

Computer Vision at the Hyundai Autonomous Challenge

Pietro Cerri[†], Giacomo Soprani[†], Paolo Zani[†], Jaewoong Choi^{††}, Junyung Lee^{††}, Dongwook Kim^{††},
Kyongsu Yi^{††} and Alberto Broggi[†]

[†] VisLab, Dipartimento di Ingegneria dell'Informazione, Parma University,
Via G.P. Usberti, 181/A, 43124 – Parma, Italy

^{††}VDCL, School of Mechanical and Aerospace Engineering, Seoul National University,
599 Gwanangno, Gwanak-Gu, Seoul, 151-742, Korea

Abstract—As interest on autonomous vehicles is growing worldwide, different approaches, based on different perception technologies and concepts, are being followed. This paper exposes the importance of the use of vision technology in most of these approaches, and presents the experience of the SNUCLE autonomous vehicle which successfully completed the Hyundai Autonomous Challenge in November 2010.

I. INTRODUCTION

The driving task involves the sensing of the environment; humans rely mostly on vision, although other senses such as sound, vibrations, and even torque are also used. Vision provides a very rich set of information which -by itself- provides enough data to allow driving in almost every scenario. Indeed, data provided by the analysis of a video image -especially when the picture is captured by a low-cost sensor- is only a qualitative description of the world, but an appropriate analysis can lead to a sufficient quantitative representation of distances, speeds, and other morphological measurements.

Besides their undisputed richness in content, images require complex processings to extract the information they hold. And this is the main reason why in the field of autonomous or supervised driving vision had a slow start. Around the 80's or 90's, all available technologies were considered and installed on board of prototype vehicles, including vision, and in the following years different research groups have developed a serious interest into computer vision.

In the 2005 DARPA Grand Challenge, almost all of the vehicles that reached the end of the race were based on a sensing technology different from computer vision [1][2][3]. TerraMax [4] was the only vehicle whose primary sensor was through the use of cameras.

On the other hand, the rules of the DARPA Urban Challenge [5], in 2007, put some more pressure on the use of vision -for example for the detection of lane markings or stop lines- but new sensors were born which were able to pick the position of lane markings by using a lidar technology [6].

Further challenges, like the Hyundai Autonomous Challenge, in 2010, explicitly required the use of vision to detect patterns

such as pedestrian crossing zebras or stop lines. And the next version of this challenge, predicted for 2012, will require a massive use of artificial vision to detect visible features, such as variable message signs or pedestrians waiting on the road side.

The use of vision, however, is not mandatory in a future world in which the infrastructure will also be active: transponders installed in traffic lights and traffic signs, as well as into variable message panels, may complement the information available in an extremely detailed map, therefore enabling the full autonomy of vehicles.

A bright example of this approach is shown by the Google Cars [7] which are strongly based on the use of a map for static information, and the use of a lidar to detect dynamic objects. In this case vision is only used to detect the status of traffic lights that are not yet incorporating a beacon. This approach, although very successful, requires the availability of a very precise map, a very precise inertial platform, and a quite complex overhead sensor. Vision provides a quite similar capability to perceive the environment, at a fraction of the cost.

This paper is focused on the description of the vision systems (shown in figure 1) developed and installed on SNUCLE, a successful entry in the Hyundai Autonomous Challenge that took place in Seoul, South Korea, in November 2010.

The rules of the Hyundai Autonomous Challenge were simple. Contestants were required to build an autonomous vehicle capable of following a course and performing missions. A Hyundai Motors Grandeur TG has been used as the base platform of the SNUCLE autonomous vehicle. Three cameras and six LIDARs have been installed on SNUCLE to detect obstacles and the environment: two cameras are used to detect obstacles, bumps, and pedestrian crossings in front of the vehicle, and the other camera is used to detect lanes. Three networked computers and a DSP located in the back seat compose the computing system of SNUCLE: one computer is used for interfacing and logging, another one is used for processing vision data from the cameras, and the last one is used for gathering the high level data from LIDARs, vision and GPS and determining the path.

The Hyundai Autonomous Challenge required autonomous

vehicles to perform various missions, like avoiding obstacles, passing through gates and tunnels on the course. The autonomous vehicles were also required to stop in a specific region in front of a pedestrian crossing stop line and to slow down in front of speed bumps.



