No conventional imagery for secure urban mobility – ICUB: Camera-Lidar calibration and fusion / Deep fusion for Disparity maps enhancement

Kevin SECRET-MORLAND

Supervised by:
Marc Blanchon, Fabrice Meriaudeau, Olivier Morel

Laboratoire ImVia VIBOT ERL-CNRS 6000
Université de Bourgogne Franche-Comté, LE CREUSOT

A Thesis Submitted for the Degree of
MSc in Computer Vision (VIBOT)

· 2020 ·

Contents

List of Figures

List of Tables

Special Thanks

I would like to thank first of all the Imvia laboratory at the IUT du Creusot, within the University of Burgundy Franche-Comté, as well as my supervisors Marc Blanchon, Fabrice Meriaudeau and Olivier Morel who accepted and received me to do this internship. I would also like thanking the company Nvidia who provided the Imvia laboratory with the necessary equipment and power to carry out all our experiments correctly and reliably.

To my colleagues with whom I have worked and shared our experiences during these two years of Master's studies.

To Yohan Fougerolles who offered me a second chance in my studies by accepting me in the DUT GEII and without whom I would absolutely not be here presently.

Finally, to my parents who, without them, all this would be unthinkable, thanks to their support and the mentality they transmitted to me: never to give up despite the torments we encounter in our life. Be stupid.

Chapter 1

Introduction

This manuscript consists in two separate projects: the calibration of a camera and the merging of its images with a Lidar and the improvement of depth maps by fusing them with corresponding images. These two projects, very different at first view, are not so different because they can be totally complementary. Indeed, these two projects will be used on autonomous vehicles for an accurater estimation of distances on the road.

Camera calibration with the Lidar will be performed using a U-Eye camera and a Lidar Robosense RS-Lidar-16. The system implemented requires a Lidar Venolyne: the Robosense cloud point algorithm is not compatible and needs to be adapted upstream. The Calibration of extrinsic parameters consists in the detection of a Polygonal pattern by the point clouds of the Lidar and the pixel frame of the camera. By this fact, we can obtain calibration coordinates to fuse each pixel in each point of the Lidar cloud result.
Finally, after the calibration, the camera pixels will be merged with the Lidar point cloud to work the pixels in a 3D environment. This technique is notably employed in the field of culture, for the preservation of heritage, by reconstructing historical monuments in 3D by photogrammetry. Indeed, photogrammetry allows to obtain precise images and Lidar the shapes of the monument to obtain all the details necessary for the study and the safeguard of the heritage.

The improvement of depth maps is made through Deep Learning for a fusion between RGB images and their depth map equivalence. This learning obtains the final goal of improving distance recognition on unconventional image captures (rain, snow, overcast, etc.).

Contribution of this manuscript:

- To give a resume of two different approaches about the comprehension of the environment by a robot: Camera-Lidar, Depth camera
- To offer a new experimental protocol for different system
- To explore the different solutions to optimize correctly the loss functions by using different type of Networks

Chapter 2

Background

2.1 Introduction

Since the beginning of the using of computers, we always have needed to perform the precision of the sensors we use. Many scientists deals with this problem to replace the mechanical sensors by mixing the vision context with the computers. This innovative method drastically enhances the comprehension of the robot in its environment and evolve the software engineering topic in a new way of robotics: The artificial intelligence. These new vision methods can detect and extracts features, modify the format of images, analyse and detect similarities on the extracted measures and simulate the comportment of the world with different parameters.
But we know the vision will never replace but constantly need to works with important mechanical stuffs like Lidar to fill the gaps or adding more details that the resolution or the constitution of the cameras cannot offers. Accordingly we can fuse the 3D points of the Lidar with the 2D resolution and the color pixels of the camera whose goal is for a robot to be able to understand and interact with its environment.

2.2 Camera calibration

2.2.1 Conventional camera calibration

The camera calibration can be used for monocular of stereo pin-hole cameras to estimate and configure the sensor parameters including the extrinsic, intrinsic and distortion parameters with the goal of removing the lens distortion, estimate depth using stereo cameras, measuring planar objects or estimate the 3D structures from camera motion for 3D reconstruction.

The convenient way, such as Bouguet tool box proposed by Jean Yves Bouguet [4] to calibrate the camera with multiple images from a calibration pattern like a checkerboard to obtain the 3D world point and their corresponding 2D image point in Equation 3.1 and 3.2

$$w[XY1] = [XYZ1]P \tag{2.1}$$

$$P = \begin{bmatrix} R & t \end{bmatrix} K \tag{2.2}$$

With P is the 4 by 3 camera matrix, $[XY1]$ if the image points and the scale factor w, $[XYZ1]$ the 3D world points, R & t the extrinsic rotation and translation matrices, K the intrinsic matrix.

Steve McGuire and al. 2018. [32] propose a method to recover the extrinsic parameters of the camera in outside environment by the calculus of the likely hood between the main camera of the robot and another one positioned on the kinematic arm through rotation estimation. The system combines known arm kinematics with observations of conics in the image plane to calculate maximum-likelihood estimates to model the image noise as "Zero-mean Gaussian result" in the maximum likely hood estimator which can be stated as a nonlinear least squares optimization problem in 2.3 of the form:

$$r = e(z, x)R^{-1}e(z, x)^T \tag{2.3}$$

Where $e(z, x)$ is the residual error vector, dependent of the observation z and the model parameter x and R & T are the Rotation and Translation matrices.

D. Jarron and al. 2019. [19] Propose a method to calibrate an array of multiple panoramic cameras with a Lidar to collects georeferenced spatial data with integrated navigation and imaging sensors from a moving vehicle with a mobile mapping system (MMS) which is a three-dimensional reality capture setup. The panoramic camera (Ladybug5) has to be perfectly calibrated to ensure accurate fusion of images to point clouds created by the Lidar system to reconstruct 3D environment. The calibration of Ladybug by Ikeda et al. 2003. [18] consists in the using of 2D calibration pattern to generate spherical panorama imagery by projecting onto a curved surface at a significant distance from the camera's centre of gravity (single projection point) Figure 2.1. This technique generates a panorama with an angular error of 0.3°regarding to older approach like Schneider and Forstner. 2013. [43] which has 0.6°of angular error.

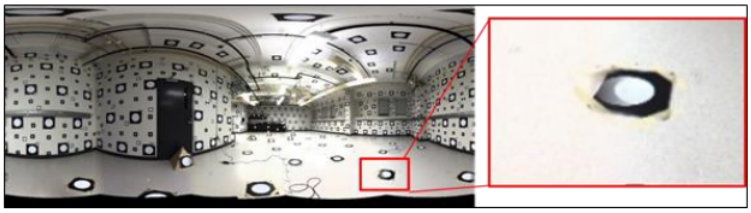


Figure 2.1: Generated panorama and calibration pattern with Ladybug5

2.2.2 Camera auto-calibration

Other calibration methods consists in the auto-calibration to determine automatically the parameters directly from multiple uncalibrated images of unstructured scenes. It does not require any calibration pattern or any particular objects but it's a part of "Match Moving" process where a synthetic camera trajectory and intrinsic projection model are solved to re-project synthetic content into video.
Romil BHARDWAJ and al. 2018. [3] propose and automatic and low cost camera calibration at scale for traffic sensors to estimate real-world traffic distances with the aim of the speed evaluation of vehicles, generate automated traffic reports.
The auto calibration is based on multi-camera fusion which consists in the construction of 3D images from different video sensors viewing a scene to the same frame of reference. These techniques use 4 corresponding points from real-world coordinates. With the camera intrinsic parameters, we can solve the calibration using the equation 2.4

$$sp_c = C[R|T]_{p_w} \tag{2.4}$$

Otto Korkalo and al. 2019. [23] developed an auto-calibration with depth camera networks for people tracking. It is based on Shape matching-based detection simplified by the 3D information from deepness images to evaluate the fitness of the model and gradient based-methods to depend exclusively on the geometry of the scene. The advantage of this technique is the depth images which don't depend on the illumination of the scene and thus, not sensitive to changes of lightning conditions and can be operated in dark environments. Depth is a powerful cue for foreground segmentation, 3D shape and metric observations which simplifies foreground object classification. Occlusions can be detected and handled more explicitly, and the third dimension can be used for the prediction step in tracking. The system is kindly the same than general, based on feature matching and detection. The sensors are align on top-view coordinate using pair-wise affine transformation to create a map and reconstruct the scene. Figure 2.2

The calibration pattern are not only used with solid materials but also implemented virtually.

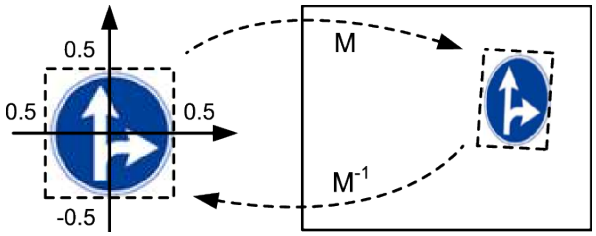


Figure 2.2: Affine transformation for feature detection

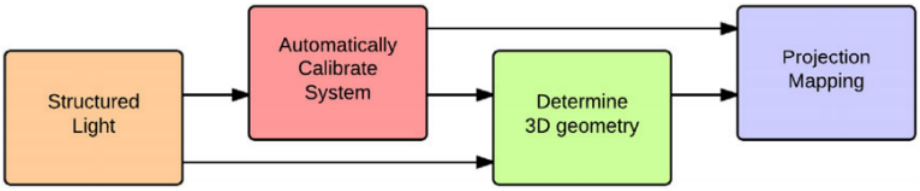


Figure 2.3: Auto-calibration combines the camera calibration and acquisition of three-dimensional (3D) geometry into a single pipeline. Because no additional calibration targets are needed for calibration, the process is much faster compared with the traditional methods.

