Multi-sensor based object detection in driving scenes

Philippe Xu

MULTI-SENSOR BASED OBJECT DETECTION IN DRIVING SCENES

Internship report, Research Master's Degree in Computer Science
ENS Cachan/Rennes and University of Rennes 1

Internship dates : February-June 2011
At Peking University, LIAMA, Beijing, China
Key Laboratory on Machine Perception

PHILIPPE XU
ENS Cachan/Rennes
University of Rennes 1

Advisors :

FRANCK DAVOINE
Researcher, CNRS
LIAMA, Sino-French Laboratory

HUIJING ZHAO
Professor, Peking University
Key Lab on Machine Perception

Abstract

The work done in this internship consists in two main part. The first part is the design of an experimental platform to acquire data for testing and training. To design the experiments, onboard and onroad sensors have been considered. A calibration process has been conducted in order to integrated all the data from different sources. The second part was the use of a stereo system and a laser scanner to extract the free navigable space and to detect obstacles. This has been conducted through the use of an occupancy grid map representation.

Keywords: Stereo camera, Sensors calibration, Occupancy grid map, Obstacle detection

June 2, 2011

DÉTECTION MUTLI-CAPTEURS D'OBJETS EN SCÈNE DE CONDUITE ROUTIÈRE.

Rapport de stage, Master en Informatique
ENS Cachan/Rennes et Université de Rennes 1

Dates de stage : Février-Juin 2011
À l'Université de Pékin, LIAMA, Beijing, Chine
Key Laboratory on Machine Perception

PHILIPPE XU
ENS Cachan/Rennes
Université de Rennes 1

Encadrants :

FRANCK DAVOINE
Chercheur, CNRS
LIAMA, Laboratoire Franco-Chinois

HUIJING ZHAO
Professeur, Université de Pékin
Key Lab on Machine Perception

Résumé

Le travail effectué durant ce stage se compose principalement en deux parties. La première partie du travail a été l'élaboration d'une platforme expérimentale dans le but d'acquérir des données de test et d'apprentissage. Les expériences ont mené à considérer des capteurs embarqués mais aussi d'autres placés sur la route. Pour pouvoir intégrer toutes les données issues de différentes sources, une calibration des capteurs a été nécessaire. La deuxième partie du stage a consisté à utiliser un système de caméra stéréo ainsi qu'un laser pour extraire l'espace navigable et aussi pour détecter des obstacles. Cela a été fait en considérant un grille d'occupation comme représentation des

Mots-clés: Caméra stéréo, Calibration de capteurs, Grille d'occupation, Détection d'obstacle

2 Juin 2011

Contents

Introduction

Intelligent and autonomous vehicles field is a very popular yet challenging research topic. Their applications are numerous and can be of primordial importance. A case like pedestrian safety can have the potential of saving a lot of lives. Traffic accidents being one of the major causes of death around the world, the impact of intelligent vehicle can be very important.

Totally autonomous vehicles still seems out of reach in complex environment like within a crowded city. Even though very impressive achievements have be reached in events like the DARPA grand challenge. Aside from autonomous vehicles, intelligent vehicle used in a context of assistance for the driver is also an important research topic.

To assist the driver, the vehicle need to perceive and analyze the surrounding environment and then let the driver know in case of danger. The concept of danger is directly linking to the presence of obstacles like cars or pedestrians. Thus a lot of works have been done in detecting pedestrian as well as cars. However, all kind of objects can be considered as dangerous, animals or various kind of obstacles can represent a danger. It is not possible to built a specific detector to all possible kind of obstacles, thus generic obstacle detection will be considered in our work.

The first part of our work consisted in designing an experimental platform to capture data for training and testing. Different kind of sensors calibration has thus been done. The study was first focusing on the analysis of road intersection, which is potentially dangerous.

Then our study focused on the use of a stereo camera system to compute an occupancy grid map from which free space and obstacle detection were carried out. The fusion with a laser scanner is also considered. This work was continuation of our previous work [32] in which laser data was projected onto the camera frame. Now the inverse approach is actually considered as we will use the stereo system to be able to project the image to the laser frame.

The work done during this internship was also part a new Sino-French scientific collaboration between Université de Compiègne and Peking University. One goal of this collaboration being to test different kind of algorithms in very different environment where drivers and pedestrians may behave very differently.

1 Literature Review

Object detection for intelligent vehicles is of primordial importance. Two kind of approaches can be seen in the literature. The first kind can be seen as object discriminative detection. It covers the detection of specific object such as pedestrians or cars. The second group will grasp approaches which will detect generic object considered as important like moving objects or salient object that are very different from the background.

1.1 Object discriminative based detection

For intelligent vehicle, extensive effort has been put on object detection. And it is especially true for pedestrian detection. Because pedestrian safety is the focus of many application, a lot work has been done toward this goal. It remains however a very challenging task and is still at the center of a lot of researches.

A main part of human detection is done in the field of computer vision. This is because for targets as complex as human beings, one needs very discriminative information to differentiate it from other objects. Range sensors like laser are sometime not enough to discriminate human from structures like trees.

Monocular pedestrian detection Many surveys, experiments and benchmarks have appeared in the literature like [8, 10, 22, 26]. More and more datasets have also appeared with increasing difficulties each time. The INRIA dataset [6] has been largely used for person detection. It however tackles the case of quite high definition images with non-occluded person only. More recently, the Caltech Pedestrians [8] and the TUD-Brussels [31] benchmarks have provided a very large and challenging set of data especially designed for monocular pedestrian detection in driving scenes. Fig. 1 shows some samples from those datasets.

(a) (b) (c) (d)

Figure 1: Samples from INRIA (a), (b), Caltech (c) and TUD-Brussels (d) datasets

Pedestrian detection in monocular images is often based on a sliding window approach. Because the size and the position of the person is unknown, the image is densely scanned over a large set of scales and positions. Then for each window some features are extracted and a classifier is used to decide whether it is a person or not.

One of the most popular feature is the histogram of oriented gradient (HOG) introduced by Dalal and Triggs [6]. A window of the image is cut into cells in which a histogram is built upon the orientation of the gradient. All histogram are normalized and concatenated to form the final feature vector. An overview of this features extraction is shown on Fig. 2. Then this vector is used for classification. A classifier like a linear SVM is well adapted. This HOG approach was the detector used in our previous work on pedestrian detection [32].

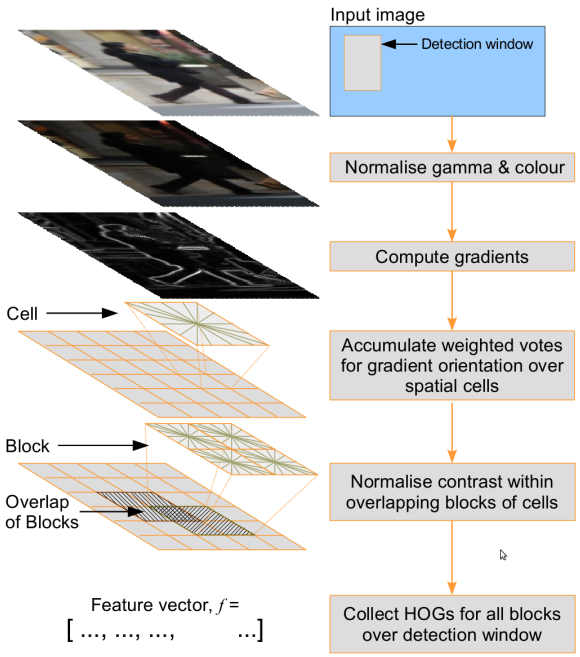


Figure 2: HOG computation over a dense grid. Image taken from [5].

Because the sliding window method need to compute the features for many windows, it is very computationally demanding. However, the use of GPU implementation like [23] enables such approach for real time application.

A lot of works have contributed to improve the pioneer work of Dalal and Triggs by integrating new features or introducing new classifiers. Association with new information like motion through histogram on the orientation of optical flow has proved to increase the performance [5, 28]. The recent work of Walk et al. [28] has shown a good improvement with the consideration of color self-similarity and local binary pattern features [30]. While using intersection kernel SVM [19] has also shown better performance compared to normal linear SVM.

Pedestrian tracking Detection only will often fail in case of strong occlusions or even partial occlusions. On way of increasing the reliability of a pedestrian detection system is to associate it with a tracker. There exists a lot of tracking methods for pedestrian tracking but the case of urban driving scenes is much more challenging compared to static camera configurations such as in a video-surveillance context.

For static camera, impressive results had been achieved by Song et al. [27]. In their study, the use of a very simple background subtraction was enough to do the detection step. Once a object is detected, an individual model of it is learned in an online way. They sampled patches over the detected bounding boxes then extracted color and texture descriptor, the set of patches are then used as training sample for learning. The negative samples being the ones from other detected windows. Such models make it possible to differentiate the different tracked person while strong pedestrian model can only discriminate person from non-person image.