Figure 1: SNUCLE Vision Systems

II. VISION ALGORITHMS

A. Lane detection

The lane detection module is based on the visual processing of the image obtained by an Inverse Perspective Mapping (IPM)[9] applied on the image framed by the camera, under a flat road assumption. Using this algorithm it is possible to detect single and double lines, both solid and dashed, and to classify them into ego, right and left lanes. Figure 2 shows the algorithm flow chart.

As the IPM approach is affected by pitch variations, a pitch detection stage is performed before the actual IPM, as described by Medici in [11].

Two filters are applied to the IPM image to detect Dark-Light-Dark (DLD) patterns, or DLDDL patterns for double lane markings. These filters search for transitions in each row separately, considering the width of lane markings. The resulting pixels are clustered according to proximity using an expansion method. A polyline is finally extracted from each group of pixels using the Ramer–Douglas–Peucker algorithm [14]. At the end of the low level processing stage, a list of candidate polylines is obtained, each of them corresponding to a pattern that looks like a line in the IPM image.

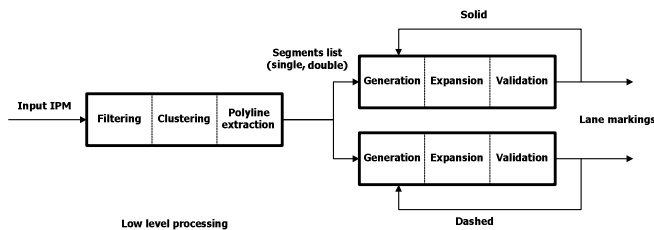


Figure 2: Lane detection flow chart.

A tracking stage is used to make the detection robust: each candidate polyline is matched against markings detected in the previous frame using the function described in [10], and the correspondence originating the lowest distance value is considered as valid; a similar method is also used for dashed lines, albeit using a different distance metric: in this situation both the distance of the dash from the tracked marking and the distance of the dash from the ego-vehicle are evaluated, in order to compensate for the uncertainty arising from the analysis of short segments, as explained in [10].

All the polylines, tracked and non-tracked, are then expanded to obtain longer lines, considering any compatible candidates according to proximity and orientation. At the end of this stage a score is assigned to each polyline considering its length and distance from the ego-vehicle; if the polyline has been successfully tracked, the old score is also added to the current. Polylines with a score above a fixed threshold are finally accepted as valid lane markings.



Figure 3: Algorithm output and processing images: IPM image (top left), candidate polylines (top center) and detected lane markings (top right); original image with detected lane markings superimposed (bottom).

The lane detection algorithm runs on the VELD [8] embedded system, equipped with a Texas Instruments TMS320DM642-600 600MHz DSP capable of 4800 MIPS. Figure 3 shows the lane detection output, although when installed on board, the results (lanes position) are made available via Ethernet, and the graphical output is disabled.

B. Pedestrian crossing and stop line detection

The pedestrian crossing algorithm is also based on the IPM image. First a stabilization stage, based on histogram analysis

[12], is performed; a Sobel filter is then applied to detect edges which appear almost vertical on the IPM image, as they correspond to borders which lay on the pavement approximately parallel to the vehicle direction. Border direction, i.e. transition signs, is saved to be used in the following step to distinguish between dark-light transitions and light-dark ones. Figure 4 presents the results of the filter.

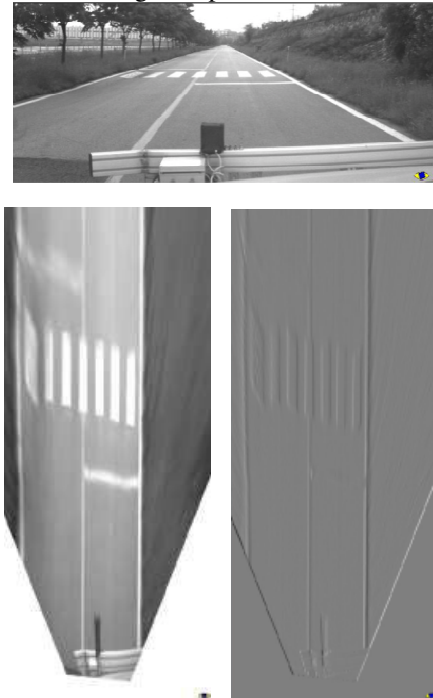


Figure 4: Original image, IPM image and vertical Sobel filter. Black and white pixels represent dark-bright and bright-dark transitions respectively.