Jason Deglint and al. 2016. [7] developed this type of auto-calibration with a new type of virtual pattern diffused with a projector. The assumption estimates the manual calibration with 2D checkerboard pattern is not reliable for 3D surface estimation for cost, error-prone and time consuming. This auto-calibration project a series of gray code structured light patterns to calibrate stereo systems Figure 2.4. The algorithm has to minimize the cost at each iteration, respect the camera parameters and the noisy image frames. It has to determine a 3D geometry and create a projection matrix to reconstruct 3D objects. Figure 2.3

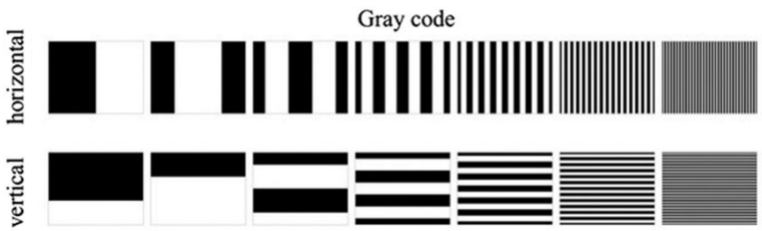


Figure 2.4: Gray code used like virtual pattern calibration.

2.2.3 Deep camera calibration

New methods to calibrate cameras are proposed using Neural Networks to improve the precision and the speed of calibration processes.

Syed Navid Raza and al. 2019. [41] propose a new type of conventional method fused with deep learning method. It is clearly based on the improving of checkerboard calibration patterns in every situation like illuminated, dark, noisy, high lens malformation images, etc. . The data-set is composed of a huge bunch of checkerboard images in various conditions.

Alexander Hanel and Uwe Stilla. 2018. [15] propose a method to calibrate iteratively the camera parameters on cars using scale references extracted from traffic signs as reference patterns. Mechanical and thermal effects might cause the parameters to modify over time, requiring deep iterative calibration to detect clearly and in real time road signs.

Yannick Hold-Geoffroy and al. 2019. [16] propose a full deep network calibration camera without pattern calibration but from a unique image and trained using automatically generated samples from a large-scale panorama data-set in order to reconstruct 3D environment or objects with better precision. The structure chosen is DenseNet Figure 2.5 which performs camera re-localization by jointly learning location and orientation. Deep convolutional neural networks in calibration are operated to estimate field of view and horizon lines, bringing camera calibration on single images to a wider variety of scenes in order to solve the need for high-level reasoning. The data-set for deep learning needs an enormous bunch of images and has to be specified to the work; the work is to train a deep network to estimate the camera roll, pitch, and field of view from a single image. This could be possible by the using of SUN360 database which contains a large number of 360°panoramas (399,728 pairs of photos) to extract 7 rectified images from each panorama using a standard pinhole camera model of random parameters.

DenseNet Figure 2.5 compute with an Adam optimizer is modified by changing the last layer onto three heads:

- One for estimating the horizon angle ψ
- A second one to estimate the horizon's distance to the center of the image ρ
- A third one to estimate the vertical field of view of the image $h\theta$

The output layer of this network is obtained by a Softmax to calculate a probability distribution by discretize their respective parameter into 256 bins.

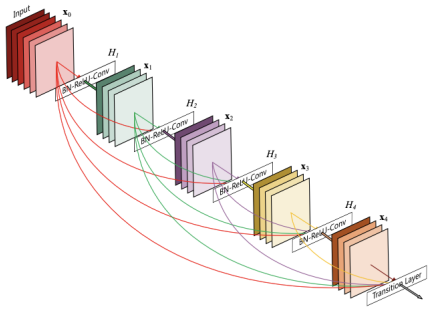


Figure 2.5: Standard DenseNet, here the last layer is modified to have three separate heads

2.3 Camera-Lidar calibration

In this section we will see many ways to calibrate a camera with different type of Lidar. Different to traditional methods, this camera calibration will use the 3D point clouds obtained by the Lidar. It is generally used for 3D mapping robot or 3D reconstruction in architecture (like photogrammetry) or autonomous vehicles. The methods presented are dated from 2010 to 2019.

Naveed Muhammad and Simon Lacroix. 2010 [35] propose a technique to calibrate only the intrinsic parameters of 3D multi-layer Lidar (Velodyne HDL-64E S2, which contains 64 layers). It consists in an optimization process, which gives precise estimation of calibration parameters starting from an initial estimate. The optimization process is based on the comparison of scan data with the ground truth environment for scene modeling, obstacle detection, and SLAM. The intrinsic calibration for such systems is the estimation of parameters that define the position and orientation of each of the laser beams in a sensor-fixed coordinate frame.

The calibration has to be performed in few steps:

- To convert the raw scan data into a 3D point cloud.
- The calibration environment is designed and constructed to acquire Lidar data for calibration.
- Five parameters are required to define one laser beam in a 3D coordinate frame:
  - Two angles to define the direction of the associated line.
  - Three parameters to define the point origin of the beam.

	Default Calibration	Recalibration
4m	0.0312	0.0378
6m	0.0325	0.0301
8m	0.0200	0.0106
10m	0.0192	0.0064
12m	0.0190	0.0090
14m	0.0217	0.0117

Figure 2.6: Biased estimation of calibration parameters

- Choosing a cost function to minimize and optimize the process ti acquire 3D point cloud data in a real environment.

$$C_x = \sum (P_{x,i} - P_{x,mean})^2 / n$$
$$C_y = \sum (P_{y,i} - P_{y,mean})^2 / n$$

This technique represent a suitable base to appreciate the functionally of a 3D Lidar system, here with high resolution. It improves correctly the depth estimation in higher distances higher than 4m and validate the improving of the calibration after 10m, see Table 2.6

Sergio A Rodriguez and al. 2010. [42] & [10] propose a method of extrinsic parameters calibration of cameras with a multi-Layer Lidar with four layers for distance estimation in "intelligent vehicle application" This is a circle-based calibration object because its geometry allows to obtain not only an accurate estimation pose by taking advantage of the 3D multi-layer Lidar perception but also a simultaneous estimation of the pose in the camera frame and the camera intrinsic parameters Figure 2.8. Its simplifies the calibration tasks in outdoor condition. This method determines the relative position of the sensors by estimating sets of corresponded features and by solving the classical absolute orientation problem Figure 2.7. There are no projection pixels from the camera frame in the Lidar point cloud, the technique is only used to calibrate the extrinsic parameters of the camera and the laser data from the Lidar helps the estimation of the distance of the camera.

This type of calibration needs a specific rigid transformation to define the corresponded points of the Lidar ${}^{t}P_{ij}$ by the camera frame ${}^{c}P_{ij}$ in 2.5:

$${}^{c}P_{ij} = {}^{c}R_t \cdot {}^{t}P_{ij} + {}^{c}T_t \tag{2.5}$$

Kiho Kwak, Daniel F. Huber, Hernan Badino, and Takeo Kanade. 2011. [25] provide a method of extrinsic camera calibration by the scanning of a single line from the Lidar. It is base on the minimizing of the distance between corresponding features projected onto a distance known image plane which is a v-shaped calibration target in order to improve the calibration

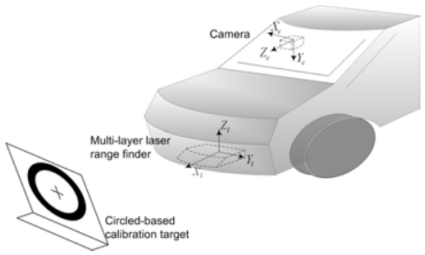


Figure 2.7: Experimental protocol of circle-based camera calibration

Figure 2.8: Results of circle-based calibration process

error. Figure 2.9
The algorithm proposes two different ways:

- The using of different weights to distance between a point and a line feature according to the correspondence accuracy of the features.
- The applying of a penalizing function to exclude the influence of unwanted objects in the calibration data sets.

This technique has a calibration accuracy over 50% which is better than traditional camera calibration methods. The problem of this technique, in comparison to the newer methods with real multiple layers Lidar is the 3D reconstruction that could be less precise and quick because the 2D-3D correspondence mix the camera pixels frame with the single line Lidar which is rotated in 90°Figure 2.10. This could be considerate as one of the pillars of newer camera-Lidar calibration.

Gaurav Pandey and al. 2012 [38] propose a method of extrinsic parameters calibration with 3D multi-layers Lidar (Velodyne-64E) and an omni-directional camera (Ladybug3) mounted on the roof a vehicle by maximizing mutual information without the need of specific calibration target. This algorithm uses a mutual information (MI) framework based on the registration of the intensity and reflectivity information between the camera Figure 2.12 and laser Figure 2.11 modalities
The method has to assume:

- The intrinsic calibration parameters of both the camera system and laser scanner are known
- the laser scanner reports meaningful surface reflectivity values.