For the tracking stage, they used a color-based tracking method with particle filter. To

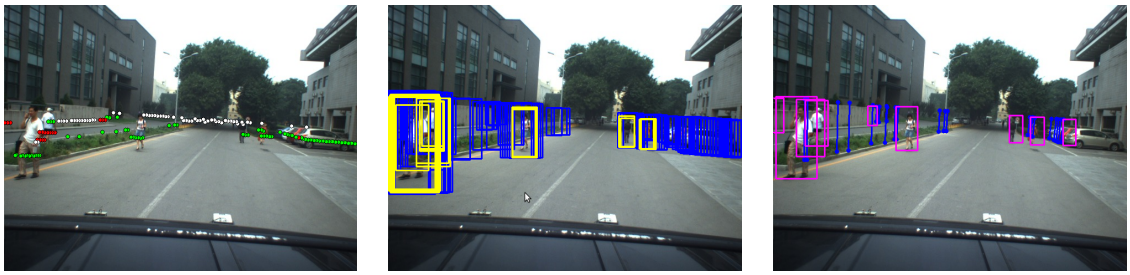
handle occlusion, they introduced the notion of correlation between targets. In the case where a target in not correlated with any other targets, meaning that it is spatially isolated from the others, it is tracked by an independent tracker. But when multiple targets get correlated, meaning that they are close to each other, thus prone to occlusion, then the targets merge to form a new target which will be tracked as a new whole entity. When this new merged target finally split up again, the appearance model learned in earlier stage in used to recognize each newly non-correlated targets.

A direct use of this approach in a moving camera context is not that easy. Especially because we have no direct access to a moving object segmentation. Moreover, the use of a person detector cannot detect the larger region formed by the merging of two correlated pedestrians. However, the use of stereo-cameras can give information on the presence of obstacles. Thus an extension of their method could be built using a sliding window approach for initial detection and stereo obstacle segmentation when multiple targets merge together.

Another efficient approach for pedestrian tracking has been proposed recently by Ess et al. [11]. In their case, the detection was done by a simple HOG detector. They also used an appearance model to discriminate each target from each other, thus avoiding unlikely association. They used a very simple appearance model, they used a color histogram computed inside an ellipse fitted to the bounding box.

Their tracking system is a bit more complex than the method of Song et al. [27] as they do reasoning in the 3D space, with range information retrieved by a stereo system. Their tracking is based on an Extended Kalman Filter approach using a proper motion model. They actually over-sample trajectory prediction and try to find the one that explain best the targets trajectory w.r.t. their history. In their approach, occlusion is handled by using an occupancy grid map where non-visible regions are explicitly represented. Whenever a target gets in one of them, meaning that it will be occluded (thus prone to miss detection), its trajectory is extrapolated using its motion model. When a predicted reappearance stage is reached, the tracker try to track back the target and recognizing it using its learning appearance model.

Other sensors based detectors The use of one monocular camera is sometime not enough to achieve good enough results. Thus, the integration of others sensors can be of a great help. One of the most used sensor in robotics is the laser range finder. It consists on sending one or multiple layer of laser beam then computes the range using the time-of-flight information. The main advantage of a laser scanner is its capability to provide very accurate range information. However, the resolution is often limited, typically 0.5 degree, which leads to the fact that very few beams actually hit a target if it is a bit far. Due to this sparse representation, a single layer laser scanner is not suited to detect directly pedestrian. Rather it will be used to get region of interest to be used with detector like HOG. In our previous work [32], we have used a laser range finder to first find some potential obstacles then the region of interest is given as an input to the visual detector. Fig. 3 shows same results from our previous work. The use of laser ROI leads to several benefits. First of all they reduce the computational time by limiting the search space of the visual sliding window. It also increases the detection rate for small targets which are usually ignored by the sliding window as it will be too computationally demanding to scan the image at a very small scale. Finally it also helps rejecting some false detections where we know from the laser that there is obviously no obstacle. However, it can also lead to additional miss detections because if a target is not within the generated ROI, then it will not be considered by the visual system.



(a) (b) (c)

Figure 3: (a) The laser points are classified to detect potential pedestrian, represented as green dots. (b) Region of interest are generated by considering the pedestrian having a approximate height of 1.8 meters. (c) HOG detector is run over the generated ROI.

The use of the 3D depth information turns out to be of important use. Having this consideration, a lot researches have also been done considering stereo camera systems. Using two cameras observing the same space at different positions gives access to the depth information by simple triangulation. In some works like in [12], the stereo is only used for visual confirmation after generating ROI from a laser scanner. Bansal et al. [2] used the stereo 3D information with a 3D template to get ROI from the stereo depth map. Additionally, the depth information can be used to estimate the ground plane. This information can then be used to reject false positive by adding the constrain that all pedestrian should lie on the ground.

Evaluation method Pedestrian detection problem in the image space has been well defined and evaluation methodology have been proposed such as in [28]. To evaluate the efficiency of a detector we will usually consider the number of false positive, the detection rate and the miss detection rate. Even if this kind of evaluation has been largely used in intelligent vehicle related work, it may not be the ideal measure for method comparison. This kind of evaluation is actually well suited for tasks like photo collection classification. And it is thus used in the PASCAL challenge for example.

However, in the context of road safety, all the target do not have the same importance. Not detecting a very far or off road pedestrian should be considered less important than not detecting a pedestrian crossing just in from of the vehicle. Furthermore, object discriminative detector is only evaluated for the considered target. Similar work can be done to detect a car or a bicycle. But in the case of driving scene, we are rather interested by detection a potentially dangerous target rather than knowing what it is.

Thus a evaluation over one specific object detection like pedestrian may be too limited in application like driving safety. That's why a second kind of object detection will have to be considered.

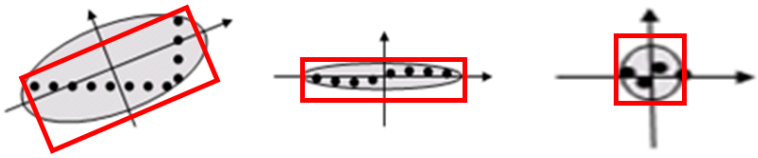
1.2 Object none discriminative based detection

As concluded in the previous part, we are trying to detect generic obstacle rather than specific target. We are now not interested in detecting potential dangerous object without trying to know what exactly it is. This goal can be seen as separating objects from background. One important consideration can be that moving object are the main targets to focus on.

Mapping based approaches In this context, the use of mapping technique can be considered. The mapping task is to construct a map of the environment which is composed of all the background or static structures. Objects not belonging to this map can then considering a obstacles. Mapping as well as localization has been widely studied in robotics within the Simultaneous Mapping And Localization (SLAM) framework [9]. The usual approach is for mapping is the use of a stochastic occupancy grid map. Using a Bayesian framework a map is generated, accumulating occupancy likelihood over time naturally filtering out moving object. Specifically tracking moving object within a framework SLAM is know as SLAMMOT introduced by [29]. More recently, a credibilist approach using Dempster-Shafer theory was introduced [15]. Here the moving object are extracting as conflicting case where at different time a grid cell is detected as free and occupied at other time or the inverse.

Based on the same approach, similar work can be done using a stereo camera system by projecting the 3D points cloud onto a 2D map. Kohara et al. [16] first virtually transfer the stereo disparity map onto the ground frame the project along the vertical axis to get a measurement comparable to the one got from laser.

One typical failure case for object detection using mapping techniques is that static objects are always considered as being within the background. This can be an important issue for driving safety application. One simple way to tackle this problem is to detect small structure objects. Typical background structures are large object like buildings. Those objects will typically be detected on a laser data by extracting long lines. To do so, a clustering stage is necessary. Laser points being close enough to each other are grouped into an unique cluster. Then depending on the size and form, it can be classified differently. In our previous work [32], we have for example considered three kind of clusters like illustrated on Fig. 4.

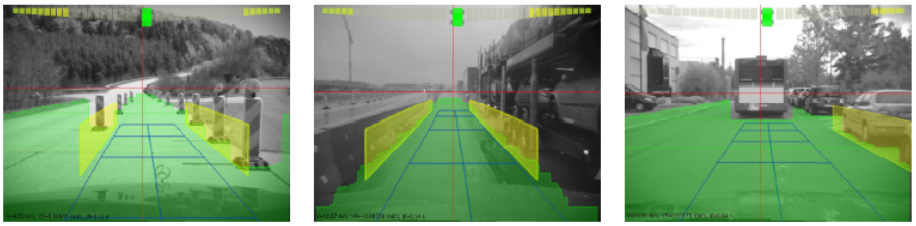


(a) 'L' type (b) 'I' type (c) 'O' type

Figure 4: Different laser points clusters types. 'L' type is typical for cars and trucks, 'I' for bicycles and 'O' for humans

In the same way some object detection can be done using stereo vision. In some special space like the U-V-disparity space, where we represent an image regarding the pixel column or row and their disparity, vertical structure are projected as line segment. In [13] the U-V-disparity is used to detect on road structures and obstacles. Badino et al. [1] considered only the free space in front of the vehicle using a stereo system. They detect for this the first obstacle considering a occupancy grid map generated by stereo vision. Fig. 5 shows some example of free space computation from [1]. Our use of a stereo camera will be based their approach.

Saliency based approaches Another kind of generic object detection is the use of the saliency concept. An object is considered salient if it pops out from the background. The human vision system is sensitive to this concept as those silent object naturally attract the attention of the viewer. The importance of those object can be great as unexpected object may be the source of danger.



(a) Highway. (b) Freeway. (c) Downtown.

Figure 5: Free space analysis from [1].

A famous work on saliency was done by Itti et al. [14]. More recently Liu et al. [18] proposed a general approach by fusing numerous features. They used a Conditional Random Field (CRF) framework to integrate numerous features as contrast, local color histogram and well as motion features among others. Fig. 6 shows some examples of outputs from those kind of approaches. The main advantage of this kind of approach is that they are very

Figure 6: The first row is the raw input image. The second row represent the saliency computed by Itti et al. [14]. The last row is the results from Liu et al. [18]

generic and can be used to detect all kind of objects.

