Independent Study (W2016)

Yu Yao

April 2016

1 Ground and obstacles classification

Algorithm 1 Ground and obstacles classification

procedure POINTS CLASSIFICATION
    $P \leftarrow$ point cloud
    threshDist $\leftarrow$ *0.3* m
    $G \leftarrow$ empty 3D cubic voxels for groud points
    $U \leftarrow$ empty 3D cubic voxels for above groud points
    for m different hypothesis planes **do**
        $[\textit{inliers, outliers}] = RANSAC(P,\ \textit{threshDist})$
        add *inliers* into G
        add *outliers* into U
procedure OBSTACLES IDENTIFICATION
    $C \leftarrow$ an empty list of clusters
    $Q \leftarrow$ an queue of the points that need to be checked
    Adjacency threshold $d \leftarrow 2$
    for each $p_{i,j,k} \in P$ **do**
        add $p_{i,j,k}$ into Q
        for each $p_{i,j,k} \in Q$ **do**
            for each $p_{i',j',k'} \in P$ **do**
                if $p_{i',j',k'}$ hasn't been processed and $(|i-i'| < d$ or $|j-j'| < d$ or $|k-k'| < d)$ **then**
                    add $p_{i',j',k'}$ into Q
                else continue
        if all points in Q has been processed **then**
            add Q to C and reset Q
        if $\forall p_{i,j,k} \in P$have been processed and $p_{i,j,k} \in C$ **then return**

2 Image parsing module

Algorithm 2 Image parsing

procedure Bottom-up classification
  $S \leftarrow$ segments of one image [1]
  $C \leftarrow$ an empty list of categories of all patches
  for each $s_i \subset S, c_i \in C$ **do**
    $F = \mathit{featureExtraction}(s_i)$ [2]
    add $c_i \leftarrow MLP(s_i, F)$ into C
procedure Top-down correction
  $O \leftarrow$ an empty list of components
  for each category **do**
    find the set of s_i with the same c_i
    add all s_i into O_i
  for each $O_i \subset O$ **do**
    $N_i \leftarrow$ find the neighbor categories of O_i
    $flag_i \leftarrow$ Check if O_i is adjacent to boundaries
    $P_i \leftarrow BayesianNetwork(O, N, flag)$

3 Fuzzy logic based sensor fusion

Algorithm 3 Fuzzification

procedure Fuzzification
  $O \leftarrow$ a list of obstacles
  $Size \leftarrow$ size of candidate obstacles, Lidar
  $Class \leftarrow$ image classification results, camera
  $H \leftarrow$ absolute heights
  $SC \leftarrow$ empty list of spatial context
  $TC \leftarrow$ empty list temporal context
  $k \leftarrow$ the number of current frame
  for $eachO_i \in O$ **do**
    if the number of neighboring ground pixels > 4 **then**
      $SC_i \leftarrow' OBS'$
    else
      $SC_i \leftarrow' NOBS'$
    if $Size_i^k = Size_i^{k-1}$ *and* $Class_i^k = Class_i^{k-1}$ **then**
      $TC_i \leftarrow' HIG'$
    else
      $TC_i \leftarrow' LOW'$
    $Rc_i \leftarrow KnowledgeRules(Size_i, Class_i)$