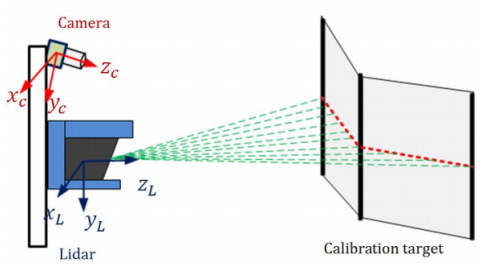


Figure 2.9: Schematic of the experimental protocol

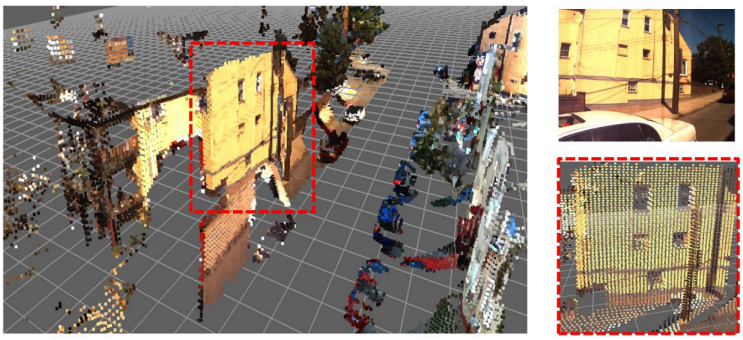


Figure 2.10: 3D reconstruction of scene after calibration

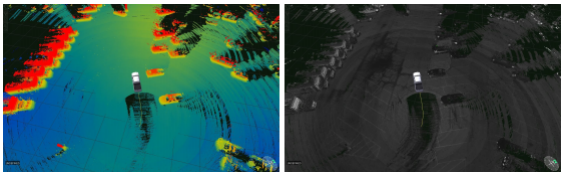


Figure 2.11: Point cloud acquisition from 3D multiple layers Lidar

The correlation coefficient for the reflectivity and intensity values for the scan-image pair at different values of the calibration parameter is calculated, and the distinct maxima at the true value are observed.
To be used surely in robotic situations, in order to use them in any vision or simultaneous localization and mapping (SLAM) algorithm, the Cramer-Rao-Lower-Bound (CRLB) of the variance of the estimated parameters is calculated as a measure of the uncertainty.

Faraz M Mirzaei, Dimitrios G Kottas and Stergios I Roumeliotis. 2012 [33] focused their interest to the problem on the calibration of the intrinsic parameters of a 3D Lidar in the same time of the extrinsic parameters of a camera. It improves the technique based on iterative minimization of nonlinear cost functions to solve this problem. The algorithm has to divide the

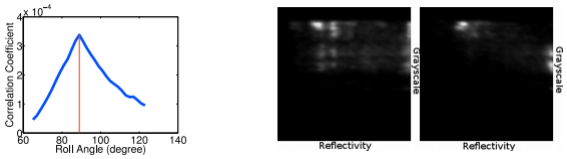


Figure 2.12: Reflectivity results from camera frame

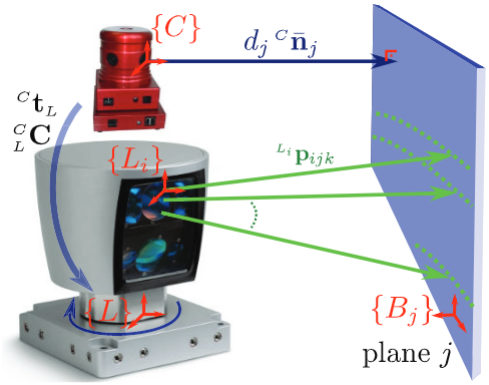


Figure 2.13: Geometric constraint between the jth plane, the camera C, and the ith laser scanner, Li. Each laser beam is described by a vector ${}^{Li}P_{ijk}$. The plane is described by its normal vector ${}^C\bar{n}_j$ and its distanced j both expressed with respect to the camera.

problem into two least-squares sub problems, analytically solve each one to determine a precise initial estimate for the unknown parameters and finally increase the accuracy of these initial estimates by iteratively minimizing a batch of nonlinear least-squares cost function.
All this problem is solved by the using of least-squares method to perform the batch iterative joint optimization of the LIDAR–camera transformation and the LIDAR's intrinsic parameters.

- First to estimate the 3 degrees of freedom rotation between each conic laser scanner and the camera in which the polynomial equations has to be solved with an algebraic-geometry approach to find all of its critical points.
- Second to compute the initial estimate for the relative translation between the camera and the conic laser scanners, and the remaining intrinsic parameters.

Xiaojin Gong, Ying Lin and Jilin Liu. 2013 [14] developed a method of extrinsic parameter calibration method with a Camera (Ladybug3) and a 3D Lidar (Velodyne HDL-64E) by the using of an Arbitrary Trihedron, solved by Nonlinear least squares method 2.14. The relative transformation between the two sensors is calibrated via a nonlinear least squares problem,

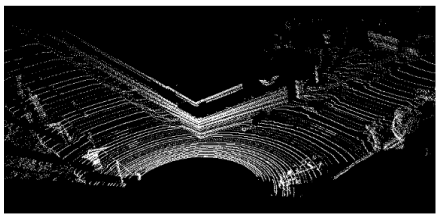


Figure 2.14: Point cloud capture of the calibrated Lidar

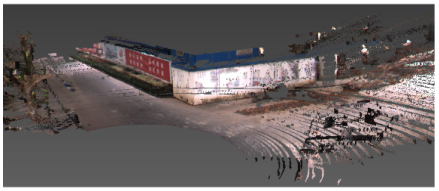


Figure 2.15: Fusion between the Lidar point cloud and the camera frame

which is formulated in terms of the geometric constraints associated with a Trihedral object. The initial estimates of Least squares method are obtained by dividing it into two sub-problems solved individually and the calibration parameters and calculated by optimizing the Least square method. This technic benefit of the same advantage than Yoonsu Park and al. 2014. [39] by using "Polygonal like" patterns as:

- The structure could be orthogonal
- The pattern detection could be used both in indoor and outdoor
- Quite convenient for a robot to detect easily this pattern
- Less perturbed by bad weather conditions.

This technique is a 2D-3D correspondence between the camera frame and the Point cloud of the 3D Lidar Figure 2.17. To compute the transformation, the two sensors have to capture the same Trihedron pattern individually. Then, the extrinsic calibration is formulated as a nonlinear least squares problem in terms of some specific constraints to this pattern: Planarity constraint between two frames, Planarity constraint between two images, the motion constraint.

Ashley Napier and Peter Corke and Paul Newman. 2013 [36] propose a method of cross-calibration of 2D Push-Broom Lidar and extrinsic parameters of a camera on moving plate-form like Google street view system. The algorithm is automatically estimating the relative pose between a push-broom LIDAR and a camera without features extraction/the need of artificial calibration targets or other human intervention; it exploits the motion of the vehicle to retrospectively compare the LIDAR point cloud with camera image frame data to re-calibrate in real time the sensors.
To proceed, it synthesises images from Lidar reflectance acquisition, based on the calibration between the sensors and measures their alignment accuracy to exploit the gross appearance of the scene using a robust gradient based Sum of Squares Objective function optimized in

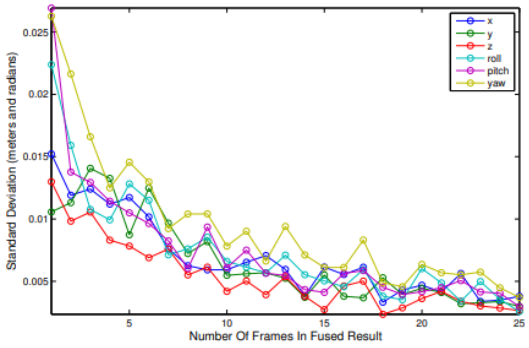


Figure 2.16: The evolution of the standard deviation in function of acquired frames

Standard Deviation	*Translation (mm)*			*Rotation (degrees)*		
	x	y	z	roll	pitch	yaw
Individual Results	28	21	15	1.4	1.4	1.5
Fused Results	4.5	5.2	4.6	0.38	0.39	0.44

Figure 2.17: The standard deviation of the cross_calibration estimates after 73 frames

Equation 2.6. The calibration giving maximal alignment to be accepted as the best estimate for the camera-Lidar calibration.

$$ {}^{c}\bar{T}_l = \underset{{}^{c}T_l}{argmin} \sum_{\mathcal{I}_k^c} \left\| \mathcal{Q}(\mathcal{I}_k^c) - \mathcal{Q}(\mathcal{I}_k^l({}^{c}\mathcal{T}_l)) \right\|_2 \tag{2.6} $$

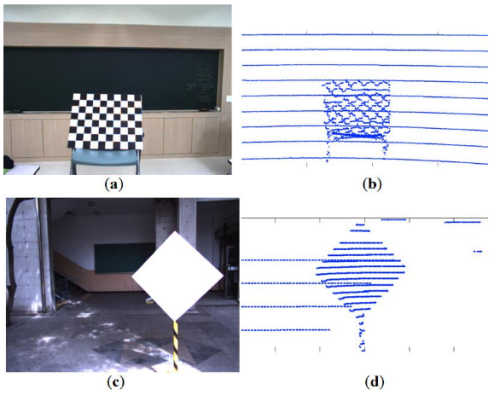
where:

- $\sum_{\mathcal{I}_k^c}$ is the sum of all pixels in the image pair.
- $\mathcal{I}_k^c$ is the camera image frame.
- $\mathcal{I}_k^l$ is the laser reflectance image.
- $\mathcal{I}_k^c$, $\mathcal{I}_k^l$ ${}^{c}\mathcal{T}_l$ & Q$(\cdot)$ is a Gaussian smoothing before taking the magnitude gradient image and performing normalisation patch.