Connected pixels are clustered to generate lines with a length compatible with pedestrian crossing stripes: two lists of lines are generated, one containing the left borders of stripes, the other containing the right ones. A left line and a right one are joined together to generate a stripe if their distance is compatible with the stripe width, and their orientation is approximately the same (see Figure 5).

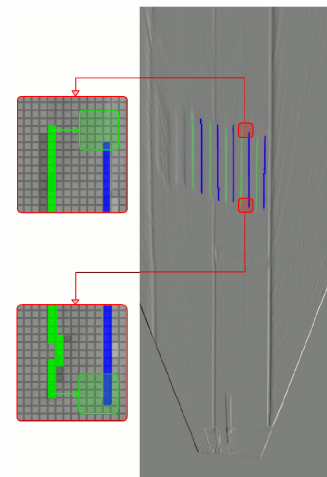


Figure 5: Adjacent lines joint into a single stripe.

Aligned stripes are then grouped together to generate a pedestrian crossing considering rules of orientation and of minimal and maximal distance to guarantee the typical zebra shape, with alternate dark and bright areas: if at least three stripes are found, the pedestrian crossing is considered as valid.

In the case of missed-detection of one stripe, the pedestrian crossing algorithm can fail: in order to cope with this problem a method to merge together two groups of stripes has been developed. This is performed by checking whether between the last stripe of the left block and the first one of the right block there is enough space: in this case these stripes are checked to have a similar orientation and their luminance is similar to the average of the luminance of the area where the stripe is missing.

Figure 6.a shows the iterative process of grouping stripes into a pedestrian crossing, while Figure 6.b shows the search area used to merge two groups of stripes.

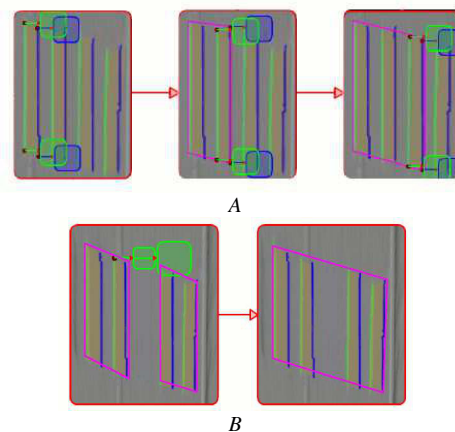


Figure 6: a) stripes grouping and b) missed stripe.

Figure 7 shows pedestrian crossing detection results: the distance is easily computed in the IPM image, considering the closer point of the detected zebra.

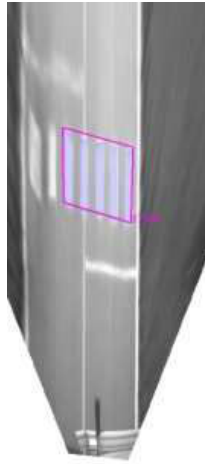


Figure 7: Pedestrian crossing detection result.

A similar algorithm has been developed to detect stop lines: a stop line is represented in the IPM image as a single horizontal stripe, therefore horizontal borders are searched for and then paired. This algorithm reaches an adequate detection rate with a reduced number of false positives, but, unfortunately, it is not able to detect stop lines farther than 15 meters, due to the sharpness degradation of horizontal borders in the IPM images; to cope with this problem, a different algorithm has been developed. The underlying idea is the same: to search for horizontal edges and merge two candidate lines, but this time the processing happens on the original image. This algorithm is capable of detections up to 25-30 meters but it is more prone to false positives, especially at higher distances. Results are shown in Figure 8: the distance of detected lines is computed using a simple transformation considering camera orientation.

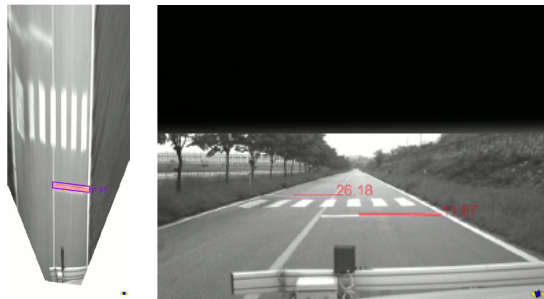


Figure 8: Output of the stop line detection algorithms.

