Multiple Obstacle Detection and Tracking using Stereo Vision: Application and Analysis

Bihao Wang, Sergio Alberto Rodriguez Florez, Vincent Frémont

Multiple Obstacle Detection and Tracking using Stereo Vision: Application and Analysis

Bihao Wang
CNRS Heudiasyc UMR 7253
Université de Technologie de Compiègne
Compiègne, France
Email: bihao.wang@hds.utc.fr

Sergio Alberto Rodríguez Florez
CNRS IEF UMR 8622
Université Paris-Sud
Paris, France
Email: sergio.rodriguez@u-psud.fr

Vincent Frémont
CNRS Heudiasyc UMR 7253
Université de Technologie de Compiègne
Compiègne, France
Email: vincent.fremont@hds.utc.fr

***Abstract*—Vision systems provide a large functional spectrum for perception applications and, in recent years, they have demonstrated to be essential in the development of Advanced Driver Assistance Systems (ADAS) and Autonomous Vehicles. In this context, this paper presents an on-road objects detection approach improved by our previous work in defining the traffic area and new strategy in obstacle extraction from U-disparity.Then, a modified particle filtering is proposed for multiple object tracking . The perception strategy of the proposed vision-only detection system is structured as follows : First, a method based on illuminant invariant image is employed at an early stage for free road space detection. A convex hull is then constructed to generate a region of interest (ROI) which includes the main traffic road area. Based on this ROI, an U-disparity map is built to characterize on-road obstacles. In this approach, connected regions extraction is applied for obstacles detection instead of standard Hough Transform. Finally, a modified particle filter framework is employed for multiple targets tracking based on the former detection results. Besides, multiple cues, such as obstacle's size verification and combination of redundant detections, are embedded in the system to improve its accuracy. Our experimental findings demonstrates that the system is effective and reliable when applied on different traffic video sequences from a public database.**

***Index Terms*—Stereo Vision, On-road Obstacles Detection, Particle Filter, Visual Tracking.**

I. Introduction

Obstacle detection and tracking are key issues of the Advanced driver assistance systems (ADAS). In the context of driver assistance, the purpose of obstacle detection and tracking system is to detect and monitor the dynamic behavior of one or more obstacles in the vicinity of the host vehicle. Hence, the ADAS can help to avoid potential collisions and to provide essential information for decision making. A reliable detection and tracking approach in real environments has been a challenge in the last two decades, especially considering various type of objects, their time varying number, and their states estimation from noisy observations at discrete intervals of time. Sensors, such as radar and LIDAR, have been used for this purpose using sensor data fusion approaches [1], [2]. In recent years, many vision-only based approaches have been developed [3], [4] for the versatility of information they can provide and their low cost. In these approaches, learning-based methods focus on the detection and tracking of specific obstacles: like pedestrians and vehicles [5], [6]; while motion-based methods can extract the moving objects [4], [7]. In this paper, we present an on-road object detection approach from our previous work [8], [9] which can effectively detect the road area in traffic scene and a multi-object tracking strategy based on particle filter. Since U-V-disparity map [10] is an effective method to detect objects regardless their appearance and motion model, we propose to use it in the obstacle detection part. In order to predict the obstacle's position and moving direction, tracking is added in the system as a complementary to object detection [5], [11]. Especially, particle filters [11], [12] are widely used to solve multiple time varying obstacles tracking problems. Their strength lies in their ability to represent non-Gaussian distributions which can capture and maintain target properties.

First, stereo-vision-based obstacle detection is applied on a region of interest (ROI), which is composed of the main traffic area. To obtain the ROI, a fast road surface detection method [8] is firstly applied. Then, a convex hull algorithm is introduced to achieve the complete traffic area. Within the ROI, a method of connected region extraction from U-disparity map is developed to locate the obstacles in the image, and furthermore, to refine their position information from the sub-region of disparity map extracted from the primary location of the obstacles. To improve the detection accuracy, multiple cues are integrated in the system, such as an adaptive height gating for obstacle detection in different distances and a combination of close detected area in the U-disparity map.