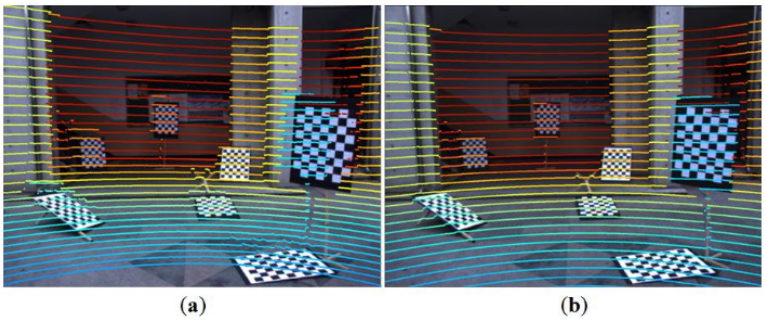
With this approach the algorithm can update the calibration of the camera, depending of the real time calculate estimations. The standard deviation of the calibration estimate reduces as more frames are encountered and the calibration estimate is updated, Figure **??**

Yoonsu Park and al. 2014. [39] propose a method of camera-Lidar calibration with a Polygonal Planar board. The goal is to improve the calibration accuracy between a camera and a 3D

LIDAR especially with a low resolution 3D LIDAR with a relatively small number of vertical sensors because nowadays the Lidar technology is still very expensive especially on high resolution 3D Lidar with multiple layers. The experimental Lidar has a number of 32 layers. In this 2D-3D correspondence method, the 3D points are exploited by the scanning of the Polygonal Planar board with known size adjacent sides Figure 2.18a Since the lengths of adjacent sides are known, the vertices of the board can be estimated as a meeting point of two projected sides of the polygonal board.
The Polygonal board has a better detection accuracy of the vertices, both the Lidar and the camera than the checkerboard pattern Figure 2.18a due to the resolution layers of the Lidar. The 2D information of the camera frame has to be fused with the 3D point clouds from the Lidar. The Figure 2.18b compares the checkerboard and Polygonal methods. The Polygonal method is more stable and accurate than checkerboard only and the 2D-3D correspondences with a Lidar is a good opportunity to calibrate easily a camera for real-time 3D reconstruction.



(a) Point cloud acquisition from the Lidar



(b) Point cloud and image frame fusion

Figure 2.18: Comparison between checkerboard and Polygonal methods with the Lidar.

Castorena, Kamilov and al. 2016. [5] propose a method to automatize the extrinsic calibration of camera parameters by joining the frame data with the point cloud of a 3D Lidar to provide complementary information about the environment, overcome hardware limitations, or reduce data uncertainty due to each individual sensor. The approach exploits the natural alignment of depth and intensity edges when the calibration parameters are correct.
It does not requires the presence or identification of known alignment targets or any specify calibration pattern like checkerboard or Polygonal targets and the joint processing evaluates and optimizes both the quality of edge alignment and the performance of the fusion algorithm uses a common cost function on the output. The low resolution point cloud from the 3D Lidar is fused with the image frame from the optical camera to produce higher-resolution depth images. The algorithm will use this first non-calibrated 2D-3D fusion process to compute the auto-calibration of the camera and the extrinsic parameters are estimated by the minimization of a cost function on θ 2.7 which penalizes miss-alignments between the gradients of the projected high-resolution depth map and the intensity image.

$$\hat{\theta} = \underset{\theta \in \Theta}{argmin} \sum_{k \in \{x,y\}} \frac{\mathcal{A}_k(\Theta_\theta)}{\mathcal{N}_k(\Theta_\theta)} \tag{2.7}$$

Zoltan Pusztai and Levente Hajder. 2017. [40] propose a method of increase the accuracy of camera-Lidar calibration by using ordinary boxes detection with Polygonal method. The goal is simple; the Lidar project and recover the laser reflected by its environment/objects around it to create a point cloud map. On the other side, the RGB camera recovers the image frames. To calibrate the camera with the Lidar, we need to have a calibration pattern. Normally it's a simple checkerboard or detects features to take the 3D information from the environment by affine transformations.
Here the 3D information is taken by the ordinary boxes with known size from the Lidar point cloud: When the boxes are detected by RANSAC method, an algorithm will detect automatically the corners of the boxes Figure 2.19. So the assumption is if we can synchronise the 3D coordinate points of the boxes from the Lidar, with the camera frame, so we have all the parameters to calibrate the camera. More of this, in fine, the camera frame could easily fused with the Lidar point cloud to create a dynamic scene for 3D reconstruction Figure 2.20. This type of calibration could be operated with multiple cameras or Lidar. The procedure needs 4 steps:

- Point clouds information from the Lidar
- At least one image frame
- Intrinsic camera parameters

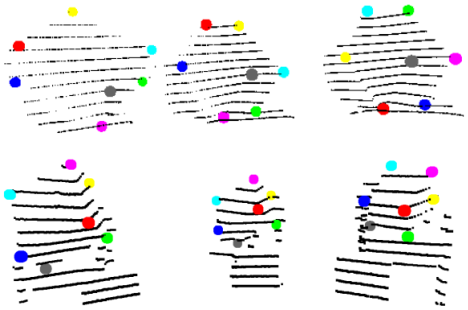


Figure 2.19: Detected boxes by RANSAC, the corners are automatically detected

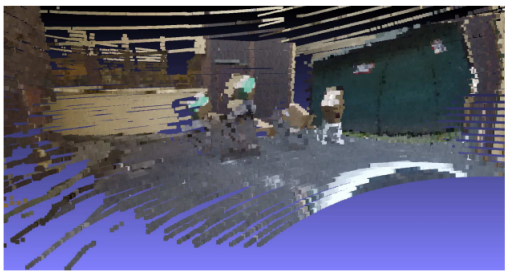


Figure 2.20: The camera frame and the point cloud from the Lidar are fused

- The measured length of the boxes

2.4 Image fusion

The fusion with images or data is very common in computer vision whether for image enhancement, feature detection, extracting high frequency details or calculating specify features in different colour spaces. This state of art will be focused on newer methods which implement convolutional neural network techniques, proved to be more accurate than older traditional methods.
According to the state of art of Shutao Li and al. 2016. [28], Pixel-level image fusion is designed to combine multiple input images into a fused image, which is expected to be more informative for human or machine perception as compared to any of the input images and shown notable achievements in remote sensing, medical imaging, and night vision applications.

Yifan Jiang, Xinyu Gong, Ding Liu and al. 2019 [21] propose a method of unsupervised deep light enhancement (EnlightenGAN) without low/normal-light paired training images.

Kuang, Xiaodong, Sui, Xiubao, Liu, Yuan. 2018 [24] propose a method of single infrared image enhancement using a GAN-based deep convutional neural network. This network is used to produce images with enhanced constrast and details. The GAN is implemented to avoid and prevent the amplification of background noise.

Guorun Yang, Hengshuang Zhao and al. [47] propose a method of disparity estimation for binocular stereo images, based on a new model called SegStereo. This network employs semantic features from segmentation and introduces semantic softmax loss, which helps improve the prediction accuracy of disparity maps. It tested with KITTI Stereo benchmark and produce good results on both CityScapes and FlyingThings3D data-sets.

Yu Liu and al. 2016 [28] propose a method fusion of multi-focus image using convolutional neural network in order to enhance blurred images. The common method to solve this problem is to fuse multiple images from the same scene to reconstruct the broken details. The CNN algorithm is trained by high-quality image patches and their blurred versions to encode the mapping. The method has to be more accurate than traditional methods which don't uses convolutional neural networks. Figure 2.21
The pre-processing if divided into 3 steps before but essential for the Fusion:

- Focus detection: The two source images are fed to the Siamese pre-trained CNN model to output a score map. The score maps contains the focus information
- Initial segmentation: the focus map is segmented into a binary map with a threshold
- Consistency verification: the binary segmented map is refined with two popular consistency verification strategies (the small region removal and the guided image filtering, to generate the final decision map.)

The fusion is processed by using the pixel-wise weighted-average strategy from the final decision map.
The CNN used is very simple and only has 3 convolutional layers set by a 3x3 kernel sized with 256 concatenated and fully connected feature vectors and the training images comes from the ILSVRC 2012 validation image set, which contains 50,000 high-quality natural images deriving from the ImageNet dataset.

Zhaodong Liu and al. 2016 [29] propose a fusion method of multi-focus image based approach on image decomposition (which accurately obtain morphological content) in order to integrate the relevant information from a set of images with the same scene, into a comprehensive image. The fusion process is based on image cartoon-texture decomposition by an improved iterative re-weighted decomposition algorithm which is designed to converges and approximates the morphological structure components. The fusion processing of the cartoon contents and

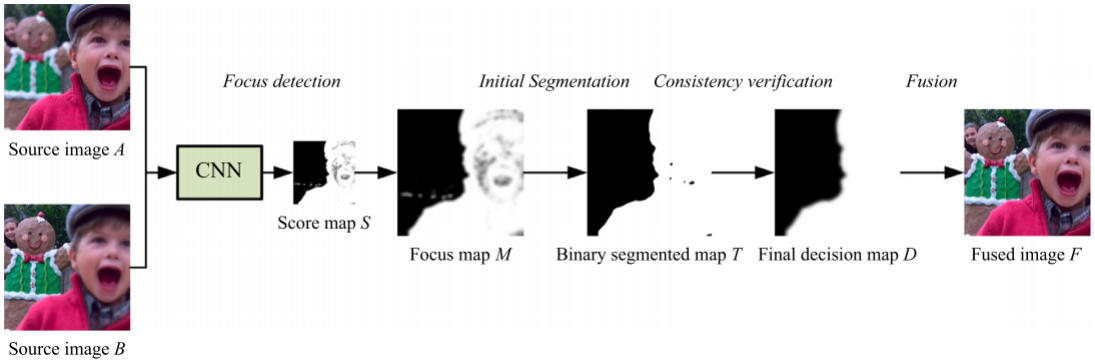


Figure 2.21: Schematic diagram of multi-focus image fusion algorithm with CNN

texture contents is combined to obtain the "all-in-focus-image" which preserve morphological structure information from source images and performs few artifacts or additional noise.