The direct use of saliency based methods is not possible as too many unnecessary objects could be detected. Michalke et al. [21] have proposed a saliency related work for driver assistance. In their framework, two kind saliency estimation is done. They are referred as bottom up (BU) and top down (TD) saliency map. In the BU path, the same concept as described previously is done, that is we try to extract all possibly important objects. In the

TD path, however, we will consider a task dependent approach, that is one kind of object will be specifically design as salient. Then the fusion of both path will give response to the current task, like detection pedestrian, but will also consider generic object detect through BU saliency detection. Their idea is inspired human biological focus of attention which will be influence by the definition of a task. Fig. 7 shows some examples of outputs from their kind works. The upper part shown the response from the BU and TD paths and their combination. Here the BU part detects various objects while the TD part is tuned to detect bicycle.

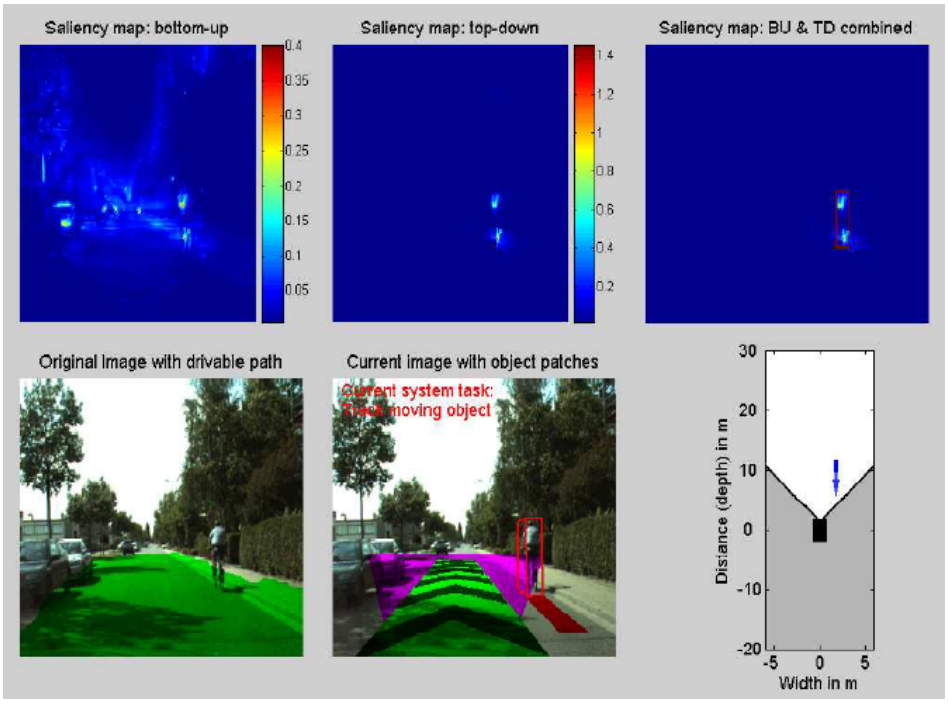


Figure 7: Example of results from [21]

2 Testing Platform and Data Acquisition

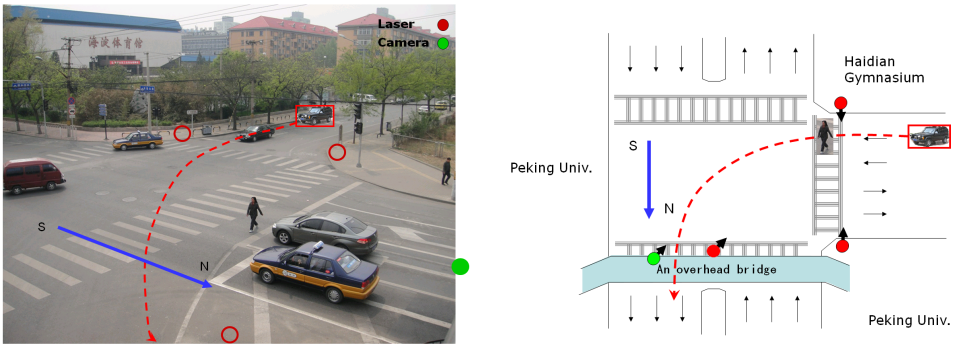
In order to develop new algorithms for intelligent vehicle application, we need to design an experimental platform. This work is considered within a collaborative framework between the Key Laboratory on Machine Perception of Peking university (PKU) and Heudiasyc from the Université Technologique de Compiègne (UTC). One goal of this collaboration is to be able to test different kind of algorithm in very different scene context. The quality of some approaches can be influenced by human habits. How people drive or how pedestrian cross the streets may vary greatly depending on which environment we are in.

2.1 Test-bed vehicle

Both laboratories have their own test-bed vehicle with different kind of sensors onboard. Fig. 8 shows the two vehicle considered in our study. Both laboratories have many sensors

(a) (b)

Figure 8: (a) Heudiasyc vehicle. (b) Peking university vehicle.



(a) (b)

Figure 9: Red circles represent stand alone laser scanner and the green circle represent the camera used for groundtruth generation

at their disposal like a Velodyne laser scanner with 360 degree field of view or Ladybug 3 Spherical Vision composed of 6 cameras to achieve a $360 * 135$ degree field of view. However, those high quality sensors are mainly used for ground truth generation, them direct use being too computationally demanding.

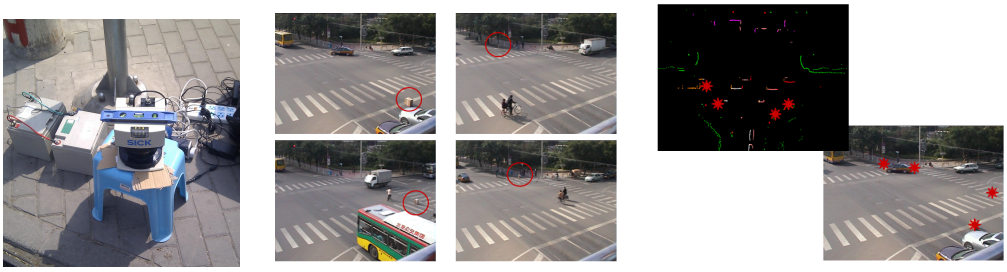
The main sensors that will be used will be stereo camera system and a laser scanner. A Videre stereovision system at UTC and own made stereo system based on two Flycap CCD camera at PKU. As for laser scanner, the Heudiasyc laboratory used a four layer laser whereas a single layer scanner was used in the Chinese part. Additionally, a GPS receiver is also used on both vehicle to get the vehicle position.

2.2 Experimental design and data acquisition

In our experimental design, we have decided to focus on a road intersection. This choice was motivated because it is a very challenging environment with vehicle coming from different direction of pedestrian crossing the road. Intersections are also dangerous places where an intelligent vehicle should be able to provide help to the driver.

The intersection is depicted on Fig. 9. In other to have a ground truth for our data like the obstacles' position as well as the host vehicle's position w.r.t a common world coordinate

system we use three stand alone laser scanners. Additionally another camera was installed at a high position on a bridge to capture a view of the scene as less prone to occlusion as possible. The three lasers are used jointly to capture a good view of the scene. The sensors first need to be placed so that they capture a horizontal slice of the environment. Then they are registered into a common reference frame thanks to the use of a calibration box.



(a) (b) (c)

Figure 10: (a) A stand alone laser scanner. (b) A target box is placed at different places in the intersection. (c) The box is used as a calibration target to register all the lasers within a common frame.

Furthermore, to be able to register all the sensors, a wireless network is also built in other to have a temporal synchronization. All the external lasers are synchronized among themselves but also with the host vehicle system. As for the stereo system, because the two cameras are independent, an external trigger has been used so that they both capture the same image at an exact same time.

2.3 Onboard sensors calibration

The testing vehicle that we consider in our work has two main sensors. The first is a laser range finder in front of the car which will can be considered as roughly horizontal w.r.t to the ground. The second sensor is a stereo rig composed of two individual monocular cameras. The front laser is approximately at 70 cm front the ground while the stereo rig is at about 170 cm high with a baseline separating the two cameras of about 70 cm.

In other to use the two cameras as a stereo system, we need to process to its calibration to get the relative position of one camera w.r.t the other. Furthermore, if we want to integrate the laser range finder and the stereo system into a common framework, calibration between those two sensors is also necessary. In the following, we will first present each individual camera intrinsic calibration, then the stereo system calibration and finally the cameras and laser calibration.

2.3.1 Camera intrinsic calibration

Pinhole camera model The simplest model to be used to represent a camera is the pinhole model. The projection onto the image plane of a point Q is at the intersection of the image plane and the ray passing through Q and O the center of projection. On Fig. 11 the point $Q = (X, Y, Z)$ is projected at the point $q = (x, y, f)$ which lies on the image plane, f being the focal length of the camera. The points Q and q are related by (1).