C. Speed bump detection

Speed bumps detection is a non-trivial problem since laser scanners have serious issues in locating small objects (around

5 cm) on the ground. Even human drivers would have difficulties in locating speed bumps if only shape were used: to enhance their visibility, speed bumps are painted with a specific color and pattern, which are also used by our vision system to precisely detect them.

Color detection is a well known technique and a lot of approaches are present in literature. A method based on RGB image processing (avoiding color coordinate conversions, which are time consuming) has been developed and used to detect the yellow stripes composing the bump: ratios between the three color channel are checked to be between given thresholds; after this operation, a morphological erosion is performed to remove isolated pixels and speed up the following clustering step. These filters are applied on the lower part of the image, to focalize the attention only towards the road. Figure 9 shows the clustering results.

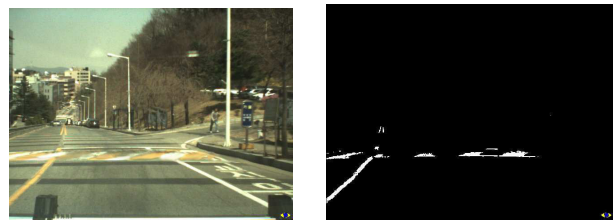


Figure 9: Original image and detected yellow pixels

Adjacent pixels are grouped together to generate a block. The lower part of the image is divided into squares of fixed size, then yellow pixels are searched for in each area separately. A group of adjacent blocks found in an area is validated if its width and height are greater than a threshold. Candidate groups are then joined together considering the distance between closer extremes.

The image is then separated into horizontal stripes in which the two extrema of the blocks are searched for: in each row the average column of the detected stripes is computed. Using this information three other values are computed:

- A = Maximum distance of farthest blocks;
- B = Maximum gap between two subsequent blocks;
- C = Distance between the average column and the two closest blocks.

Figure 10 shows these three values in two different cases: in the first one the C value is 0 because the average column of the row is included into a block, in the second case the C value is appropriately computed.

The A value is used to check whether the detected object is large enough to represent a speed bump: yellow vehicles or road structures can be detected as bumps if this check is not performed, while the B and C values are used to avoid the validation of small yellow objects, lying approximately at the same distance, on the sides of the road.

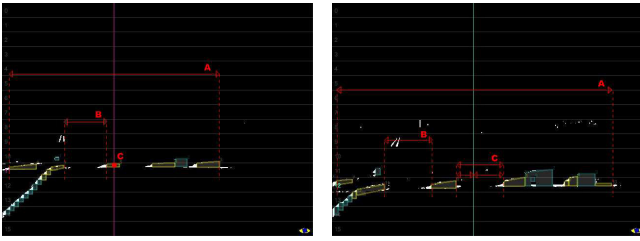


Figure 10: Values considered to evaluate the bump detection. Yellow areas refer to blocks which are considered inside the bump.

III. RESULTS

The developed algorithms were tested in different scenarios: in urban environments (both in Italy and in Korea), in the VDCL test facilities, and finally on the challenge track.

The detection rate on test and race tracks was almost perfect for all the algorithms, with no false positives, except for the stop line detection system, that generates a considerable number of false positives in specific scenarios (off-road, sharp curves), as a matter of fact stop lines have a simple pattern, that can be generated also by shadows, road structure, or vehicles. For this reason the stop line detection system was disabled during the race, considering also the fact that the only stop line was just before a pedestrian crossing. Stop line detection needs a serious revision, even if in simple cases it is effective.

The speed bump detection system was successfully tested in Korean urban scenarios obtaining no false positives and a high detection rate; the same results were reached for pedestrian crossing detection both in Italian and Korean environments. Detection range is around 20-25 meter for speed bumps and 25-30 meters for pedestrian crossings. Unfortunately, even if the stabilization stage reduces the errors, the computed distance can be affected by pitch variations: to detect the exact position of the pedestrian crossing, in order to stop the vehicle in the correct position and avoid penalty during the competition, the speed was set to be low in the areas of the pedestrian crossing.

Processing times of the three algorithms on an Intel Quad CPU Q9550 @ 2.83GHz range from 40 to 80 ms, with an average of about 50 ms.

Pedestrian crossing detection algorithm is the most robust one: it performed perfectly during the race, using a set of thresholds and parameters based on the pedestrian crossing present in the route. During tests on Italian urban roads, using a set of loose thresholds tuned on Italian regulations, the results are still good: no false positives and a low number of miss-detection, mainly due to scratched stripes (the detection rate is about 82%, while reaches 92% excluding ruined zebras). It is interesting to note that in one case the pedestrian crossing did not get detected because the length of the stripes were below the minimum required by current Italian laws!