Zhiqin Zhu and al. 2017 [50] propose the same type of fusion technique, based on image decomposition and sparse representation in order to preserve the structure information and perform the detailed information of source images. The decomposition based method is the same than Zhaodong Liu and al. [], to decompose image to cartoon component with a spatial-based method for morphological structure preservation and texture component by a sparse representation composed of a trained dictionary with strong representation ability.
The two transformed component are fuse according to the texture enhancement fusion rule.

Wenda Zhao, Huimin Lu, and Dong Wang. 2017. [49] propose a method of fusion for the enhancement of images in spectral domains, based on spectral total variation (TV) method and image enhancement. It verifies the decomposition components of each subband can be modeled efficiently by the tailed Rayleigh distribution rather than the commonly used Gaussian distribution to obtain a high-contrast and edge-enhanced fused image Figure 2.22.
The fusion computation is divided into four parts:

- Multiscale Decomposition Based on the Total Variation Spectral Method which decompose the source images into multiscale representations using the spectral total variation method.
- Detail Layer Fusion and Enhancement Based on the Tailed Rayleigh distribution. It consists in several process:
  - A match measure between the subbands of the input images to determine which

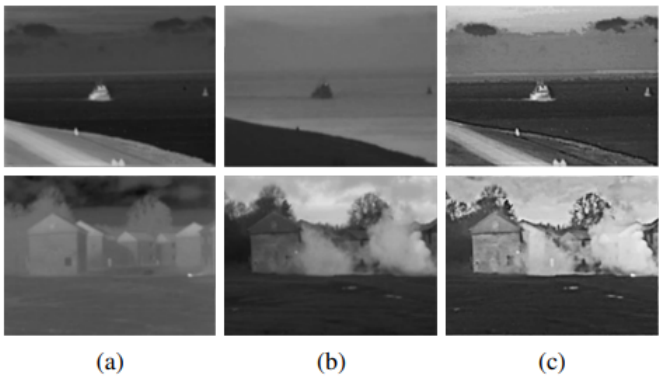


Figure 2.22: Infrared fusion result : (a) Infrared image (b) visible image (c) fusion result

coefficient will be assigned a larger weight

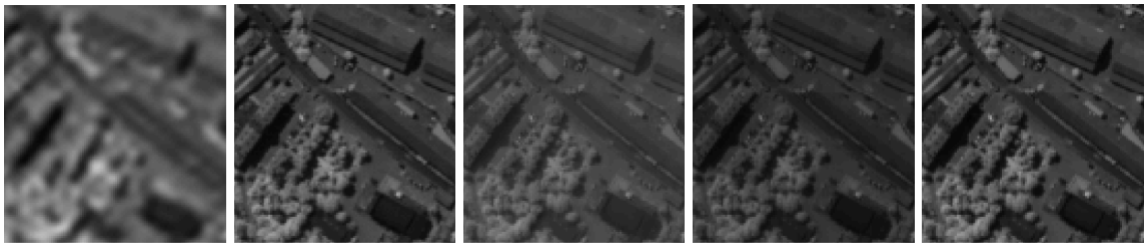
- A salience measure that is calculated to determine which subband coefficient will be copied in the fused subband.

- Basic Layer Fusion Based on Local Energy

- Multiscale Image Reconstruction : By adding adaptive gains to each fused subband decomposition, a high-quality fused image is obtained, although all the input images have low contrast and blurred edge details.

The proposed method effectively preserves the main features of source images while enhancing the edge details and contrast of the fused image but it doesn't work well when the image is disturbed by noise.

Frosti Palsson, Johannes R. Sveinsson and Magnus O.Ulfarsson. 2017. [37] propose a fusion method with Multispectral and Hyperspectral images using 3D Convolutional Neural Network in order to enhance the resolution of low spatial resolution Hyperspectral images. Hyperspectral image which is more robust to noise than other dimensions, is used for the identification of different materials based on their spectral signature, which is useful for applications such as classification of land cover types.
The CNN architecture used for this technique is the 3D-CNN architecture cause it has the specificity to learn spectral-spatial features and HS image has two spatial dimensions and one spectral dimension. It is composed of three zero-padded convolutional layers with a 3x3x3 and 1x1x1 kernel for the first and the last convolutional layer followed by a Gaussian noise regulation to reduce overfitting in the network.



(a) Interpolated HS image (b) Reference (c) MAP1 (d) MAP2 (e) 3D-CNN

Figure 2.23: A subset of band 102 of the HS image is shown. (a) shows the interpolated HS image band, (b) is the reference band, (c) shows the image obtained using the MAP1 method, (d) shows the image obtained using the MAP2 method and (e) shows the image obtained using the proposed method.

In a 3D-CNN, which has 3D filters and 3D receptive fields, the output of the nth feature map y^n at location i, j, k is given by:

$$y^n_{i,j,k} = \sigma(b^n + (\mathbf{H}^n * \mathbf{x})_{i,j,k}) \tag{2.8}$$

With $\mathbf{x}$ is the input data, b^n and H^n are the shared bias and filter (shared weights) and σ is a non-linear activation function.

The final step is the reconstruction of the estimated high resolution HS image (obtained by the pre-trained CNN) Figure 2.23

Junho Jeon and Seungyong Lee. 2018. [20] propose a method of pairwise depth image data-set generation method using dense 3D surface reconstruction with a filtering method to remove low quality pairs and enhance the input depth images. The final goal of this technique is to be used as noise removal for 3D reconstruction.

Yukang Gan, Xiangyu Xu and al. 2018. [12] propose a method of monocular depth estimation with affinity, vertical pooling and label enhancement with Convolutional Neural Network to improve depth accuracy. The depth labels are enhanced by generating high quality dense depth maps with of-the-shelf stereo matching method by taking image pairs as input.

Kaiyue Lu, Shaodi You and Nick Barnes. 2017. [31] propose a method of deep texture and structure aware filtering network for image smoothing. This technique has to preserve essential textures that other CNN smoothing methods doesn't provides. A large data-set is generated by blending natural textures with clean structure-only images and use the result to build a texture

prediction network that predicts the location and magnitude of textures.

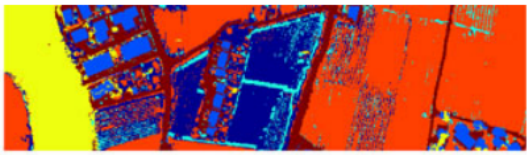
Pedram Ghamisi and al. 2017 [13] propose for the fist time a fusion method for Hyperspectral and Lidar data fusion with Deep convolutional neural network and Extinction Profiles method in order to extract spatial and elevation information from both the sources, including height, area, volume, diagonal of the bounding box, and standard deviation and classify all the extracted features, materials, with a deep convolutional with logistic regression to produce a classification map. The method is tested with two data-sets from Houston (US) and Trento (Italy) in the size of 27x27 pixels.
The fusion is divided into two different strategies:

- Feature stacking which is a simple approach to integrating extracted features from Lidar and Hyperspectral images. The problem of this approach is it increase dimensionality in the feature space and downgrade the classification accuracy of classifiers Figure 2.24a.
- Graph-Based Feature Fusion used for the fusion of spectral, spatial, and elevation features in order to fuse the features described above, the number of dimensions should first be normalized in order to put the same weight on each type of the features and reduce the computational cost and noise throughout the feature space Figure 2.24b.

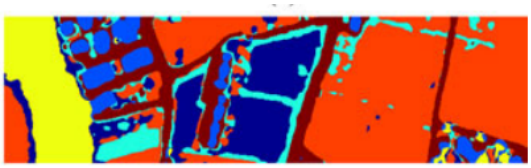
The deep convolutional network used is composed of three convolutional layers with 4x4 and 5x5 kernel sized and 2x2 pooling matrices. This new method takes the advantages of every sensor and algorithm techniques to improve and give nice feature extraction and classification with fusion than traditional and prove the stack technique is less accurate than graph-based feature fusion. It demonstrates that it doesn't depend to only one mapping area and could be used on every image from earth. But the Lidar has its limit and has to be use only on nice Zhenfeng Shao, Member, IEEE, and Jiajun Caiweather conditions because of light sensor system.

Luyang Jing and al. 2017 [22] propose a method to evaluate accurately the damage and problem detection on mechanical planetary gearbox with an adaptive multi-sensor data Fusion method based on Deep Convolutional Neural Network. To deal with two challenges: The feature extraction and the selection of a suitable fusion level for a specific fault diagnosis task without the using of human expertise task. The proposed method can learn features from raw data and optimize a combination of different fusion levels adaptively to satisfy the requirements of any fault diagnosis task.

Huihui Song, Qingshan Liu, Guojie Wang, Renlong Hang, and Bo Huang. 2018. [44] propose a novel approach of satellite images fusion with deep convolutional neural networks for enhance



(a) Stack fusion method



(b) Graph-based feature fusion method

Figure 2.24: Comparison between The stack and graph methods.

the resolution and extract landscape features on dynamic monitoring. This method could solve a huge problem in aerospace imagery: The remote sensing data with both high spatial and temporal resolutions are difficult to capture by current satellite platforms due to the constraints in technology and budget.
Spatiotemporal fusion aims to integrate two types of remote sensing data with similar spectral information, including the number of bands and the bandwidths.

- One type is featured by high spatial resolution but low temporal resolution (HSLT)
- The second by low spatial resolution but high temporal resolution (LSHT)

The fusion process is divided into two steps: Prediction & training. Given one pair or two pairs of HSLT–LSHT images on prior dates and one or more LSHT images on prediction dates, spatio-temporal fusion models integrate these images to produce high spatial resolution images on prediction dates. The training is constituted with two image data-sets: Landsat and MODIS, which are different due to the resolution; Landsat is a Low-spatial resolution (250m) whereas MODIS has 500m resolution.
For both the prediction and training step, the CNN used is the Super-Resolution CNN (SR-CNN) which is specialized in the transform from low-resolution image, an higher one.