After obtaining the position, shape and depth information of the on-road obstacles, multiple target tracking hypotheses are managed by the means of a bank of particle filters. Target-to-track association is carried out following a global nearest neighbor (GNN) criterion. The tracking is performed in the image plane of the left camera in stereo vision system. In the 2D image plan, the obstacle's position and size is effected by it's distance to camera. To cope with this factor, the observed dynamics of the tracked obstacles is employed to define an adaptive association gate. Besides, a dynamic noise generation function is implemented in the filter. Considering the number of obstacles is time varying, for each iteration of the filtering, multiple hypotheses are made to create, delete and update the existing tracks. Fig. 1 shows the outline of the proposed perception system.

The approach is structured into two parts: on-road ob-

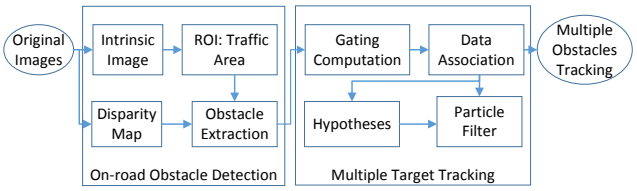


Figure 1: System Outline

stacles detection based on connected region in U-disparity map; and modified particle filter tracking of multiple targets. The strengths of this approach are: (1) It proposes a reliable detection and tracking system that can be directly applied in different driving scenarios. (2) It is capable of detecting all the on-road obstacles with efficiency and accuracy, regardless of their shapes and poses. (3) It presents a modified particle filter for visual tracking, which has a great tolerance for the dynamics of obstacles in image plan caused by depth factor.

The paper is organized as follows: First, a stereo-vison based on-road obstacle detection is introduced in Section II which includes two parts, traffic area extraction (Section II-A) and obstacle detection (Section II-B). Then, a modified particle filter tracking of multiple targets is presented in Section III. Experimental results and analysis on publicly available dataset [13] are shown in Section IV. Finally our paper ends with conclusions and future work.

II. On-road Obstacles Detection

The disparity map, $I\Delta$, can be extracted from stereo images[14]. It refers to the displacement of the relative features or pixels between two views. A bigger disparity value corresponds to a closer distance to the camera. In $I\Delta$, an obstacle is represented as a homogeneous part with the same disparity value. U-V disparity maps are built by accumulating the pixels with same disparity value along the u, v axis of $I\Delta$ separately. The V-disparity map, $I_v\Delta$, is usually employed to estimate the longitudinal profile of the road and to detect the presence of obstacles by the means of a line extraction algorithm. However, in complex scenario, the V-disparity map $I_v\Delta$ is ambiguity prone. When obstacles are closed to each other, their representative lines in $I_v\Delta$ are mixed together. On the contrary, the U-disparity map preserves more information of the scene: the objects width, their relative positions and their depth information are kept. Therefore, in this paper, the U-disparity map leads to an accurate obstacles detection, while the V-disparity only assist in this procedure.

A. Traffic Area Extraction

In order to reduce the computational cost and to improve the detection efficiency on the road area, a suitable road ROI needs to be defined. It removes off-road information which may interfere the precision of on-road obstacle detection, for instance, continuous high walls/buildings along the road. From this consideration, a free road surface detection combined with a convex hull construction is proposed hereafter.

1) Application of Convex hull : According to [8], the free road surface I_R can be computed by the conjunction of intrinsic road surface and the ground plane. Convex hull algorithm here provides the smallest convex area which contains all the free road surface I_R. It fix up the holes and the depressions caused by on-road obstacles. A complete traffic area, i.e. the ROI, is then generated from free road surface. Therefore, obstacle detection can be focused on this approximated road traffic area. Even if the convex hull may not exactly follows the shape of the road, in most of the cases, it is sufficient to provide a satisfying ROI for further detection. For other cases, the tracking process detailed in Section III will efficiently deal with this issue.

B. U-disparity map based obstacles detection

In traffic scenes, there exist pedestrians, vehicles, traffic lights and signs, etc. For that purpose, the U-disparity map can be used to handle all the on-road obstacles, without prior knowledge on their types and motion models.