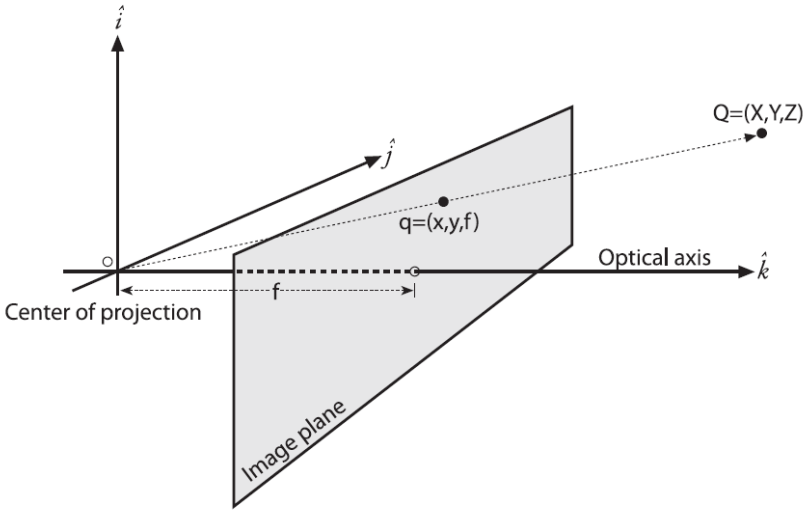


Figure 11: A point $Q = (X, Y, Z)$ is projected onto the image plane by the ray passing through the center of projection, and the resulting point on the image is $q = (x, y, f)$. The image is taken from [4].

$$x = f_x\left(\frac{X}{Z}\right) + c_x, \quad y = f_y\left(\frac{Y}{Z}\right) + c_y \tag{1}$$

By considering homogeneous coordinates, we can rewrite (1) under a matrix product form (2).

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{2}$$

Lens distortions The pinhole model described above is correct considering that the lens doesn't introduce any distortions. In practice, however, it is never the case as the lens are not perfect. To model the distortions induced by the lens, we will consider two kind of distortions, radial distortions and tangential distortions. The radial distortions tend to bent the rays that are far from the center of the lens. Fig. 12 illustrates the effect of radial distortions. This distortion is modeled by three coefficients k_1, k_2 and k_3, which will correct the distortion by using (3).

$$\begin{cases} x_{corrected} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \\ y_{corrected} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \end{cases} \text{, where } r^2 = x^2 + y^2 \tag{3}$$

The other type of distortion is the tangential distortion, it is used to model distortion resulting from lens not being exactly parallel to the image plane. This new distortion will be modeled by two new coefficient p_1 and p_2, which will correct the distortion by using (4).

$$\begin{cases} x_{corrected} = x + \left[2p_1y + p_2\left(r^2 + 2x^2\right)\right] \\ y_{corrected} = y + \left[p_1\left(r^2 + 2y^2\right) + 2p_2x\right] \end{cases} \tag{4}$$

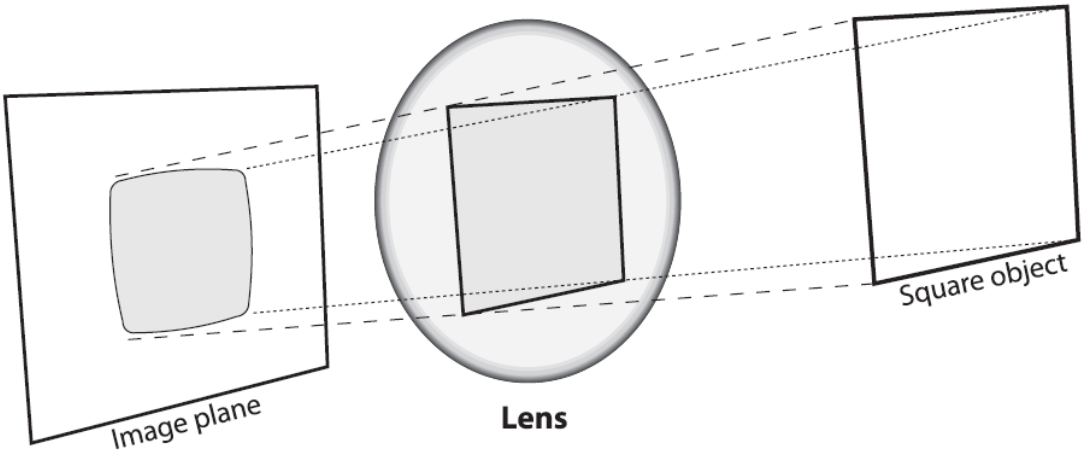


Figure 12: The square edges are bended in the image plane because of radial lens distortions. The image is taken from [4].

Finally, a skew coefficient α_c defines the angle between the x and y pixel axes. The undistorted relation between the point Q and its undistorted projection $q_d = (x_d, y_d)$ is then given by (5)

$$\begin{pmatrix} x_d \\ y_d \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & \alpha_c f_x & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{corrected} \\ y_{corrected} \\ 1 \end{pmatrix} \tag{5}$$

with

$$\begin{cases} x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + \left[2p_1 y + p_2 \left(r^2 + 2x^2\right)\right] \\ y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + \left[p_1 \left(r^2 + 2y^2\right) + 2p_2 x\right] \end{cases}$$

Chessboards Now that the camera is properly modeled, we need to estimate all the camera's intrinsic parameters. To do so we will use a chessboard to get constrained system over those parameters. As illustrated in Fig. 13 we can define a coordinate system attached to our chessboard. Because the size of the chessboard is know, we know the coordinates of all corners of the grid in the chessboard coordinate system. First, some extrinsic parameters need to be introduced. Those parameters will related the camera coordinate system to the chessboard coordinate system. The relative position of the chessboard coordinate system to the camera's one can be described as the combination of a 3D rotation R and a 3D translation T. A point P in the grid reference frame will have coordinates $P_g = (X_g, Y_g, Z_g)$ and coordinates $P_c = (X_c, Y_c, Z_c)$ in the camera reference frame following the rigid motion equation (6).

$$P_c = R * P_g + T \tag{6}$$

The relative position of the chessboard to the camera being unknown, we get additional unknowns. The rotation can be parameterized by three angles, being the rotation around each of the coordinate system axes. In the same way the translation if also parameterized

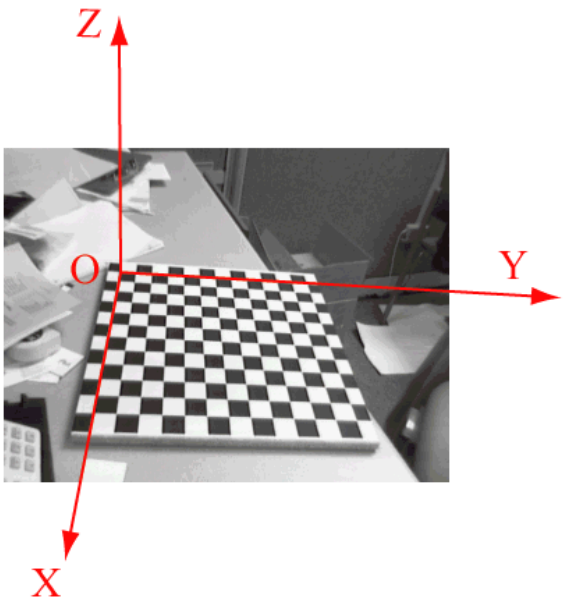


Figure 13: A chessboard and its attached coordinate system.

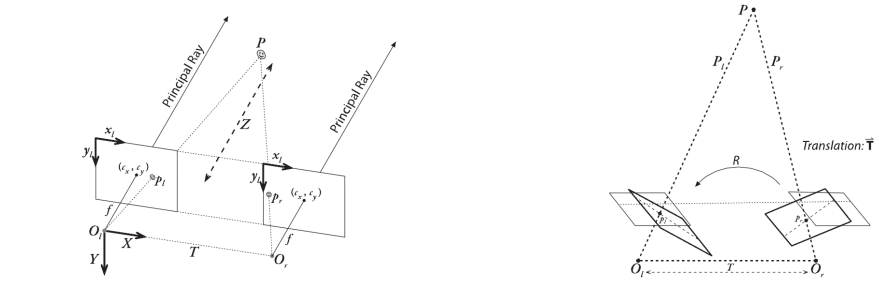
by three values describing the translation along each of the axes. Those six new unknown parameters will be know as extrinsic parameters which we will have to estimate too.

Now the calibration step is to relate the coordinates of the image plane to the grid's one. In the image plane, we first extract the corners of the grid manually or automatically by using a corner detector. For all the extract corners, we will know their coordinates in the image plane as well as their coordinates in the grid frame. The coordinates in the image plane is linked to the camera frame through the intrinsic parameters (5) which in turn is linked to the grid frame through the extrinsic parameters (6). By having enough corners and grid pose, we will have an overconstrained system to solve. For a given estimate of all the intrinsic and extrinsic parameters, we can project back the grid corners onto the image and measure the error between the actual extracted corners and the theoretical ones using the intrinsic and extrinsic parameters. Using error as the value to minimize we can use methods like gradient descent to solve the minimization which will give the final estimate of the intrinsic and extrinsic parameters.

The actual camera calibration has been done using the Matlab camera calibration toolbox [3]. A 1×1 meter chessboard with 10×10 cm cells has been used for calibration while assuming that both the coefficient α_c and k_3 were null.

2.3.2 Stereo rig calibration

The two cameras can be used jointly to compute depth by using a simple triangulation. As illustrated on Fig. 14 (a), if the two cameras are well aligned, the point $P = (X, Y, Z)$ will be projected at $p_l = (x_l, y_l)$ and $p_r = (x_r, y_r)$ respectively on the left and right image plane. Because the x_l and x_r axis are aligned and are parallel to the X axis, we'll actually



(a) (b)

Figure 14: (a) Ideal stereo cameras system. (b) Actual uncalibrated stereo system. The image is taken from [4].

have $y_l = y_r$ and the 3D depth Z of the point P can be compute using (7).

$$Z = \frac{fT}{x_l - x_r}, \tag{7}$$

where T is baseline distance separating the two optical center.