Yunlong Yu and Fuxian Liu. 2018 [48] propose a method of detection ans classification of aerial scene features based on Deep convolutional neural network multi-level fusion.

Afonso M. Teodoro and al. 2019 [45] propose a novel fusion method of convergent images for denoising using Scene-Adapted Gaussian-Mixture-Based Denoising. The denoiser has the

objective to enhance the analyse of convergent images which, in conventional techniques, hard to analyse in order to correct the convergence errors by the fusion technique. The proposed method is tested on two different problems: Hyperspectral fusion/sharpening and fusion of blurred-noisy image pairs.

Kin Gwn Lore Adedotun Akintayo Soumik Sarkar. 2016 [30] propose a method to enhance low-high images by a deep autoencoder approach to be used for the monitoring and tactical recognition on dynamic environment. It is based on two techniques : simultaneous and Sequential learning of contrast-enhancement and denoising (LLNet and Staged LLNET). The trained model is evaluated with natural low-high images.

Yu-Sheng Chen Yu-Ching Wang and al. 2018 [6] propose a method of image enhancement for photographs. The algorithm has to enhance the desired images without image pairs: the method, based of Generative Adversarial Networks (GANs) and U-NET, learns a photo enhancer which transforms an input image into an enhanced image with those characteristics.

Xueyang Fu, Jiabin Huang and al. 2017 [11] propose a method of image enhancement based on new deep network architecture called DerainNet. The proposed architecture has to remove the rain on single-image. The model learn the mapping relationship between rainy and clean image detail layers from data. This could be considered has data Fusion of image enhancement. Because it's not possible to obtain rainy and sunny images in the same time naturally to create a database, the rainy images are synthesized automatically. This new technique is important to improve depth computation.

Saeed Anwar, Chongyi Li, Fatih Porikli. 2018 [1] propose a method to enhance underwater image based on convolutional neural network (UWCNN) trained with synthetic underwater image dataset. This technique could improve low contrast and distorted color casts. Their model, in comparison to others, directly reconstructs the clear latent underwater images by leveraging on an automatic end-to-end and data-driven training mechanism.
Chongyi Li, Saeed Anwar, Fatih Porikli. 2019 [27] propose a new deep method of underwater image and video enhancement based on Underwater scene prior (UWCNN). To improve light absorption and scattering which degrade the visibility of images and videos. This affect the accuracy of pattern recognition, visual understanding, and key feature extraction in underwater scenes.

Vaishnavi Hurakadli ; Sujaykumar Kulkarni ; Ujwala Patil and al. 2019 [17] propose a a deep learning based pipeline to estimate the radial blur and enhance the deblurred image to be

used on autonomous vehicle systems. The enhancement module is designed with convolutional autoencoder which enhances the deblurred image to remove artefacts in order to detect the traffic signs.

Konstantinos Batsos ; Philippos Mordohai. 2018 [2] propose a recurrent residual convolutional neural network architecture (RecResNet) for disparity map enhancement generated by a stereo algorithm. The proposed method try to modify disparity values and how to estimate the new disparity map and their corresponding ground truth.

Chapter 3

Camera Lidar calibration

3.1 Introduction

3.1.1 Definition

The using of Camera with a Lidar empirically the correspondence of the 2D pixel information from an RGB camera with the 3D point cloud information from the Lidar (Light Detection and Ranging) to create a 3D depth map and determine the distances in the environment without the using of stereo cameras by taking the advantage of 360°3D viewing from the Lidar point cloud and the 2D RBG pixels from the Camera frame in order to be operate on autonomous vehicles for distance estimation or default detection on precise materials.

3.1.2 RGB camera

The RGB camera is a two dimensional image taken from the three dimensional world by a pin hole camera. The result depends on:

- The illumination.
- The position and the number of light sources.
- The wave length.
- The physical properties, the color and the orientation of the object.
- The resolution of the sensor matrix.
- The projection distortion.

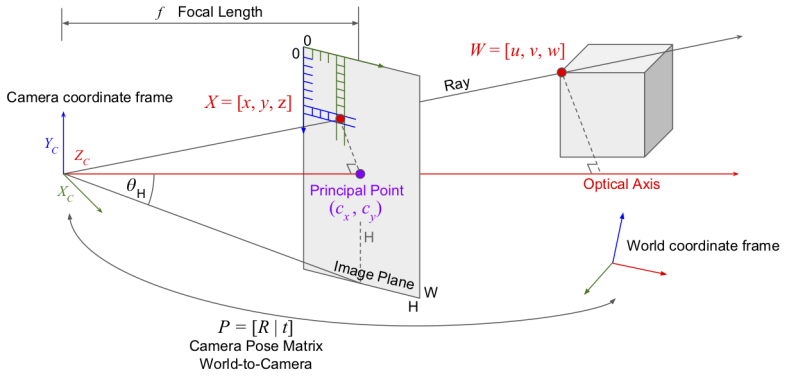


Figure 3.1: Schematic diagram of different camera parameters in pinhole camera calibration

In cameras, to determine precisely the 2D coordinates from the 3D world and determine the location of the camera in the scene or reconstruct in 3D a specific object, it needs a geometric calibration to estimate the extrinsic parameters and calculate the re-projection errors, remove the lens distortion by corresponding the 3D-world points with their 2D image point from the camera frame see Figure 3.1. Commonly these correspondences are possible using calibration patterns like checkerboard like Bouguet toolbox [4] or other exotic patterns like Sergio A. Rodriguez and al. 2010 [10] & [42] by using circle-based calibration pattern for auto-calibration. The intrinsic parameters, different to extrinsic including the focal length, the optical center and the skew coefficient, are innate to the camera and doesn't have to be calibrated.

The calibration of extrinsic parameters takes account of some essential sub parameters for the camera: The rotation (R) , translation (T) and intrinsic matrix (K), the 2D image (XY1) and 3D world points (XYZ1) and the scale factor (w). Equations 3.1 & 3.3

$$w[XY1] = [XYZ1]P \tag{3.1}$$

$$P = \begin{bmatrix} R & | & T \end{bmatrix} K \tag{3.2}$$

$$w \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.3}$$

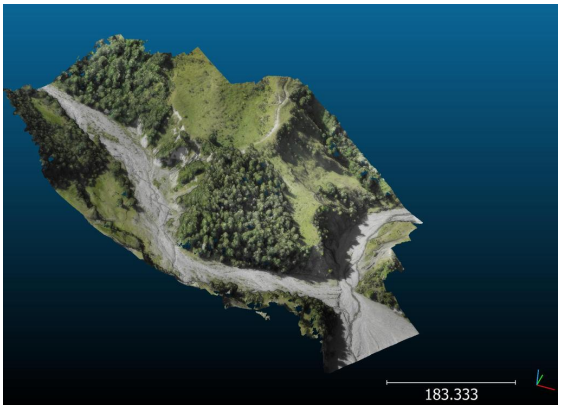


Figure 3.2: Capture of 3D Lidar point cloud and associated with pixels from camera sensor

3.1.3 Lidar

The Lidar is a 3D sensor which measure the depth of an object or an area with a laser (could be with different wavelengths (250 for ultraviolet to 2000nm for infrared, depending on the environment in which the object or the area has to be captured)) by measuring the back-scattered light. This sensor is a point-wise Time-of-flight sensor that measures depth by estimating the time delay from light emission to light detection Equation 3.4.
"Light is emitted from the LiDAR and travels to a target. It will reflect off of its surface and comes back to its source. As the speed of light is a constant value, the LiDAR is able to calculate the distance to the target."

$$Distance = (Speed of Light \times Time of Flight)/2 \tag{3.4}$$

This type of sensor could be occurred to light source position or illumination or environmental noises (Like smoke, rain, snow, etc.) depending on wavelength type, but does not requires any pre-calibration before using. Knowing the position and orientation of the sensor, the XYZ coordinate of the reflective surface can be calculated, represented by a point and the whole obtained points allows to build a map in order to reconstruct an area for example.
There are three types of Lidar:

- 1D composed of one single beam used for scanning simple pattern like bar codes.
- 2D composed of one rotated beam to collect horizontal distance from the targets to get data on X and Y axes.
- 3D composed of multiple rotated beams spread out on the vertical axe with an defined and precise angle δ to get X, Y and Z axes.

The maximum effective distance is 200m but the greater the distance, the more the Lidar is dependent on its point cloud resolution because the distance between each points and the error between them increase with the measured distance.
By calculus, with an angular resolution equal to 0.35°, at 1m, the distance between two points is 0.006m and at 200m the distance between the same points is 1.22m.

In spite of these disadvantages, the Lidar is a powerful sensor in its using domains like 3D reconstruction or distance estimation cause it does not requires huge resource-intensive algorithms and each point from the cloud could be associated with the pixel from a camera sensor to produce a precise coloured 3D environment in real time see Figure 3.2.

3.1.4 Goal

The main goal of this subject is the using of an existing technique for the calibration of the extrinsic parameters of an RGB camera with an high resolution 3D Lidar providing 16 laser beam rings in order to project/fuse the pixels from the camera into the point clouds of the Lidar. This method from Ankit Dhall, Kunal Chelani, Vishnu Radhakrishnan, K.M. Krishna. 2017. [8], propose a package, performed in ROS environment, which use the Polygonal technique like Yoonsu Park and al. 2014. [39] and Xiaojin Gong, Ying Lin and Jilin Liu. 2013 [14] both with monocular and stereo-vision camera, by changing the traditional checkerboard calibration pattern for a simpler one, more accurate for the detection by the Lidar sensor.