1) Connected-region extraction: In the U-disparity map $I_u\Delta$, obstacles are usually represented as straight lines. However, when an obstacle is passing near the camera side, both the frontage and side face of the obstacle are observed. The obstacle is then represented by a polyline: an horizontal part for frontage and a connected oblique part for its side face (As shown in Fig. 2 for the second obstacle in the second column). This situation happens frequently in driving scenes. To cope with this problem and to simplify the detection processing, a connected-component extraction algorithm is introduced in this paper to replace the classical Hough line extraction.

After a preprocessing using Eq. (1), high intensity regions are preserved in the U-disparity map $I_u\Delta$ and the other pixels are set to 0 (i.e. background).

$$I_u\Delta = \mathrm{sgn}(I_u\Delta(p) - \varepsilon) \tag{1}$$

The definition of the intensity threshold, ε, is related to the camera calibration parameters and object's depth information. In our experiment, it has been set to 8 to 10 accumulated pixels.

After applying morphological operations (here, erosion and clean), noisy pixels are removed from $I_u\Delta$. Each connected-region L being preserved in $I_u\Delta$, indicates a potential obstacle O_L. Thus, the passing-by obstacle's information can be obtained by connected-component extraction algorithm. These information include: left bound u_l and right bound u_r of O_L on the u-axis of the image; and its disparity value d_O. The complementary information about O_L like the height h_O and the bottom position on the v-axis v_b can be extracted from $I_v\Delta$ and furthermore refined by sub-region of the disparity map which contains the obstacle.

2) Obstacle localization with sub-Disparity map : For the obstacles standing at the same distance to camera, their accurate height information is mixed in $I_v\Delta$. In order to refine the location of each potential obstacle O_L, a sub-region of disparity map $I_O\Delta$ for each obstacle is extracted from the complete disparity map $I\Delta$ according to their primarily

Algorithm 1 On-road Obstacle Detection Algorithm

Input: - Stereo color images $I_l,\ I_r$
Output: Number of detected obstacles N_{obs}, and their location information $O_{1...N_{obs}}$

for k =first frame **do** last frame ▷ Evolution of frames
  ▶ Disparity map $I_\Delta \leftarrow (I_l,\ I_r)$ and free road surface I_R;
  ▶ Convex hull construction: $I_{ROI} \leftarrow I_R$;
  ▶ U-V-disparity map on ROI: $[I_u\Delta, I_v\Delta] \leftarrow (I_{ROI},\ I_\Delta)$;
  ▶ Label the connected-regions $L_{1,...,N}$ in $I_u\Delta$:
  for $i = 1$ **do** N ▷ Location extraction
    ▶ Extract primary position and disparity value $[u_l, u_r, v_b, h_L, d_L] \leftarrow (L_i, I_u\Delta, I_v\Delta)$
    ▶ Generate sub-Disparity map for each object O_i: $I_O\Delta \leftarrow (u_l, u_r, v_b, h_L)$
    ▶ Extract obstacle by (2): $I_O \leftarrow I_O\Delta$
    ▶ Refine obstacle position $[x_O, y_O, w_O, h_O, d_O] \leftarrow I_O$,
    if $h_O \geq \delta(d_O)$ **then** ▷ Eliminate false alarm
      ▶ $N_{obs} \leftarrow N_{obs} + 1$;
      ▶ $O_{N_{obs}} = [x_O, y_O, w_O, h_O, d_O]$
    end if
  end for
end for

estimated location in the image. Pixels in these sub-regions of disparity map are classified into two classes, object or background, according to their disparity value.

$$\begin{cases} I_O = 1 \text{ object}, & \text{if } I_O\Delta(p) \in [d_1, d_2] \\ I_O = 0 \text{ background}, & \text{otherwise} \end{cases} \quad (2)$$

where, I_O is the binary image with labeled obstacles. $d_{1,2} = d_O \pm \sigma$ where, σ is the bias of possible disparity value of the same obstacle. Obstacle's position and size information is then refined in I_O. In this approach, the obstacle's information is represented by its centroid (x_O, y_O), its width w_O, its height h_O and its disparity d_O. Compare to region growing algorithms, the sub-disparity map extraction is much faster and effective. A detection example is showed in Fig. 2.