Lane detection system results were already presented in [10]; on the race track, with clear lane markings, the detection was extremely robust.

Processing times on the VELD platform range from 60 to 90 ms, with an average of about 80 ms.

IV. SNUCLE BEHAVIOR AT THE H.A.C.

The developed algorithms were deployed on SNUCLE and tested on the challenge track as mentioned above. Using the vision systems described in this paper, SNUCLE was successful in driving autonomously, and correctly negotiated the lanes, pedestrian crossings, stop lines, and speed bumps present on the *on-road* course of the challenge track, shown in Figure 11.



Figure 11: The *on-road* course of the challenge track is indicated with a red line on the map.

A. Lane Detection

Using the lane position information provided by the VELD system, the desired path was generated by a least squares filtering method as shown in Figure 12.

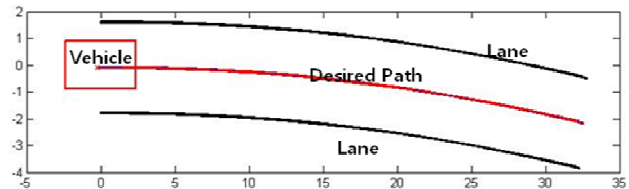


Figure 12: Desired path generation with VELD

To keep the vehicle within the lane, a bicycle vehicle model was used, and the lateral position error and yaw angle error between the vehicle and the desired path were defined with respect to the desired path. Results of lane keeping performance at the Hyundai Autonomous Challenge are shown in Figure 13. Figure 13.b shows the lateral offset between real and desired path, that was maintained below 0.2m. Figure 13.c shows the angle applied to the steering wheel.

Figure 14 shows SNUCLE while performing lane keeping at the Hyundai Autonomous Challenge.

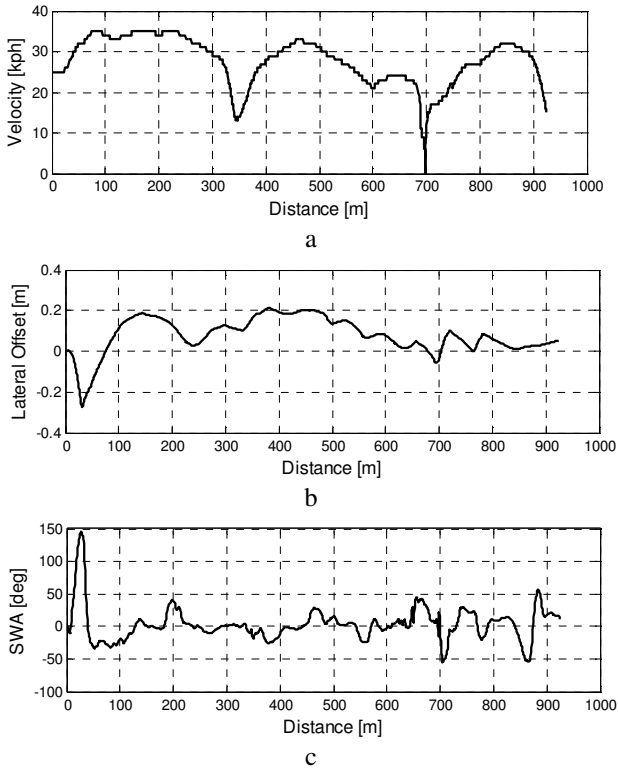


Figure 13: Results of lane keeping performance



Figure 14: SNUCLE performing lane keeping

B. Pedestrian crossing and stop line detection

SNUCLE was also required to stop in a specific region in front of the pedestrian crossing: speed was kept under 35km/h on the *on-road* part of the track because the pedestrian crossing location was not provided a-priori, and the performance of pedestrian crossing detection was stable only under 35km/h. As mentioned above, the precision of the pedestrian crossing localization can be affected by pitch variations. To guarantee a robust performance with respect to any noise by pitch variations or processing delay, a Kalman filter was used to estimate the distance. Figure 15 shows the results of SNUCLE behavior using the data of pedestrian crossing.