The most problem we will meet if technological: Indeed, the proposed method is using a Lidar from the Velodyne brand with a proprietary language. Because this type of sensor is still very expensive, we had to obtain another from a generic brand in which the point cloud matrix is not compatible with the algorithm. This will be solved by an independent algorithm in addition to the main system.
The final experiment would has to be installed on a mobile robot and tested in real time, both indoor and outdoor to give the advantages and disadvantages of this method with this type of installation and environment and the possible improvements in order to be used, for example, with different wavelengths to may solve any outdoor issues.

An extra goal could be to experiment this type of calibration with another technique which uses the traditional checkerboard calibration pattern to compare, in spit of the conclusions from others authors, the pattern detection accuracy between these and try to mix the Polygonal and the checkerboard methods in order to take the advantages and create a fully universal and functional calibration pattern for both camera and Lidar.

3.2 Experimental protocol

In this section we will first present the proposed paper all the detailed method. Secondly, we will do a technical review of all the equipment used to proceed on this way. Thirdly, we will present the experimentation buy explain in detail all the modifications brought to make this equipment usable to this method. Forth, the result from the first steps but not the final step with the Camera/Lidar fusion.

3.2.1 Paper presentation

Camera-Lidar calibration

The used method proposed by Ankit Dhall, Kunal Chelani, Vishnu Radhakrishnan, K. Madhava Krishna [8] is a fully functional ans official technique for ROS environment. It allows to calibrate the extrinsic parameter a monocular (2D-3D correspondence) or stereo (3D-3D correspondence) RGB camera with the point clouds from a Lidar with a Polygonal pattern technique, more accurate for detection by the laser beam sensor, in order to fuse the pixels from calibrate camera and the point cloud from the Lidar to provide complementary modalities and obtain a fully coloured 3D reconstruction of an area or an object by a moving indoor robot or an autonomous vehicle. Here we will focus our self only on 2D-3D correspondence method, that means, with monocular RGB camera.

The calibration pattern used for this experiment is two rectangular cardboard cutouts in the center. The matching of the 2D-3D correspond points between the camera and the Lidar provides 6 Degrees of freedom. Thanks to the planar cardboard, the feature matching algorithm RANSAC needs only 4 corners in each pattern: 4 corners on the outer rectangle and 4 corners on the inner rectangle.
To solve the horizontal nature of the Lidar's scan lines and improve the detection accuracy of horizontal and vertical edges, the patterns has an optimal position angle of 45°. Now we have 4 edges detected by the Lidar and RANSAC fit points position to obtain correct geometrical shapes.

The line segments obtained by the detected points can now being manually marked by drawing polygons with a ROS node around each line segment in order to calculate their intersections in 3D and approximate the corners. To check the approximation of the corners, the shortest line-segment is measured and compared to the real cardboard marker dimensions measuring. The length error between the edge lengths is about 1 centimeter in average. By using a hollowed pattern, we increase the number of detected corners points (8 points) by opposition to solid

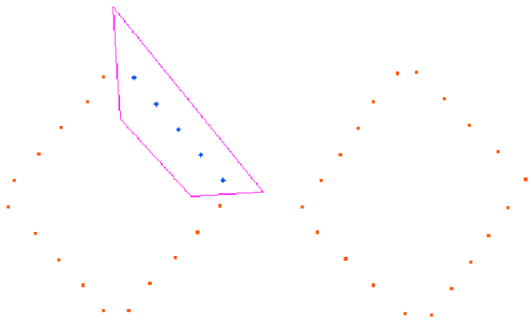


Figure 3.3: Detected points from the two cutoff cardboard and marking of line segments by drawing polygons

pattern (4 points).

Now we need to find the rigid-body transformation between the set of 2D-3D correspondences before 3D projection in Equation 3.3. This is possible by the calculus of the Perspective n-Point (PnP) cost function:

$$\underset{R \in SO(3), t \in \mathbb{R}^3}{argmin} \; ||P(RX + t) - x||^2 \tag{3.5}$$

With P is the projection from 3D to 2D on the iamge plane, X is the 3D points, x is the 2D points, R is the rotation matrix and t is the translation matrix.

Some noisy data points can appear during the point capture. These are contributing to a large back projection error which is a bad fitting of point distribution in the cloud. This could be solved by the addition of RANSAC method on top of the main algorithm. At this way the 2D-3D correspondence points from PnP algorithm are used to determine $[R|t]$ using trigonometry and then calculate 3D projection in 3.3 to calibrate the camera.

Camera-Lidar Fusion

Now, the 2D pixels from the camera is calibrated with the 3D point cloud from the Lidar. The two data can now be fused together. This fusion could be operated with both monocular and stereo camera but in stereo the Lidar can have the only functionality to calibrate the two cameras without fusion cause stereo sensors could operate 3D reconstruction without 3D point clouds. Here is the proposed method to fuse two cameras with the Lidar but the algorithm could be easily used with only one camera or extended with higher number of cameras:

- Let's consider C_1 and C_2 the two following cameras and L the Lidar. We obtain for each camera a $4 \times 4 matrix$:

$$T_{Lidar-to-C_1} T_{Lidar-to-C_2} \tag{3.6}$$

- We chain the transforms $T_{Lidar-to-C_1}$ & $T_{Lidar-to-C_2}$ to find the transform between C_1 & C_2 in order to fuse them:

$$T_{C_2-to-C_1} = T_{Lidar-to-C_1} * T^{-1}_{Lidar-to-C2} = T_{Lidar-to-C_1} * T_{C_2-to-Lidar} \tag{3.7}$$

If the transform is doing correctly, the point cloud from the Lidar will merge properly with the pixels from the camera. However, if not, there will be a translation or rotation error between them and diverge more and more with the distance from the origin.
For stereo cameras, the fusion will be from the two point clouds created with the two cameras. The merging and the transform error are the same with hallucinations or ghosts of objects are visible, increased by distance.

3.2.2 Experimental equipment

In this section we will explain the two main equipments used to compute this algorithm: The Lidar Robosense RS-LiDAR-16 and the RGB IDS camera UEye from IDS Imaging.

- The Lidar Robosense RS-LiDAR-16:
  - 3D high resolution sensor with 16 layers of laser beam.
  - Measurement range: 20cm to 150m
  - Measurement accuracy: 2 centimeters
  - Data rate: up to 320,000 points/second
  - Horizontal Field of View: 360°
  - Vertical Field of View: 30°
  - Angular Resolution (Horizontal/Azimuth): 0.09°at 5Hz to 0.36°at 20Hz
  - Rotation Speed: 300 to 1200rpm
  - Laser wavelength: 905nm

- IDS UEye camera UI-3240CP-C-HQ
  - Sensor Technology: CMOS Color
  - Resolution (h x v): 1280 x 1024
  - Analog/digital converter: 10 bit
  - Pixel size: 5.3 μm
  - Interface: USB 3

Figure 3.4: Pattern feature detection with U-Eye camera and Aruco algorithm

3.2.3 Experimentation

In this part we will explain the whole experiment, until the acquisition of the corner points from the cardboard pattern. We first speak about the Robosense/Velodyne compatibility issue and solving, Secondly the conception of the pattern and the point cloud acquisition of the cardboard corners.

Robosense/Velodyne compatibility issue

There's a major compatibility issue between Robosense and Velodyne: Despite the fact that both sensors are totally the same in their technical architecture and displaying, their internal software has a different functioning. The Robosense is generating ring values with XYZ1 and Velodyne with XYZ1R. It naturally produces errors when computing one of these sensors with the method from the other one. Velodyne is still very expensive in the market, and Robosense give the scientist a good opportunity to experiment with these technologies and find new approaches for 3D reconstruction and robotics.

That's why the ROS independent community has developed tools in order to convert the Robosense point cloud format into Velodyne format. It doesn't change anything about the functioning of the sensors, and it's calculus or environment interpretation but solves compatibility issue.

Pattern point cloud acquisition

The method to calibrate the camera with a Lidar and transform all the points in the LiDAR frame to the monocular camera frame is from Ankit Dhall, Kunal Chelani, Vishnu Radhakrishnana and KM Krishna. It is composed of many packages:

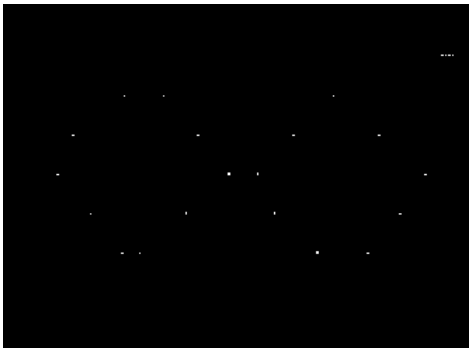


Figure 3.5: Point cloud acquisition and detection of pattern corners

- Aruco_ros used for marker matching and object pose estimation. It will be efficient to detect the pattern thanks to the specific fixed marker. The algorithm can generate it's own marker if desired. see Figure 3.4

- Aruco_mapping has the task to generate a map of the different calibration pattern in the visual environment if the sensor.

After the installation if the different packages on ROS environment, the two patterns can be installed in front of the Lidar and the camera, and with an optimal position angle of 45°to improve the detection of the pattern corners, see Figure 3.6. Then we have to pre-calibrate the extrinsic parameter of the camera to correct the lens distortion or any projection error: We use the simple ROS package "camera calibration" inspired from the Bouguet toolbox [4] which uses checkerboard pattern calibration.

When all the sensors are ready to operate, we have to configure the file with the information of the camera and the Lidar. It consists on:

- Camera resolution
- Distances has measured from the Lidar sensor to the calibration pattern in order to generate a window to include the interesting points and remove unwanted points in the cloud in meter, including:
  - The horizontal (x & -x) dimension distance.