In practice however, the cameras are not well calibrated are the image plane are never perfectly aligned, as shown on Fig. 14 (b). The goal of the stereo calibration is then to virtually align the two image planes.

To do stereo calibration we will use the same chessboard as the one for intrinsic calibration. The relative position between the cameras, which can be represented by a rotation and a translation, can be easily retrieved thanks to the intrinsic calibration. As described in Sec. 2.3.1 we can get the transformation from the camera frame to the chessboard frame. If we capture the same chessboard from the two cameras, a point P will be related to its projections in both image plane by $P_l = R_l P + T_l$ and $P_r = R_r P + T_r$. The transformation R and T that link the right image frame to the left one, $P_l = R^T(P_r - T)$, can be derived using (8).

$$\begin{aligned} R &= R_r (R_l)^T \\ T &= T_r - R T_l \end{aligned} \tag{8}$$

Due to noise, the computed transformation will be slightly different for each grid pose, thus just as in the intrinsic calibration part, a none linear minimization over the reprojection error will be conducted to have the optimal R and T transformation. In the OpenCV [4] stereo calibration implementation, used in our work, the Levenberg-Marquardt algorithm is used to find the minimum of the reprojection error.

Camera alignment To align the two camera on a common plane, we need to decide the location of this plane. One simple solution would be to project either the right camera onto the left camera image plane or the inverse. However, a better solution is to cut the rotation transformation into two rotation and rotating both camera onto the middle common frame. In this way, both camera will be transformed but with a smaller transformation compared to transferring one camera into the other one's frame. The virtual translation between the two frame can also be chosen to maximize the common field of view between the two point of views.

Rectification map The alignment part described just above is only a geometrical constraint over the extrinsic parameters of the system. Even if the camera are aligned, the pixels on the images may not be correctly aligned because of different intrinsic parameters. If the images are perfectly aligned one horizontal line of the left image should be projected on the same line on the right image. The calibration step should be done such as to satisfy this constrain. This can be done by estimating the fundamental matrix F and a rectification map which will given the relationship between the pixels of the two images, detail of the actually computation can be found in [4]. The rectification map will give the proper projection from one camera frame onto the common image plane of the stereo system. Because the image space is discretized into integer coordinate pixels, the new projected image will be filled using bilinear interpolation. An overview of the stereo system calibration and rectification is shown on Fig. 15.

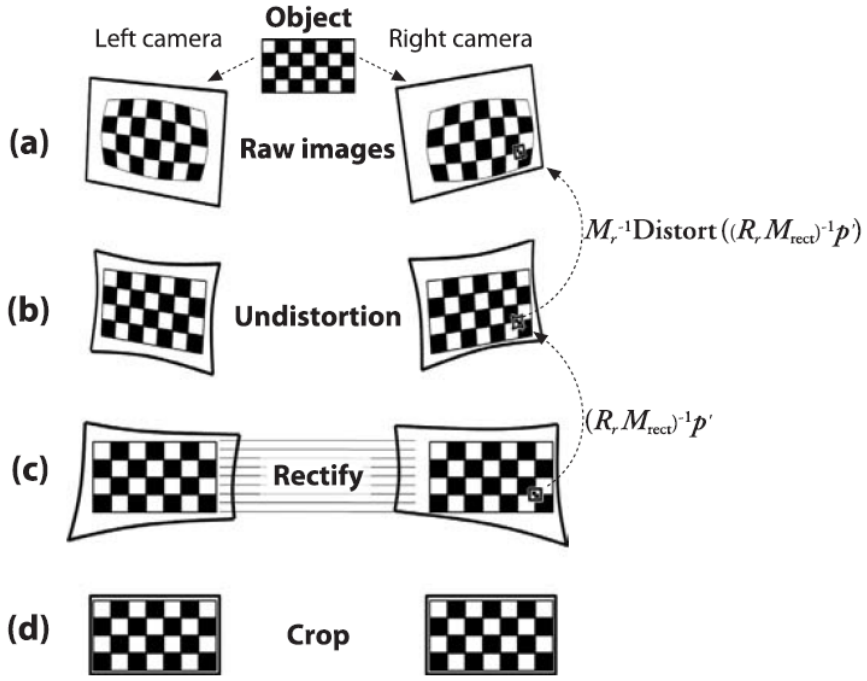


Figure 15: (a) Raw images. (b) Undistortion through intrinsic calibration. (c) Stereo rectification. (d) Crop to keep only the common field of view.

2.3.3 Stereo and laser calibration

The other sensor used in our work is a laser scanner placed in front of the car. In other to use camera and laser jointly, we need to calibrate those two sensors so that we are able to reason in a common reference frame. At Peking university, a single layer laser is used as range sensor, while a four layer laser scanner is used at UTC. Rodríguez et al. [24] used circle shaped target in order to register the laser to the cameras. Because the stereo system used at Heudiasyc lab is rigidly built system, its calibration can be done once for all apart from laser calibration. In the experiment design of Peking university, the stereo rig was built

by our own using two independent monocular camera. Furthermore, the onboard sensors have to be mounted before each acquisition which will provide different configuration each time. Thus, we need a method a do both stereo as well camera/laser calibration before each experiment.

Our calibration framework is mainly based on the works from [25, 33]. The original method is designed to calibrate a monocular system to a laser range finder. However it can also be used for stereo calibration as they make use of a calibration chessboard just like the one needed to do stereo calibration.

Just like stereo calibration we want to know what is the transformation from the laser frame to the camera frame or the inverse. That is given a point $P_L = (X_L, Y_L, Z_L)$ in the laser coordinate system and its corresponding point $P_C = (X_C, Y_C, Z_C)$ in the camera frame, we want to find the rotation R and the translation T satisfying (9).

$$P_L = RP_C + T \tag{9}$$

Contrary to camera calibration, we do not directly have access to pairs of points in both coordinate systems. So other constrains will be needed.

First, the equation of the chessboard is estimated in the camera frame thanks to intrinsic calibration. In our case, we will actually proceed to the stereo calibration as described in Sec. 2.3.2. Using a stereo system will actually give us a better estimation of the grid plane. The grid plane can be uniquely parameterized by its normal vector N and its distance to the camera optical center d. That way, every point in the camera frame lying on the plane will verify (10).

$$N \cdot P_C - d = 0 \tag{10}$$

Thus by using (9) we can derive (11).

$$N \cdot \left(R^{-1}\left(P_L - T\right)\right) - d = 0; \tag{11}$$

That means every laser points hitting the calibration board will have to satisfy (11).

Now, to find the optimal rotation R and translation T we will first need to consider an error measurement. In this calibration process we define the error e as (12).

$$e = \sum_{i}^{N} \sum_{j} D_{ij}^2(R, T) \tag{12}$$

where D_{ij} is the distance of the j^{th} laser point hitting the board of the i^{th} pose. The distance can be the normal Euclidean distance like in [33] or the orthogonal distance advised in [25]. To minimize e, the Levenberg-Marquadt optimization algorithm will be used just like for stereo calibration.

Additionally, because of laser sensing noise, some filtering are performed by considering several laser scans over time. We can use a simple mean or median over the laser range data as a more robust measurement.

Zhang and Pless [33] have also considered this method to improve the camera intrinsic calibration. They do it by minimizing jointly the reprojection error on the image and the

laser error. A further improvement can also be done by considering our stereo system by a global minimization. One additional constrain actually appears as the three transformation linking the two camera together and each camera to the laser are actually related. However one issue could be to find the optimal weights to assign to each error we want to minimize and this could be the object of a short term future work.

3 Stereo based system

Camera is often considered as one of the most important sensor as it is somehow an extension of human eyes. Cameras provide dense information about the environment however monocular system cannot perceive depth. 3D modeling of the scene using monocular system still remains possible through methods like structure from motion. However they tend to be inaccurate and will sometime fail in certain motion pattern, typically when there is no lateral motion. On the other hand laser range finder can provide accurate depth measurement but the drawback being sparse information which are not suited for dense 3D scene modeling. An alternative to those two sensors is the use of a stereo camera. Using two or more cameras can provide images of the environment as well as depth information.

3.1 Stereo system and depth computation

The computation of the depth in a stereo system is based on simple 3D triangulation. The system is supposed to be well calibrated as described in Sec.2.3 and can be represented as Fig. 14(a). A slightly different notation will be used from now on. The (x, y) axis of the image plane will be denoted as (u, v) axis, and the baseline distance referred as T in Sec. 2.3.2 will be denoted b. Let's consider a point P in the world coordinate system which is projected at the pixel positions (u_l, v_l) and (u_r, v_r) in respectively the left and right image coordinate system. We define the disparity as $d = u_r - u_l$. Following this definition, the point P's depth in the camera coordinate system can be recovered using (13).

$$z_p = \frac{fb}{d} \tag{13}$$

Thus to get the depth of a pixel (u_l, v_l) of the left image, we need find its correspondence (u_r, v_r) in the right image. If the images are well rectified as described in Sec.2.3 , we should have $v_l = v_r$. Thus we need only to search the correspondence on one line the image.