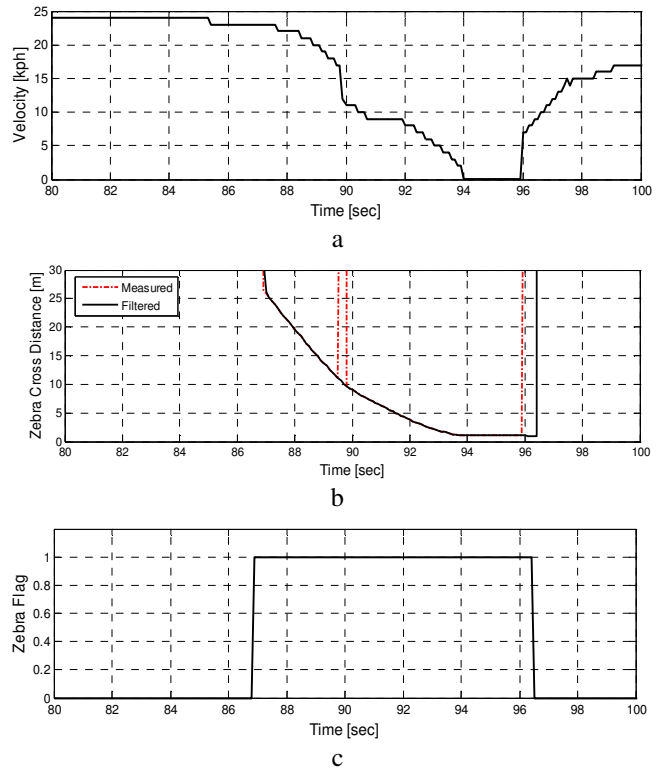


Figure 15: SNUCLE behavior when detecting a pedestrian crossing

As shown in Figure 15.a and Figure 15.b, SNUCLE began to decelerate as it approached the pedestrian crossing. Because of the rapid deceleration, the pedestrian crossing was missed during a short period; therefore an estimated distance was used to decelerate and stop SNUCLE at about 1 meter in front of the pedestrian crossing for 2 seconds to accomplish the mission.

C. Speed bump detection

Speed bumps detection was needed because the Hyundai Motors Grandeur TG used as the base platform of SNUCLE is a sedan, and the height of its frame is quite low, so there was a concrete risk of damaging the sensor systems installed on it when passing over a speed bump at high speed. SNUCLE was therefore required to decelerate when approaching a speed bump. To decelerate the vehicle properly, a robust speed bump detection need to be guaranteed. In order to provide the necessary performance, a Kalman filter was used to estimate a distance between the speed bump and the vehicle using measurements by image processing as shown in Figure 16.b. The sharp vertical line after passing the speed bump indicates the transition to the default distance value that is produced when no speed bump is observed, as shown in Figure 16.b. As shown in Figure 16, SNUCLE began to decelerate when the speed bump was detected and it passed speed bump at a speed

of 14km/h.

V. CONCLUSION

The SNUCLE autonomous vehicle, presented in this paper, was equipped with a perception system consisting of cameras, LIDARs, and a GPS unit. The developed systems have proven to be effective in most cases, and will be extensively tested and eventually refined to be ported into a unified perception architecture [13]; they will also be used as the starting point for the new Hyundai Autonomous Challenge that will take place in autumn of 2012. New difficulties will be present: traffic sign and traffic light recognition, pedestrian detection, and parking slot detection, all of them to be handled through the use of computer vision. The competition could also be the chance to test the stereo obstacle detection system that was already developed for the previous challenge in 2010, but was not integrated because of lack of time.