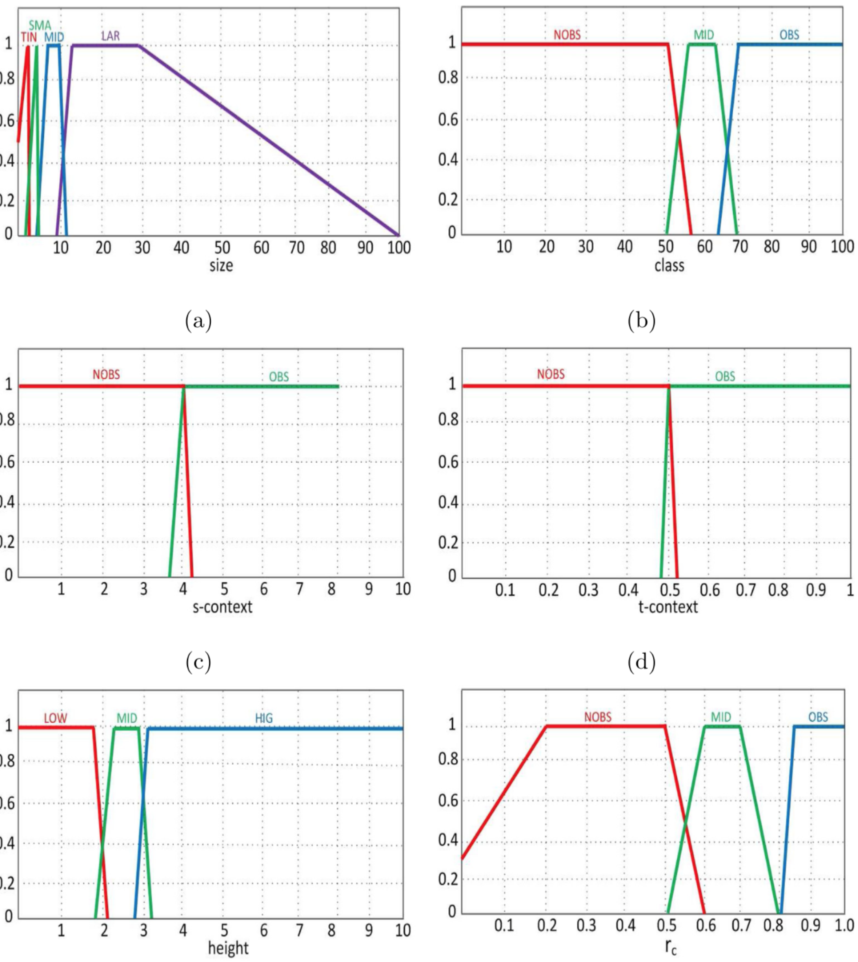

Figure 1: Caption

The membership functions of the labels are presented in Fig.1. For example, the membership function presented in Fig.1(a) when the size of obstacle is identified as 'LAR' can be notated as $U_{LAR}(Size)$. Similarly there are $U_{LAR}(Size)$, $U_{MID}(Size)$, $U_{SMA}(Size)$, $U_{TIN}(Size)$, $U_{OBS}(Class)$...

Algorithm 4 Knowledge rules of scene classification, Lidar

procedure 20RULES
    R1: if $Size$ is 'LAR' and $Class$ is 'OBS' then Rc is 'OBS';
    R2: if $Size$ is 'LAR' and $Class$ is 'MID' then Rc is 'MID';
    R3: if $Size$ is 'LAR' and $Class$ is '$NOBS$' then Rc is '$NOBS$';
    R4: if $Size$ is 'LAR' and $Class$ is '$NOBS$' and SC is $NOBS$ then Rc is '$NOBS$';
    R5: if $Size$ is 'LAR' and $Class$ is '$NOBS$' and TC is $NOBS$ then Rc is '$NOBS$';
    R6: if $Size$ is 'LAR' and $Class$ is '$NOBS$' and SC is OBS then Rc is 'OBS';
    R7: if $Size$ is 'MID' and $Class$ is 'OBS' then Rc is 'OBS';
    R8: if $Size$ is 'MID' and $Class$ is 'MID' then Rc is 'MID';
    R9: if $Size$ is 'MID' and $Class$ is '$NOBS$' then Rc is 'MID';
    R10: if $Size$ is 'MID' and $Class$ is 'MID' and SC is $NOBS$ then Rc is '$NOBS$';
    R11: if $Size$ is 'MID' and $Class$ is '$NOBS$' and SC is $NOBS$ then Rc is '$NOBS$';
    R12: if $Size$ is 'MID' and $Class$ is '$NOBS$' and TC is $NOBS$ then Rc is '$NOBS$';
    R13: if $Size$ is 'MID' and $Class$ is '$NOBS$' and H is MID then Rc is '$NOBS$';
    R14: if $Size$ is 'SMA' and $Class$ is 'OBS' then Rc is 'MID';
    R15: if $Size$ is 'SMA' and $Class$ is 'OBS' and SC is OBS then Rc is 'OBS';
    R16: if $Size$ is 'SMA' and $Class$ is 'OBS' and SC is $NOBS$ then Rc is '$NOBS$';
    R17: if $Size$ is 'TIN' then Rc is 'MID';
    R18: if $Class$ is '$NOBS$' and H is MID then Rc is '$NOBS$';
    R19: if $Class$ is 'MID' and H is MID then Rc is '$NOBS$';
    R20: if H is HIG then Rc is '$NOBS$';

Algorithm 5 Fuzzy reasoning

procedure FUZZY REASONING
    $L \leftarrow$ empty list of labels candidate obstacles
    for $eachL_i \in L$ **do**
        for $eachR_i \in 20RULES$ **do**
            $U^i_{label}(Rc^*) = min(U^i_{label}(Size^*), U^i_{label}(Class^*), U^i_{label}(H^*), U^i_{label}(SC^*), U^i_{label}(TC^*))$
        $U_{OBS}(Rc^*) = max(U^1_{OBS}(Rc^*), ..., U^2_{OBS}0(Rc^*))$
        $U_{MID}(Rc^*) = max(U^1_{MID}(Rc^*), ..., U^2_{MID}0(Rc^*))$
        $U_{NOBS}(Rc^*) = max(U^1_{NOBS}(Rc^*), ..., U^2_{NOBS}0(Rc^*))$
        $U_O(Rc) = max(min(U_{OBS}(Rc^*), U_{OBS}(Rc)), min(U_{MID}(Rc^*), U_{MID}(Rc)), min(U_{NOBS}(Rc^*), U_{NOBS}(Rc)))$
        $Rc^* = \frac{\int U_O(Rc)RcdRc}{\int U_O(Rc)dRc}$
        if $Rc^* > 0.65$ **then**
            $L_i \leftarrow' OBS'$
        else $L_i \leftarrow$ labels of other categories

References

[1] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," in *International Journal of Computer Vision*, vol. 59, no. 2, 2004, pp. 167–181.

[2] J. Zhang, M. Marszalek, S. LAZEBNIK, and C. SCHMID, "Local features and kernels for classification of texture and object categories: A comprehensive study," in *International Journal of Computer Vision*, vol. 72, no. 2, 2007, pp. 213–238.

Algorithm 6 Temporal fusion of consecutive frames

procedure CONSECUTIVE FRAMES FUSION
  $f_k \leftarrow$ Markov random field model
  $corr \leftarrow DenseOpticalFlowMethod(f_k, f_{k-1})$
  $BeliefPropagation(f_k, f_{k-1}, corr)$