3) Multiple cues integration: In some cases, the connected component extraction may lead to false alarms or multiple detections on a single obstacle. Thus, multiple cues can be combined to refine the detection result.

- Height limitation of potential obstacles: If the object's height is smaller than a threshold δ, it will not be considered as an obstacle. This threshold is proportional to the disparity value of this potential obstacle $\delta \propto d_O$. The closer O_L stands to the camera, the higher δ will be. Thus, the system can eliminate part of the false alarms. As in Fig. 2, the yellow lines in left middle image are the false alarms detected by $I_u\Delta$.
- Combination of closely stand connected-regions: Since U-disparity image is accumulated on discrete values from the disparity map, there could exist fragments of the same obstacle using the representation of connected-region. This will lead to redundant detections. To handle this problem, a combination operations, like bridge and dilation morphological operations are introduced.

The complete on-road obstacle detection pipeline is summarized in Algorithm 1.

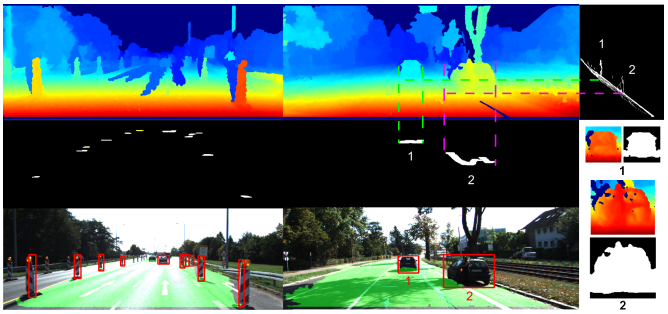


Figure 2: Procedure of on-road obstacle detection. For first two columns, from top to bottom: disparity map; U-disparity map within ROI (convex hull of the green area in detection results); detection result. Top-right is V-disparity map, examples of sub-regions of Disparity map for obstacle extraction are shown under it.

III. Multiple Obstacles Tracking

When multiple obstacles have been detected and they are tracked over frames, several unknowns must be properly handled, such as the number of targets at each time step and the data association between targets and tracks. On-road obstacles are regarded as the targets, see Fig. 1. Hereafter, a modified particle filter based on the condensation algorithm, is proposed to track every target. A dynamic noise update function is introduced and a self-adaptive gating is also designed to provide a reliable data association result.

A. Particle filter model

The condensation algorithm, as a special case of particle filtering[15], provides a well-established methodology for generating samples from the required distribution without requiring assumptions about the state-space model or the state distributions. The samples from the distribution are represented by a set of particles; each particle has a weight representing the probability of that particle being sampled from the probability density function. At each time step, particles' weight and spatial distribution are used for tracker state estimation and re-sampling. In this paper, the tracking of the surrounding obstacles is carried in 2D image plane. In order to define an uniform state space, all the states are described in pixel level. The depth information is then represented by its corresponding disparity value. The filter model is build as follows:

- State vector:

$$S = [x, y, v_x, v_y, w, h, d]^T \quad (3)$$

It is composed by the centroid object position (x, y) on the image; the velocity of the centroid v_x, v_y ; the width w; the height h and the disparity value d respectively.

- Observation:

$$Z = [x_O, y_O, 0, 0, w_O, h_O, d_O]^T \quad (4)$$

The values x_O, y_O, w_O, h_O, d_O represent the information of detection result of Algorithm 1. Since the velocity of the object cannot be measured directly, it has been set to a 0 value.

- Estimation:

$$S(k) = \sum_{i=1}^{N_s} s^i(k) \cdot \pi^i(k) \quad (5)$$

where, S is the condensation state of the tracker, N_s is the number of samples. $s^i(k)$ is current sample states, where, $i = 1, \ldots, N_s$, and k represents the time step. The variable π^i is the normalized weight distributed for each sample.