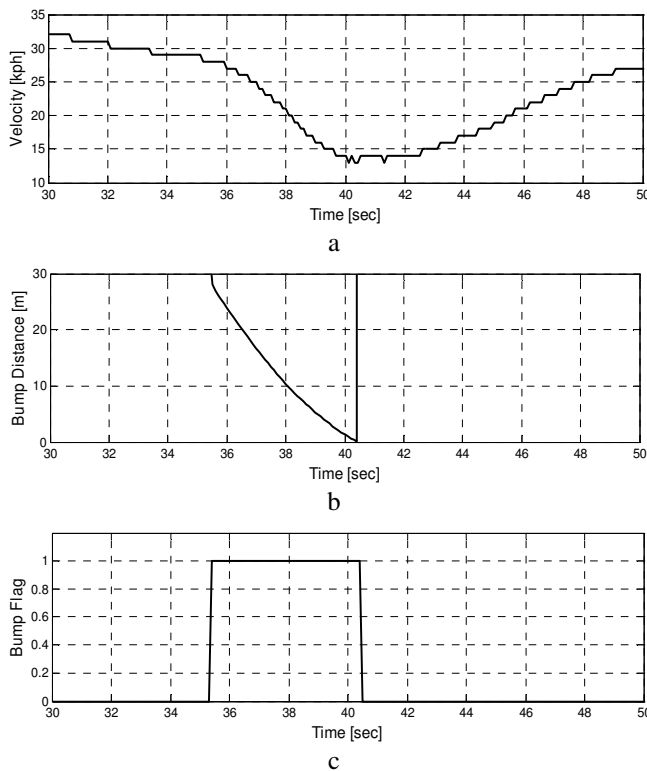


Figure 16: SNUCLE behavior when detecting a speed bump

ACKNOWLEDGMENT

This research was jointly supported by SNU-IAMD, BK21 program, the Korea Research Foundation Grant funded by the Korean Government (MEST) (KRF-2009-200-D00003), and the National Research Foundation of Korea Grant Funded by the Korean Government (2010-0001958).

REFERENCES

- [1] Redmill, K.A.; Martin, J.I.; Ozguner, O., "Sensing and Sensor Fusion for the 2005 Desert Buckeyes DARPA Grand Challenge Offroad Autonomous Vehicle", In Procs. IEEE Intelligent Vehicles Symposium, 2006.
- [2] Thrun, S., "Winning the DARPA Grand Challenge: A Robot Race through the Mojave Desert", 21st IEEE/ACM International Conference on Automated Software Engineering, 2006. ASE '06.
- [3] Behringer, R.; Travis, W.; Daily, R.; Bevely, D.; Kubinger, W.; Herzner, W.; Fehlberg, V., "RASCAL - an autonomous ground vehicle for desert driving in the DARPA Grand Challenge 2005" In Procs IEEE Intelligent Transportation Systems, 2005.
- [4] Stefano Cattani, Pietro Cerri, Paolo Grisleri, "Perception and Data Fusion on Autonomous Vehicles: the TerraMax(TM) Experience", Procs. Intl. Conf. VISION 2008.
- [5] "Special Issue on the 2007 DARPA Urban Challenge, Part I-III", Journal of Field Robotics, Wiley Blackwell, Volume 25, Issue 8-10, 2008.
- [6] Kammel, S.; Pitzer, B., "Lidar-based lane marker detection and mapping", In Procs. IEEE Intelligent Vehicles Symposium, 2008.
- [7] <http://www.nytimes.com/2010/10/10/science/10google.html>
- [8] <http://vislab.it/Products/view/46/VELD>
- [9] M. Bertozzi, A. Broggi, and A. Fascioli, "Stereo Inverse Perspective Mapping: Theory and Applications," Image and Vision Computing Journal, vol. 8, no. 16, pp. 585-590, 1998.
- [10] M. Felisa and P. Zani, "Robust monocular lane detection in urban environment," In Proceedings of IEEE Intelligent Vehicles Symposium, San Diego 2010, pp. 591-596.
- [11] M. Bertozzi, L. Bombini, P. Cerri, P. Medici, P. C. Antonello, and M. Miglietta, "Obstacle Detection and Classification fusing Radar and Vision," in Procs. IEEE Intelligent Vehicles Symposium 2008, Eindhoven, Netherlands, June 2008, pp. 608-613.
- [12] Luca Bombini, Pietro Cerri, Paolo Grisleri, Simone Scaffardi, and Paolo Zani, "An Evaluation of Monocular Image Stabilization Algorithms for Automotive Applications", In Proceedings. IEEE Intl. Conf. on Intelligent Transportation Systems 2006, pages 1562-1567, Toronto, Canada, September 2006.
- [13] Luca Bombini, Stefano Cattani, Pietro Cerri, Rean Isabella Fedriga, Mirko Felisa, and Pier Paolo Porta, "Test bed for Unified Perception & Decision Architecture", In Procs. 13th Int. Forum on Advanced Microsystems for Automotive Applications, Berlin, Germany, May 2009.
- [14] Urs Ramer, "An iterative procedure for the polygonal approximation of plane curves", In Computer Graphics and Image Processing, vol 1, num 3, pp. 244-256, 1972.