For that purpose, we need to first define a similarity measure between image pixels or actually between image patches. Different kind of similarity metrics exist in the literature. The two most commonly used ones are the Sum of Absolute Distances (SAD) and the Sum of Squared Distances (SSD). Given two intensity images I_1, I_2 and a pixel window W of size $m \times n$, the SAD and SSD similarities over W are respectively defined as (14) and (15).

$$SAD(I_1(W), I_2(W)) = \frac{1}{m \times n} \sum_{i,j=1}^{m,n} |I_1(i,j) - I_2(i,j)| \tag{14}$$

$$SSD(I_1(W), I_2(W)) = \frac{1}{m \times n} \sum_{i,j=1}^{m,n} (I_1(i,j) - I_2(i,j))^2 \tag{15}$$

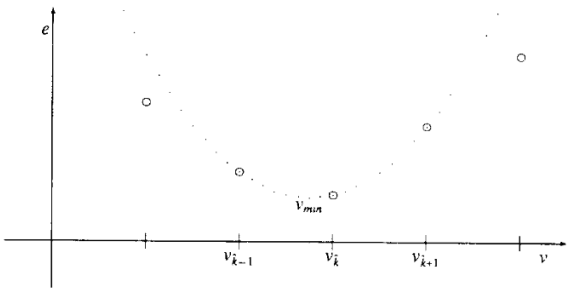


Figure 16: Subpixel precision by fitting a second degree polynomial through adjacent points.

Here $I(W)$ represents the subimage of I covered by the window W. In our case, the SSD similarity will be used. Now, given a left pixel (u_l, v_l), the corresponding right pixel (u_r, v_r) will be defined as (16).

$$\begin{aligned} v_r &= v_l \\ u_r &= \arg\min_u SSD(I_l(W(u_l, v_l)), I_r(W(u, v_r))) \end{aligned} \tag{16}$$

where $W(u, v)$ is a window centered at the pixel (u, v).

Additionally, to get sub-pixel precision the two adjacent pixels will also be used for fitting a second degree polynomial. Suppose that for a pixel (u_l, v_l), it's corresponding pixel in the right image is at (u_r, v_r) w.r.t the SSD similarity. Then as shown on Fig. 16, we can fit a second degree polynomial passing through $s_{-1} = SSD(I_l(W(u_l, v_l)), I_r(W(u_r - 1, v_r)))$, $s_0 = SSD(I_l(W(u_l, v_l)), I_r(W(u_r, v_r)))$ and $s_{+1} = SSD(I_l(W(u_l, v_l)), I_r(W(u_r + 1, v_r)))$.

Then the resulting disparity will actually be the point where the curve reaches its minimum. That is if the curve equation is $y = ax^2 + bx + c$, then the minimum is reach at $x = -b/2a$ which actually correspond to the point u_r derived in (17).

$$u_r = \frac{1}{2}\frac{s_{-1} - s_{+1}}{s_{-1} - 2s_0 + s_{+1}} \tag{17}$$

Furthermore, we can use the curve shape to have an estimation of the variance of the association. Following [20], we can use (18) as an variance estimation,

$$\sigma = \frac{2}{a} = \frac{1}{s_{-1} - 2s_0 + s_{+1}}. \tag{18}$$

The overall disparity computation is then to find the correspondence of each left pixel in the right image. A threshold can be set such that only small enough SSD values can be considered as potential correct association. Finally, a consistency check can be performed by searching the correspondences from the right image to the left and filtering the points that have conflicting associations. Going through the whole image to search for association is very computationaly demanding. However the search for each pixel is independent of each other, thus it is well suited for parallelization. This aspect has been used through a GPU implementation of the stereo matching process. The implementation was based on the one from [7] while adding a variance estimation.

3.2 Egomotion and Kalman filter over the disparity

One important information that can be used for many applications is the host vehicle egomotion. Knowing how the vehicle has moved at each time stamp is often necessary when we want to integrate measurements over time. The use of GPS is one way to get the vehicle position at each frame. However it is sometime not precise enough and may not be always available. Another simple method is the use of internal sensors like odometry sensors which can provide information about wheels' rotation. This information can be used for short term estimation but integrating it over time often leads to inconsistent localization. Moreover, this kind of information are sometime not available for all vehicle like the one used at Peking university.

One other kind of method for motion estimation is the use of sensors like laser or camera. From laser scanner, it is possible to get simultaneously a localization and a mapping through a SLAM framework. Knowing the localization provides directly the vehicle motion. However, doing SLAM in outdoor environment is not easy due to the presence of a lot of moving obstacles. And because laser provides only 2D information, the overall 3D motion of the vehicle cannot be estimated completely. Finally, the use of camera is one the solution to get the vehicle motion properly.

The vehicle motion will be estimated using our stereo system and conducted as following. First a sparse disparity estimation is done over a limited set of points. We want to have accurate 3D estimations of points location, so we will only consider proper points. To do so, we will first extract some SURF points in both the left and right images. Then a correspondance search will be done over those two set of points to have a proper association. A left/right consistency check is performed just like normal stereo computation to delete uncorrect association. Then those points will be tracked over the next time stamp by using the well known Lucas and Kanade tracker. At the same time we will extract new SURF points on the new images then we will proceed to another data association between the tracked SURF points and the new extracted points. Only tracked points with a correspondance will be kept. Finally the point on the new left and right images are associated the same way the old images, the pairs of points that do not correspond to the same association as at the previous time are discarded.

From now on, we will have at our disposal a set of points at the current time with their 3D positions known and also the same set of points at the next time stamp. Because the data association is known we can recover the vehicle motion by computing the rotation and translation that will map the two set of points best. But before doing that, one important point that need to be considered is that this method will given a correct estimation only if the extract points are actually static. To handle the case of points extracted on moving objects, a robust estimator like RANSAC is used to get rid of those outliers. In this case, if at least one half of the extracted points are on static structures, the correct motion can be derived.

As said above, the vehicle egomotion will help integrating measurements over time. Knowing the motion of the vehicle enables to have measurements at different time in the same reference. An application of this egomotion is to improve the disparity estimation. The method is based on the approach of [20] which was originaly designed for depth estimation from monocular camera. The idea is to use a Kalman filter to track each pixel independently. For each pixel the only variable we want to estimate is its depth. Thus we will proceed as follow.

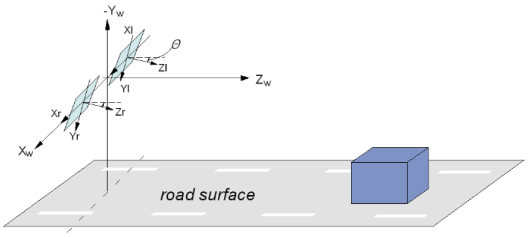


Figure 17: Camera-Ground system

A initial disparity is computed along with the variance at each pixel. Then all the pixels are predicted for the next time stamp thanks to the egomotion. At the next time, a new disparity measurement is also computed. Now new pixels that don't have any previous correspondance are added as new. If a correspondance exists but seems incorrect meaning that the depth difference is higher than three times the current variance, the new point replaces the previous one. And finaly, if a good correspondance exists then an Extended Kalman Filter (EKF) is used to update the new depth estimation as well as its new variance.

3.3 Ground plane estimation

Extracting the ground plane is very usefull in many application. It can be used to constrain detected objects for example, the obstacle that are not lying on the ground can be considered as false positive. There exist different approaches to estimate the equation of the ground surface. For simplicity, we usually assume that the ground is planar or a succession of planar surfaces. Thanks to the stereo system, it is possible to have the 3D position of the points in the scene. A direct approach is to directly fit a hyperplane using the 3D points cloud. Methods like RANSAC or Least Median Square can be used to get a more robust result regarding noise. Another approach is to transform the point representation into a space in which the extraction of the ground plane would be easier. This can be done using for example the v-disparity map introduced in [17]. The idea is to represent all the points by their v coordinate, which is the vertical pixel position, and the value of their disparity. A 3D point (u, v, d) will be projected on the v-disparity space by rejecting its u coordinate.

Let's suppose that we are in the configuration represented in Fig. 17. In the camera coordinate, a point (X_W, Y_W, Z_W) will be projected as (19).

$$\begin{aligned} u_{l,r} &= u_{l,r} - u_0 = f\frac{X_{l,r}}{Z_{l,r}} \\ v &= v - v_0 = f\frac{Y_{l,r}}{Z_{l,r}} \end{aligned} \tag{19}$$

Furthermore we have the following relationship:

$$X_{l,r} = X_W \pm b/2 \tag{20}$$

$$Y_{l,r} = Y_W \cos\theta - Z_W \sin\theta \tag{21}$$

$$Z_{l,r} = Y_W \sin\theta + Z_W \cos\theta \tag{22}$$

Thus we have:

$$u_{l,r} = f\frac{X_W \pm b/2}{Y_W \sin\theta + Z_W \cos\theta} \tag{23}$$

$$v = f\frac{Y_W \cos\theta - Z_W \sin\theta}{Y_W \sin\theta + Z_W \cos\theta} \tag{24}$$

and the disparity is given by:

$$d = u_l - u_r \tag{25}$$

$$= f\frac{b}{Y_W \sin\theta + Z_W \cos\theta} \tag{26}$$

Now if we suppose that the ground is planar with equation $Y_W = h$, the we will have:

$$v = f\frac{h\cos\theta - Z_W \sin\theta}{h\sin\theta + Z_W \cos\theta} \tag{27}$$

$$\text{and } d = f\frac{b}{h\sin\theta + Z_W \cos\theta} \tag{28}$$

$$= \frac{bv}{h\cos\theta - Z_W \sin\theta}, \text{ where } Z_W = \frac{1}{\cos\theta}\left(\frac{fb}{d} - h\sin\theta\right)$$

$$d = bv\left[h\cos\theta - \frac{\sin\theta}{\cos\theta}\left(\frac{fb}{d} - \sin\theta\right)\right]^{-1} \tag{29}$$