- Prediction:

with
$$s^i(k+1) = f(s^i(k)) \tag{6}$$

$$s^i(k) = [x^i(k), y^i(k), v_x^i(k), v_y^i(k), w^i(k), h^i(k), d^i(k)] \tag{7}$$

$s^i(k)$ and $s^i(k+1)$ are consecutive particle states of current time and next time respectively. f is the dynamic model for evolution. In this approach, f is a constant velocity model. Thus, (6) can be written as :

$$\begin{cases} x(k+1) = x(k) + T \cdot v_x(k) + W_x(k) \\ y(k+1) = y(k) + T \cdot v_y(k) + W_y(k) \\ v_x(k+1) = v_x(k) + W_{v_x}(k) \\ v_y(k+1) = v_y(k) + W_{v_y}(k) \\ w(k+1) = w(k) + W_w(k) \\ h(k+1) = h(k) + W_h(k) \\ d(k+1) = d(k) + W_d(k) \end{cases} \tag{8}$$

$$W(k) = [W_x(k), W_y(k), W_{v_x}(k), W_{v_y}(k), W_w(k), W_h(k), W_d(k)]^T \tag{9}$$

where, $W(k)$ is the noise vector at time step k that added for filter evolution.

- Update and Re-sampling:

$$\pi^i(k+1) = \frac{P(s^i(k+1 \mid k) \mid Z(k+1))}{\sum_{i=1}^{N} P(s^i(k+1 \mid k) \mid Z(k+1))} \tag{10}$$

For each evolution, particles $s^i(k+1 \mid k)$ are predicted from their previous state $s^i(k)$ by Eq. (7). A new confidence density $P(s^i(k) \mid Z(k))$ is then distributed to $s^i(k+1 \mid k)$ through the comparison between particle states and associated observation $Z(k+1)$. The weights of samples are then updated from the confidence density by Eq. (10). Subsequently, a new set of N_s particles $s^i(k+1)$ are constituted from the current sample set $\{s^i(k+1|k)\}$ with probability proportional to the confidence distribution [15].

B. Dynamic noise update

In the image coordinate frame, the scale and the displacement of an obstacle change rapidly with its distance to the camera. To handle this issue, the noise vector added in Eq. (8) should follow this change as well, i.e. $W(k) \propto d_O(k)$. Therefore, the tracker is able to keep up with observation state in 2D image coordinates. At every sampling time, a dynamic noise update function is build as follows:

$$W(k) = C \cdot d_O(k) \cdot Z(k) \tag{11}$$

where, C is the coefficient vector that needs to be adjusted given an application. For a given equipment, these parameters can be set for once, because the variations lead by depth are related to the camera's essential matrix [16]. The noise vector $W(0)$ is initialized with high values to provide a broader range of sample distribution. During the filtering, the noise vector $W(k)$ is set with lower order of values to provide a convergent range for particle predictions.

C. Data association

Global nearest neighbor (GNN) is the most natural data association process with a low complexity. When obstacle observations are perceived from the camera, Mahalanobis distances between each observation and prediction are calculated. In our approach, the distance between object and track is defined as follows:

$$dist = c_1.\Delta_x^T \Delta_y + c_2.\Delta_w^T \Delta_h + c_3 \mid \Delta_d \mid \tag{12}$$

where, $c_{1,2,3}$ are the coefficients for different measurements which indicate their contributions to the distance calculation. Δ is the difference between estimation and observation measured on the state vector $[x, y, w, h, d]$ separately. Thus, the obstacle's centroid (x, y) in 2D image is not the only criterion that contributes to the distance calculation, but also the width, the height (w, h) and the disparity value d are considered as well.

1) Self-adaptive gate : Target-to-Track association is limited by the use of a gate which is set to a constant value for eliminating unlikely association. In 2D image plane, the closer the obstacle stands to the camera, the greater Mahalanobis distance it might have with respect to the tracks in 2D image coordinates. For the on-road obstacles which are far from camera, they have smaller scales and stand closer to each other; a big gate will lead to mismatching. On the contrary, for nearby obstacles, they may not be able to be associated with the proper tracks because of a small gate. Thus, a constant gate is not sufficient for all the obstacles standing in different distances. In this paper, a self adaptive gate is modified with respect to every observed obstacle according to their scale and depth information:

$$G_O = a \cdot w_O(k)^T h_O(k) + b \mid d_O(k) \mid \tag{13}$$

where, G_O is the gate for each obstacle according to their observation $[x_O, y_O, 0, 0, w_O, h_O, d_O]^T$ at time k. Here, we set $a = 0.5,\ b = 0.2$, which is basically the radius of the circumscribed circle plus a small percentage of the disparity value. Thus, the gate is only related to the current observation's scale and depth information. This design greatly improved the reliability of data association algorithm.

