlobs()***

* new boolean status var: isMarkedForDeletion
* case one blob: if hasPassed 🡪 set occlusion isMarkedForDeletion
  + if occlusion started at left border 🡪 assign blob to rightMovingTrack
  + if occlusion started at right border 🡪 assign blob to leftMovingTrack
* all cases: set occlusion isMarkedForDeletion

1. ***OcclusionIdList::assignBlobs()***
   * for\_each occlusion: if isMarkedForDeletion 🡪 remove occlusion

|  |
| --- |
|  |
|  |
|  |

***Error#20 (fixed)***

1. **avoid velocity reversion after blob assignment ->>**

* related to Track::updateTrackIntersect(): track moves to right && leaves scene to right 🡪 rightmost track is assigned (usually OK), but velocity jump happens
* see idx#9: green track velocity jumps from 5 to -9 🡪 track reverses and is deleted

|  |
| --- |
| rightSize(30,20); leftSize(60,20); collisionX(80); gapStartX(20) |
|  |
|  |
|  |

**Implementation #20**

1. ***Track::updateTrackIntersect() 🡪 collides with regular assignment(see Requirement#7)***

* ~~no fix: check if blob assignment would reverse track - function isBlobNotReversingTrack()~~
* implementation #21 (extend track entering scene) fixes it

1. **assignment does not work properly**

* idx#8: blob is not assigned to red track (works in 1)
* check last assignment step in occlusion

|  |
| --- |
| rightSize(30,20); leftSize(60,20); collisionX(80); gapStartX(20) |
|  |
|  |
|  |

**Implementation #20**

1. ***Occlusion::assignBlobs()***

* case one blob: use new function Track::updateTrackPassedOcclusion() to assign blob to track in last occlusion update step (occlusion has been passed)

***Error#21 (fixed)***

1. **long blob entering from roi border causes creation of new track, as long blob does not intersect enough with occlusion**

* track must be extended to roi border, if it enters scene

|  |
| --- |
| rightSize(30,20); leftSize(60,20); collisionX(70); gapStartX(30) |
|  |
|  |

**Implementation #21**

1. ***Occlusion::assignBlobs()***

* case one blob:
  + check: isBlobAtRightRoiBorder() or isBlobAtRightRoiBorder()
  + check: hasOppositeTrackEdgePassed
  + if so: extendRectToRightBorder() or extendRectToLeftBorder()

|  |
| --- |
| rightSize(30,20); leftSize(60,20); collisionX(60); gapStartX(40) |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| rightSize(30,20); leftSize(60,20); collisionX(80); gapStartX(20)  check opposite track edge passing roi border:  right moving track (green) reaches roi border at idx#6 🡪 don't extend left moving track at idx#7 and later anymore |
|  |
|  |