From (29) we then have:

$$h\cos\theta d + h\frac{\sin^2\theta}{\cos\theta} - \frac{\sin\theta}{\cos\theta} fb = bv \tag{30}$$

$$\left(h\cos^2\theta + h\sin^2\theta\right) d - \sin\theta fb = bv\cos\theta \tag{31}$$

$$hd - \sin\theta fb = bv\cos\theta \tag{32}$$

This last equation (32) shows that v and d are related by a linear equation. Thus in the v-disparity space, the plane $Y_W = h$ is represented by the (33).

$$d = \cos\theta\frac{b}{h}v + f\sin\theta\frac{b}{h} \tag{33}$$

If we manage to extract this line in the v-disparity space by getting an equation $d = \alpha v + \beta$, then we could recover the camera pose by solving the system (34).

$$\begin{cases} \alpha = \cos\theta\frac{b}{h} \\ \beta = \sin\theta f\frac{b}{h} \end{cases} \Leftrightarrow \begin{cases} h = \cos\theta\frac{b}{\alpha} \\ h = \sin\theta f\frac{b}{\beta} \end{cases} \tag{34}$$

$$\Leftrightarrow \begin{cases} h = \cos\theta\frac{b}{\alpha} \\ 1 = \tan\theta f\frac{\alpha}{\beta} \end{cases} \tag{35}$$

$$\Leftrightarrow \begin{cases} h = \cos\theta\frac{b}{\alpha} \\ \theta = \arctan\frac{\beta}{f\alpha} \end{cases} \tag{36}$$

As for the line extraction part, several methods also exist. One simple way to do is to represent the v-disparity as an image. Every points are accumulated in a v-disparity image, the

intensity of a pixel (v, d) then represents the number of points having that particular (v, d) value. Then using this image, we can proceed to a line extract step by using for example the Hough transform.

The Hough transform can be directly used on the v-disparity by binarizing it. To do so, we define a threshold above which a pixel will have value one and zero below. This simple approach is however quite sensitive to noise. Thus a different Hough transform weighting has been used. First the v-disparity map is normalized so that its maximum has value 1. The for each pixel the weight of its contribution to the Hough transform will be set as its actual value. In this way, a more robust line extraction is achieved.

3.4 Stereo based occupancy grids

One usual way of representing the laser scanner measurement is the use of an occupancy grid map. The space is divided into a grid in which each cell is either free or occupied. This kind of grid can also be built by using the information from the stereo cameras. To do so, we need to project the 3D points cloud onto a 2D plane. This 2D plane will in our case correspond to the estimated ground plane as computed in Sec. 3.3.

A first step it to chose how to discretize the grid plane into cells. The most common way is to represent the world by a Cartesian grid and divide it into regular cells. In this way the world is linearly mapped to the grid coordinates. This representation while being quite intuitive it not well suited for stereo system as it is quite computationally demanding. This is because the transformation from the image plane as well as the disparity onto the Cartesian grid is not linear.

Badino et al. [1] introduced two other grid representations, the column/disparity map and the polar occupancy grid. In both of those maps, each column of the grid will represent one column of the image. The only difference is that the first one will use the disparity value to discritize the space while the second one will be using the depth. The polar grid can thus be seen as a column/depth map.

The authors advise the use of the polar grid representation arguing that the depth discretization remains regular in this way. This would be actually a good choice if the depth estimation have the same degree of accuracy regardless of the depth value. However, the further an object is the less accurate its depth estimation is. Thus the use of column/disparity grid seems actually more consistent with a stereo system.

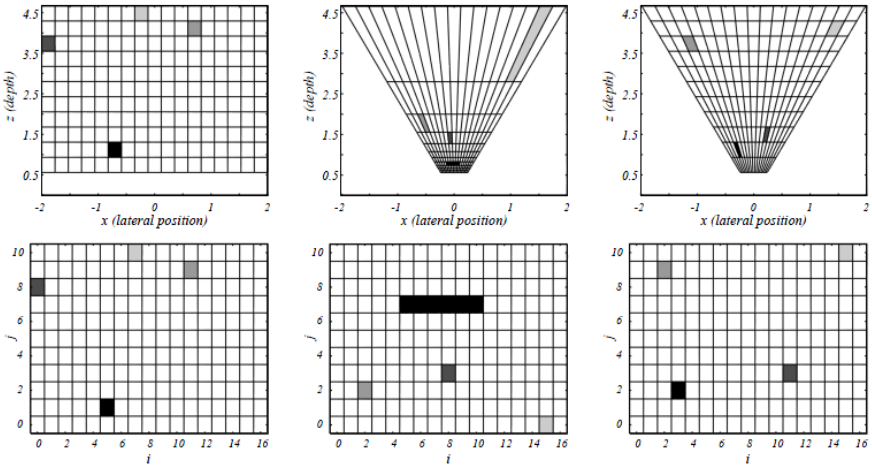
Fig. 18 shows the different mapping between the image coordinate onto the different grid representations.

Each grid cell will be represented by a likelihood value which will denote how likely the cell is occupied. Each pixel in the disparity map will contribute the grid under the form of 2-dimensional Gaussian function G.

$$G(x, y) = \frac{1}{2\pi|\sigma_x\sigma_y|} \exp\left(-\frac{x^2}{\sigma_x^2} - \frac{y^2}{\sigma_y^2}\right) \tag{37}$$

Thus a cell (i, j) of coordinate (u_{ij}, d_{ij}) will receive a likelihood weight $L_{ij}(u, d)$ from the pixel (u, d) corresponding to (38).

$$L_{ij} = G\left(u_{ij} - u, d_{ij} - d\right) \tag{38}$$



(a) Cartesian (b) Column/Disparity (c) Polar

Figure 18: Different kind of grid representations.

The final value of L_{ij} will be the sum of the contribution of all pixel in the disparity map.

$$L_{ij} = \sum_{u,d} L_{ij}(u,d) \tag{39}$$

Ideally each pixel should contribute to the all the cells of the grid but this is very computationally demanding and the contribution of a pixel is often neglectable when the cell is too far from the pixel's projection. Thus we will only update cells which have a Mahalanobis distance less than 3.

$$D_{Mahalanobis}\left(\left(u_{ij}, d_{ij}\right), (u,d)\right) = \sqrt{\begin{pmatrix} u_{ij}-u \\ d_{ij}-d \end{pmatrix}^T \begin{pmatrix} \sigma_u & 0 \\ 0 & \sigma_d \end{pmatrix} \begin{pmatrix} u_{ij}-u \\ d_{ij}-d \end{pmatrix}} \tag{40}$$

One drawback of this 2D occupancy grid representation is the loss of the height information as everything are projected onto the estimated ground plane. One solution is to consider 3D occupancy grid by also considering a space discretization w.r.t the height. However this approach will required a high amount of memory and will lead to a high computation cost. Moreover, the choice of the level of discretization is not straightforward.

Another way to proceed is to introduce a dimensionally limited feature to encode the height distribution of each cell. The features we have consider is based on the whisker box representation used in statistics. To describe a distribution given numerous samples, we use 5 quantities corresponding the first and ninth decile D_1 and D_9, the first and third quartile Q_1 and Q_3 and the median M.

But as a pixel will contribute to different cells with different weight, the likelihood being considered as a weight, we need to change the definition of those quantities. Suppose that we have a set S of $(height, weight)$ couples, and suppose $S = \{(h_i, w_i)\}_{i=1...n}$ sorted by height value. Let's define the total weight $W = \sum_{i=1}^{n} w_i$. The five values defining the height

descriptor will then be defined by:

$$D_1 = h_{d_1}, \text{where } d_1 = \arg\min_k \left\{ \sum_{i=1}^{k} w_i \leq \frac{W}{10} \right\} \tag{41}$$

$$Q_1 = h_{q_1}, \text{where } q_1 = \arg\min_k \left\{ \sum_{i=1}^{k} w_i \leq \frac{W}{4} \right\} \tag{42}$$

$$M = h_m, \text{where } m = \arg\min_k \left\{ \sum_{i=1}^{k} w_i \leq \frac{W}{2} \right\} \tag{43}$$

$$Q_3 = h_{q_3}, \text{where } q_3 = \arg\min_k \left\{ \sum_{i=1}^{k} w_i \leq \frac{3W}{4} \right\} \tag{44}$$

$$D_9 = h_{d_9}, \text{where } d_9 = \arg\min_k \left\{ \sum_{i=1}^{k} w_i \leq \frac{9W}{10} \right\} \tag{45}$$

In this way, this descriptor will become less influenced by low weight potential noise. This descriptor using up to decile quantity actually requires that there are at least 10 couples of height/weight. Discarding the cells not satisfying this condition has actually a limited effect on the final grid as those cells are very likely to be free from the start.

3.5 Free space computation and object segmentation

The next step is now the use of this occupancy grid itself. Mapping is often the goal of occupancy grid but in our case we will further focus on a local map aspect in order to extract the free navigable space in front of the vehicle as well as obstacle detection and segmentation.

To compute the free space in front of the vehicle, a simple and naive way is to set a threshold over the occupancy likelihood and then search for the first cell in each column to have a high enough likelihood.

A better way is to find the optimal separation defining the free space. This can be done using a dynamic programming approach just like in [1]. The dynamic programming problem is defined as follow.