2) Multiple hypotheses: Obstacles and tracks are associated by global minimal distance within the gate. If there is no association established. This leads to two possible situations: non-associated obstacle or non-associated track. In the first case, it is assumed that a new obstacle is just detected, and a new track needs to be created for this obstacle. In the second case, non-associated track will be preserved and updated for a short time period unless the tracking failed up to a threshold. In that case, the track would be pruned.

IV. Experimental Results

The proposed algorithm has been evaluated on different sequences of KITTI dataset [13]. There are different types of road. Tracklet labels of the dataset are used as "ground truth" for a comparison and evaluation of this work.

- Dataset 1: urban road.

Algorithm 2 Modified Particle Filter Algorithm

▶ Initialization: Set $k = 0$, generate a sample set $\{s^i(t,k)\}$ for each detected target/obstacle at current time k, where, $i = 1,\ldots,N_s$, $t = 1,\ldots,N_{obs}(k)$. $N_{obs}(k)$ is the number of detected obstacles at time k. Particle $s^i(t,k)$ is draw from Gaussian distribution around $Z(t,k)$
for $k = 1$ **do** last frame ▷ Evolution of frames
  for $t = 1$ **do** $N_{obs}(k)$ ▷ Tracking of each obstacle
    ▶ Compute adaptive gate $G_O(t,k)$ for data association by (13) for each obstacle
    ▶ Associate tracker with obstacle by GNN algorithm
    ▶ Update the weight of particles $\pi^i(t,k)$ by (10)
    ▶ Re-sampling of particles $s^i(t,k)$ from current sample set according to $\pi^i(t,k)$
    ▶ Estimate the tracker state by (5)
    ▶ Predict the state of particles $s^i(t,k+1)$ by (8)
    if Non-associated Obstacle **then** ▷ New obstacle
      ▶ Generate new tracker sample set $\{s^i(t,k)\}$ for the obstacle, where, $i = 1,\ldots,N_s$
    end if
  end for
  for Non-associated tracker **do** ▷ Obstacle left the scene
    if the tracker has not been associated for a period **then**
      ▶ Prune track hypothesis.
    end if
  end for
end for

- Dataset 2: high way.
- Dataset 3: rural road
- Dataset 4: busy urban road

The algorithm is implemented in a standard PC with Windows 7 Enterprise OS, Intel CPU of 2.66 GHz. The development environment is MATLAB R2013b. Disparity map is obtained from LIBELAS toolkit [17] and particle filter functions from OpenCV [18] are integrated in the code. The run-time is about 3.4s per frame for on-road detection processing and 0.05s per frame for multiple obstacle tracking algorithm. The detection distance in disparity map is limited to 35m in front of the camera.

A. Experiments design

KITTI 3D tracklet labels provide the position and motion history of the obstacles appeared in the scene. After projection onto the image plane, the tracklet 2D position could be seen as ground truth trajectories for object detection and tracking. However, there are some considerations that need to be made for our detection and tracking result to be evaluated based on tracklet labels. First, tracklets only labels vehicles and pedestrians, other type of obstacles are not included, such as traffic cones in Dataset 2 . However, they should be detected as on-road obstacles for ADAS. Second, the tracklet labels also provide off road information of obstacles which are beyond the traffic area considered in our approach. Third, our stereo vision based detection distance is set up to 35m, while the tracklet reaches to 70m. Thus, the evaluation is constrained to the intersection between our detection results and labeled tracklets. To establish a comparable evaluation platform, a sequence of 100 frames is chosen from the four datasets respectively. Then, the GNN association is applied to pair our experimental results with tracklets. Hence, on-road tracklets are picked out by data association. From the tracklets label list, all the obstacle presences within 35m distance to the camera are preserved as ground truth. To ensure the evaluation result integrity, some special cases, like false alarms, missed detection and redundant detections, are also noted manually during the experiment of detection and tracking.

B. On-road obstacle detection

Dataset 1 contains 150 detections on road, 136 of them are associated with trackelet, the rest are false alarms and redundant detections. Dataset 2 contains 363 detections on road, 48 of them are associated with tracklet. In Dataset 2, except for false alarms and redundant detections, 309 non-associated detections are traffic cones standing on the road which are not listed in tracklet. Dataset 3 contains 37 detections on road, 34 of them are associated with tracklet, 4 are false alarms; Dataset 4 contains 257 detections, 215 of them are associated with tracklet; the rest of them are composed of detection of traffic signs, false alarms and redundant detections.

The detection results are evaluated following 5 indicators stated in Table I: false alarm, missed detection, redundant detection and average error of position and size. According to the experiment observation, most of false alarms are caused by the obstacles standing beside the road edge, e.g. trees. In Dataset 2, there is no obstacle beside road edge, so false alarms rarely occur. Most of missed detections appear particularly in the two sides of the image on the bottom when obstacles closely pass by the camera. They are usually induced by the errors of the disparity map. Imprecise disparity map values can also lead to redundant obstacle detections. The average error (AEC) illustrates the average distance between detected obstacle centroid and labeled tracklet centroid (ground truth). In addition, the average error which measures the variation of size scale (AES) is also listed in Table I. The two indicators are measured on pixel-level of the 2D image plan.

Measures	false alarm	missed detection	redundant detection	AEC	AES
Dataset 1	8.6%	4.0%	3.3%	5.3px	8.7px
Dataset 2	1.1%	3.3%	1.3%	6.8px	12.4px
Dateset 3	10.8%	8.1%	0%	9.6px	20.2px
Dataset 4	4.6%	3.1%	4.2%	10.9px	17.1px

Table I: Evaluation of the on-road detection results

C. Multiple obstacle tracking

Multiple target tracking can not only record and predict the motion of the obstacles, but also can deal with the occasionally missed detections and/or occlusions. As show in Fig. 4a, for the obstacle with tracklet id of 1, there's one frame of missed detection, while the track remains complete by filling the blank by prediction. During the tracking, four qualities are evaluated: rate of track fragmentation, rate of overlap, the average precision of tracker's position and size at each time step. As show in Table II, the tracking result

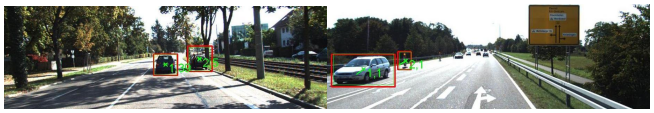
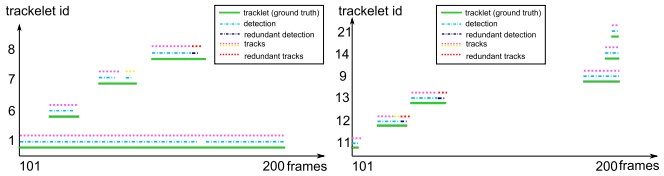


Figure 3: Examples of tracking results



(a) Dataset 1 (b) Dataset 2

Figure 4: Consistence of the tracks related to tracklets

is solid, with most track fragmentation rates under 6.81% during the sequence. The rate of tracks overlap in Dataset 1 is a bit less than in the other datasets, because in Dataset 1, the left side of road is lower than the right side. Thus, the disparity distribution of the road surface fluctuates, which makes the left-side vehicle hardly being detected. During the experiment, track fragmentation happens a lot, when obstacles move closely. They are illustrated in Fig. 4b, in which different colors of track stands for different tracks. Under the high speed circumstances, obstacles that move towards the camera have a high relative speed. As projected in the image, their centroid move faster and their sizes change rapidly over time. When the tracker can not catch up with the target for a certain period, it will be pruned and a new tracker will be created for the obstacle. One should notice here that, in Table II the missed detection and false alarms are effectively reduced by tracking process compare to Table I. Because tracking can fill up the gaps caused by instant missed detections, and prune the false alarms which happen occasionally.