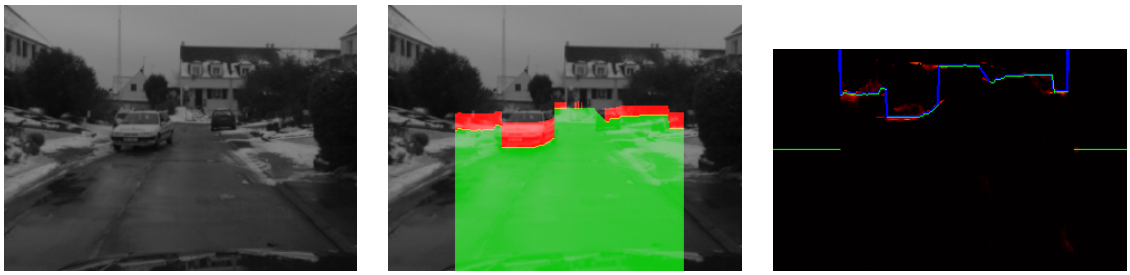
Each cell has a cost defined as $1/L_{ij}$ or a very high value if $L_{ij} = 0$ and the transitions connect every cells to the ones of the next column. The transition cost from the cell (i, j) to the cell (k, l) is originally defined in [1] as

$$E_s(i, j, k, l) = S(j, l) + T(i, j), \tag{46}$$

$$S(i, l) = \begin{cases} C_s d(j, l); \text{if } d(j, l) < T_s \\ C_s T_s; \text{if } d(j, l) \geq T_s \end{cases} \tag{47}$$

$$T(i, j) = \begin{cases} C_t d(j, j'); \text{if } d(j, j') < T_t \\ C_t T_t; \text{if } d(j, j') \geq T_t \end{cases} \tag{48}$$

The two costs S and T smooth the path by penalizing jumps in the spatial space as well as in the temporal aspect. T_s and T_t are threshold that limits the jumps but doesn't penalize true jumps separating two actual different objects. $d(i, j)$ is simply the distance in terms of cell separating the cell i from the cell j. Finally, C_s and C_t are weight put on those two different costs.



(a) (b) (c)

Figure 19: (a) Raw image. (b) Free space in green and obstacle detected in red. (c) Free space separation on the occupancy grid map.

In our case, we'll introduce an additional cost corresponding to height distribution resemblance.

$$H(\Delta_i, \Delta_j) = ||\Delta_i - \Delta_j|| \tag{49}$$

where $\Delta = (D_1, Q_1, M, Q_3, D_9)$. This cost will force the separation path to stick to the obstacles boundary.

Once those costs are defined, the optimal path minimizing the overall costs is computed by using the Viterbi algorithm.

Fig. 19 shows some examples of free space path computation.

Apart from computing the free space, one can also be interested in segmenting the obstacles. In our case, a simple approach is use to do so, while computing the free space separation, we will cluster all cells having similar height distribution and space position.

Conclusion and Future Work

During this internship, we have managed to design an experimental platform with well calibrated sensors. However the acquired data have yet to be processed and ground truth need to be generated for further use. A experimental methodology was built which will ease future data acquisitions.

The two categories of objects detection as described in Sec. 1 have both been test with different kind of approach. This first kind of method was used during our previous work [32] by considering a pedestrian detector with HOG and laser ROI generation. The work done during this internship was one belonging to the second category. We managed to use a stereo camera to extract the free space in front of the vehicle meaning in the same time that we detected the obstacles in front of us.

Several improvements can be considered in both the data acquisition step and the object detection part. As said in Sec. 2, more precise calibration between the stereo system and the laser frame can be achieved by considering a global optimization problem to estimate jointly all the relatives positions of the sensors. Concerning the object detection part, a more complex height representation should be considered to handle special case object not touching the ground like signal panels. Moreover, a coding of the height should also be able to handle the case of partial occlusion.

Future work should also consider fusing both BU and TD approach to have discriminative detector as well as salient object detection. This will be carried out in the future within a PhD thesis at the same laboratory.

Acknowledgments

I would like to thank my supervisor Franck Davoine, as well as Prof. Huijing Zhao, for welcoming for the second time in their team. I also thank the colleagues from Heudiasyc, especially Vincent Frémont for his advices and for providing me testing data. I finally thank all the students and personal from Peking University with whom I was to work with during those few months.

References

[1] Hernn Badino, Uwe Franke, Rudolf Mester, and Frankfurt Am Main. Free space computation using stochastic occupancy grids and dynamic programming. In Dynamic Vision Workshop for ICCV, 2007.

[2] Mayank Bansal, Sang-Hack Jung, Bogdan Matei, Jayan Eledath, and Harpreet S. Sawhney. A real-time pedestrian detection system based on structure and appearance classification. In IEEE International Conference on Robotics and Automation, Anchorage, Alaska, USA, pages 903–909, 2010.

[3] J. Y Bouguet. Camera calibration toolbox for matlab. 2003.

[4] Gary Bradski and Adrian Kaehler. Learning OpenCV. O'Reilly Media Inc., 2008.

[5] N. Dalal. Finding people in images and videos. PhD thesis, Institut National Polytechnique de Grenoble, july 2006.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, volume 1, pages 886 –893 vol. 1, June 2005.

[7] Jan-Michael Frahm David Gallup and Joe Stam. Cuda stereo. http://www.cs.unc.edu/ gallup/cuda-stereo/.

[8] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In Computer Vision and Pattern Recognition., pages 304 –311, June 2009.

[9] Jefferies E. and Yeap W. Robotics and cognitive approaches to spatial mapping. In Springer Tracts in Advanced Robotics, Volume 38. Springer Verlag, 2008.

[10] M. Enzweiler and D.M. Gavrila. Monocular pedestrian detection: Survey and experiments. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 31(12):2179 –2195, dec. 2009.

[11] Andreas Ess, Konrad Schindler, Bastian Leibe, and Luc Van Gool. Object detection and tracking for autonomous navigation in dynamic environments. Int. J. Rob. Res., 29:1707–1725, December 2010.

[12] S. A. Rodríguez F., V. Frémont, P. Bonnifait, and V. Cherfaoui. Visual confirmation of mobile objects tracked by a multi-layer lidar. In Intelligent Transportation Systems. ITSC., pages 849–854, 2010.

[13] Z. Hu and K. Uchimura. U-v-disparity: an efficient algorithm for stereovision based scene analysis. In Intelligent Vehicles Symposium,. Proceedings. IEEE, pages 48 – 54, june 2005.

[14] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 20(11):1254 –1259, nov 1998.

[15] Véronique Cherfaoui Julien Moras and Philippe Bonnifait. Credibilist occupancy grids for vehicle perception in dynamic environments. In IEEE International Conference on Robotics and Automation, Shanghai, 2011.

[16] K. Kohara and N. Suganuma. Obstacle detection based on occupancy grid maps from virtual disparity image. In ICCAS-SICE, pages 4617 –4622, aug. 2009.

[17] R. Labayrade, D. Aubert, and J.-P. Tarel. Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In Intelligent Vehicle Symposium, volume 2, pages 646 – 651 vol.2, june 2002.

[18] Tie Liu, Jian Sun, Nan-Ning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. In Computer Vision and Pattern Recognition, pages 1 –8, june 2007.

[19] S. Maji, A.C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In Computer Vision and Pattern Recognition, pages 1 –8, jun. 2008.

[20] Larry Matthies, Takeo Kanade, and Richard Szeliski. Kalman filter-based algorithms for estimating depth from image sequences, 1989.

[21] Thomas Michalke, Jannik Fritsch, and Christian Goerick. Enhancing robustness of a saliency-based attention system for driver assistance. In ICVS. LNCS, pages 43–55. Springer, 2008.

[22] S. Munder and D.M. Gavrila. An experimental study on pedestrian classification. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28(11):1863 –1868, nov. 2006.

[23] Victor Prisacariu and Ian Reid. fasthog - a real-time gpu implementation of hog. Technical Report 2310/09, Department of Engineering Science, Oxford University.

[24] Sergio Alberto Rodriguez Florez, Vincent Fremont, and Philippe Bonnifait. Extrinsic calibration between a multi-layer lidar and a camera. In IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pages 214–219, Korea, Republic Of, October 2008.

[25] Renaud Keriven Romain Dupont and Philippe Fuchs. An improved calibration technique for coupled single-row telemeter and ccd camera. In 3D Digital Imaging and Modeling (3DIM), 2005.

[26] E. Seemann, B. Leibe, K. Mikolajczyk, and B. Schiele. An evaluation of local shape-based features for pedestrian detection. In Proc. BMVC, 2005.

[27] Xuan Song, Jinshi Cui, Hongbin Zha, and Huijing Zhao. Vision-based multiple interacting targets tracking via on-line supervised learning. In Proceedings of the 10th European Conference on Computer Vision: Part III, pages 642–655, Berlin, Heidelberg, 2008. Springer-Verlag.

[28] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. In CVPR, pages 1030–1037, 2010.

[29] Chieh-Chih Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In IEEE International Conference on Robotics and Automation. ICRA, pages 2918–1924, 2002.

[30] Xiaoyu Wang, Tony Xu Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In IEEE International Conference on Computer Vision, Kyoto, 2009.

[31] C. Wojek, S. Walk, and B. Schiele. Multi-cue onboard pedestrian detection. In CVPR, pages 794–801, 2009.

[32] Philippe Xu, Franck Davoine, and Huijing Zhao. Multi-modal pedestrian detection. In Master Internship Report, 2010.

[33] Qilong Zhang and Robert Pless. Extrinsic calibration of a camera and laser range finder. In IEEE International Conference on Intelligent Robots and Systems (IROS), 2004.