Measures	false alarm	missed detection	track fragmentation	rate of overlap	AEC	AES
Dataset 1	5.3%	2.6%	2.36%	87.6%	5.5px	7.5px
Dataset 2	0%	2.4%	6.81%	96.3%	8.6px	11.8px
Dataset 3	7.8%	5.2%	0%	91.3%	19.5px	18.8px
Dataset 4	1.5%	2.7%	2.12%	93.6%	11.0px	18.0px

Table II: Evaluation of the multiple traget tracking results

V. Conclusion

In this paper, a previous work for on-road obstacle detection using stereo vision was improved by a reliable definition of traffic area and a multiple object tracking scheme based on particle filtering. Road traffic area extraction is first integrated in the detection process. On this ROI, connected-component extraction replaces Hough Transform for a flexible and fast detection of the obstacles. Moreover, multiple cues are considered to improve the detection accuracy. During the particle filtering, a self-adaptive gate for data association and dynamic filter noise function have been applied to enhance the tracking performance. Experimental results using a public dataset demonstrate that our detection and tracking system is efficient and reliable. Most obstacles appeared in 35m can well be detected and tracked. The proposed algorithm can work under dynamic circumstances without any prior-knowledge. Nevertheless, the use of 2D coordinates has a certain limit for further localization and tracking of the obstacles. Therefore, the next research step will be to focus on exploring the proposed approach from a 3D point of view.

References

[1] Andreas Ess, Konrad Schindler, Bastian Leibe, and Luc Van Gool. Object detection and tracking for autonomous navigation in dynamic environments. *The International Journal of Robotics Research*, 29(14):1707–1725, 2010.

[2] Roberto Manduchi, Andres Castano, Ashit Talukder, and Larry Matthies. Obstacle detection and terrain classification for autonomous off-road navigation. *Autonomous robots*, 18(1):81–102, 2005.

[3] Akihito Seki and Masatoshi Okutomi. Robust obstacle detection in general road environment based on road extraction and pose estimation. *Electronics and Communications in Japan (Part II: Electronics)*, 90(12):12–22, 2007.

[4] Amirali Jazayeri, Hongyuan Cai, Jiang Yu Zheng, and Mihran Tuceryan. Vehicle detection and tracking in car video based on motion model. *Intelligent Transportation Systems, IEEE Transactions on*, 12(2):583–595, 2011.

[5] Xue Fan, Shubham Mittal, Twisha Prasad, Suraj Saurabh, and Hyunchul Shin. Pedestrian detection and tracking using deformable part models and kalman filtering. *Journal of Communication and Computer*, 10:960–966, 2013.

[6] Mikel Rodriguez, Ivan Laptev, Josef Sivic, and J-Y Audibert. Density-aware person detection and tracking in crowds. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2423–2430. IEEE, 2011.

[7] Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system. *IJSCE, ISSN*, pages 2231–2307, 2012.

[8] Bihao Wang and Vincent Frémont. Fast road detection from color images. In *Intelligent Vehicles Symposium Proceedings, 2013 IEEE*, pages 1209–1214. IEEE, 2013.

[9] Bihao Wang, Vincent Frémont, and Sergio A Rodríguez. Color-based road detection and its evaluation on the KITTI road benchmark. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 31–36. IEEE, 2014.

[10] Zhencheng Hu and Keiichi Uchimura. UV-disparity: an efficient algorithm for stereovision based scene analysis. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 48–54. IEEE, 2005.

[11] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision-ECCV 2004*, pages 28–39. Springer, 2004.

[12] Hossein Tehrani Niknejad, Akihiro Takeuchi, Seiichi Mita, and David McAllester. On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation. *Intelligent Transportation Systems, IEEE Transactions on*, 13(2):748–758, 2012.

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[14] L.G. Shapiro and G.C. Stockman. *Computer Vision*. Prentice Hall, 2001.

[15] Michael Isard and Andrew Blake. Condensation-conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[16] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[17] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Computer Vision–ACCV 2010*, pages 25–38. Springer, 2011.

[18] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.