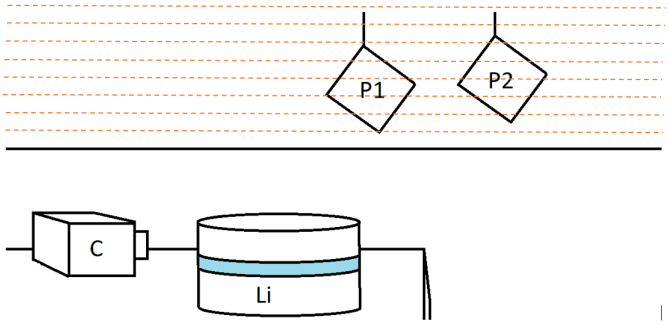


Figure 3.6: Camera-Lidar calibration and fusion: Schematic diagram

  - The vertical (y & -y) dimension distance.
  - The oblique dimension (z & -z) dimension distance.

- The cloud intensity threshold to remove points which has lower intensity value than expected in the window point cloud. By default, it's 0.05.
- The number of markers which signify the number of patterns used.
- The "use camera info topic ?" To force or not the using of the new camera calibration saved in the ROM of the sensor.
- The extrinsic calibration parameters pre-obtained by the Bouguet toolbox.
- MAX_ITER to specify the maximum of iterations to set the rotation and translation matrices.
- initial_rot_x initial_rot_y initial_rot_z from the extrinsic calibration parameters of the camera.

And then we have to measure the distance between the sensor and the pattern and their inside/outside length and width in order to make the board markers being detected by the camera thanks to aruco_ros package which computes an affine transform and display 3D boxes on the markers. The corner points detected and displayed by the Lidar, see Figure 3.5, we have to manually process of marking the following line-segments and compute the intersections.

Finally with the find_transform package, the algorithm has to fin the 2D-3D correspondence between the two sensors in order to calibrate the camera with the Lidar and fuse the pixels from the camera and the point cloud from the Lidar with the "lidar_camera_calibration" package.

Chapter 4

Image disparity enhancement by Deep Fusion method

4.1 Introduction

In this chapter we will discuss about a new method of image disparity enhancement by the fusion of reference RGB images and their disparity equivalence.

The computing of usable disparity maps in stereo or monocular vision without losing information is very difficult, mostly if we are using the algorithm in outdoor with variable environment. It results a lot of problems on the visual recognition of the environment and may cause serious accidents with autonomous vehicles for example. Usually the most important information in the image frame like cars, people, streetlights or side walk are dissolved into the ground. This is due to bad weather like rain, snow, bad or too much illumination, etc.. and to improve this obtained disparity maps, we focused our looking on the fusion technique.

This technique, based on smoothing algorithm has no background for disparity cases and needs different approaches in comparison with other image enhancement like Zhaodong Liu and al. 2016 [29] to enhance information with multi-focus images, Wenda Zhao, Huimin Lu, and Dong Wang. 2017. [49] for image spectral domains or Frosti Palsson, Johannes R. Sveinsson and Magnus O.Ulfarsson. 2017. [37] to enhance the resolution of low spatial Hyperspectral images and and Chiman Kwan, Joon Hee Choi al. 2017 [26] using super-resolution approach. Each non-conventional image enhancement technique, whether it is based on fusion or other methods, needs special approaches to be correctly controlled on its data processing.

Our approach use the fusion of RGB images and their disparity representation in order to learn a Deep Convolutional Neural Network the potential initial RGB image representation of given disparity image without any destruction of information. This fusion is based on smoothing algorithm with the aim of improving certain important details that RGB-to-disparity algorithm couldn't compute correctly due to bad illumination, weather, non-predictable artefacts, environment, etc.

Some Deep CNN architectures are tested from the simplest convolutional network ConvNet to more complex like U-NET or RESnet 50 composed of encoders and decoders. The difference between each architecture is mostly the computation speed. The type of architecture has a low influence on the learning accuracy:

4.1.1 Convolutional neural network Architectures

CONVNET

The architecture of Convet network, see Figure 4.1 is based on the connectivity patterns of neurons in Human brain with n layers composed of m virtual neurons and each layer is followed by a Convolution matrix kernel or Pooling matrix. this technique has the advantage to not flattering images into a vector which may causes bad accuracy in case of complex images. The convolutional layer is composed of a kernel matrix, a filter (with n dimension), which computes a matrix multiplication with a portion of the image and move to the right, and down to the left until the entire image is computed. The type of the kernel, depending on the computation, could be an Identity matrix, Edge detection, Sharpen, Blurring or Gaussian for the most common.
The pooling matrix has the function to reduce the spatial size of convolved features to extract special features and reduce the computing power, increased by the computation of convolutional layers. Usually, a convolutional layer is followed by a Max, Average, or Sum pooling or ReLU layer:

- Max-pooling returns the maximum value from the portion of the image, depending to the kernel size.
- Average-pooling returns the average value.
- Sum-pooling return the sum of all elements in the feature map

The ReLU layer (Rectified Linear Unit) is composed of and activation function like $max(0, x)$ to increase the non-linearity of images by removing all negative values in the features.

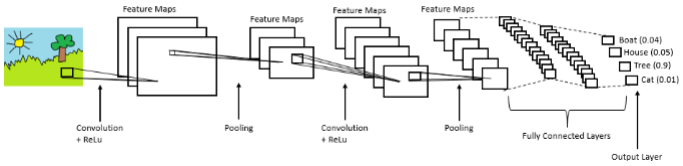


Figure 4.1: Simple ConvNet architecture

The network is ended by a fully connected layer. This last layer is flattened into a vector to be included in a fully connected neural network to create a model by combining all the features together. An activation function such as Softmax or Sigmoid is added to classify the final outputs. The fully connected layer is only used to classify elements with labels.

U-NET

The U-NET architecture network [34] is commonly used for fast and precise image segmentation and much more in Biomedical domain. This network is very different to the simple ConNet but uses the same kind of functions. The architecture looks like a "U", divided into three different parts, sees Figure 4.2:

- The contraction made of blocks of convolution layers with ReLU, followed by max-pooling. The number of kernels on each feature maps doubles after each block. That allows to learn complex images.
- The bottleneck which is the bridge between contraction and expansion block.
- The expansion composed of convolutional layer with ReLU and upsampling layer in order to half the number of feature channels and concatenate with each corresponded cropped feature map.

The pixels from the final segmentation result could be classified into classes thanks to a Softmax function applied to each pixel, followed by the loss function.

Residual Network (Res-Net) is a U-NET architecture used to solve the vanishing gradient problem due to the using of too many layers. The increasing of the layers makes the gradient small, expand the chances to not be correctly updated through back propagation, and may loose information. Res-Net solve this issue using the identity matrix by multiplying the gradient by one when the back propagation is done.

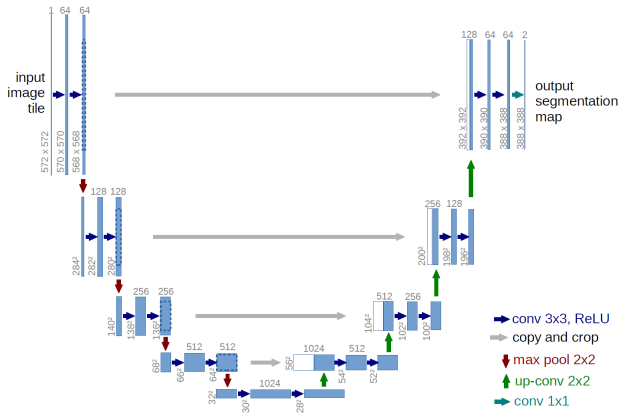


Figure 4.2: Simple U-NET Architecture

4.1.2 Cost functions

Cost function 1 : Woodford and al. 2009

The fusion is based on smoothing algorithm, which is, indeed, a kind of destructive method, but at the present time, one of the best way to merge information together. Deep Convolutional Neural Network will be used by optimizing energy function, specially written to solve or optimize complex geometrical or probabilistic issues. Our first looking was focused on the Smooth loss from O.J. Woodford and al. 2009 [46] which is initially used for global stereo reconstruction, but as we see in Equation 4.1, it is divided into two distinct parts:

- $E_{photo}(\mathcal{D})$ corresponding to the data term used to determine the dense disparity map. Not usable in our case. Equation 4.2
- $E_{smooth}(\mathcal{D}$ corresponding to the smooth part. Equation 4.3

$$E(\mathcal{D}) = E_{photo}(\mathcal{D}) + E_{smooth}(\mathcal{D}) \tag{4.1}$$

$$E_{photo}(\mathcal{D}) = \sum_{x} \sum_{i=1}^{N} f(I_i^{\pi}(x, D(x)) - I_0(x), V_x^i) \tag{4.2}$$

$$E_{smooth}(\mathcal{D}) = \sum_{\mathcal{N} \in \mathbb{N}} W(\mathcal{N}) \rho_s(S(\mathcal{N}, \mathcal{D})) \tag{4.3}$$

With ρ_s is a truncated linear kernel, $\mathcal{N}$ the neighborhood of pixels corresponding to 3×1 and 1×3 patches in the reference image, $S()$ is the second derivative smoothness function,

$S(\mathcal{N}, \mathcal{D}) = S(p, q, r), \mathcal{D} = D(p) - 2D(q) + D(r)$ and $W(\mathcal{N}$ is conditional random field (CRF) weight.

This function is interesting by its advantage to compute the smoothness with a patch of 3×1 and 1×3 pixels patches which increase "monotonically as the neighborhood diverges from collinearity."

Cost function 2 : Qinqnan Fan and al. 2018

Another method to smooth the pixels was studied to be used in our fusion algorithm.

Qinqnan Fan and al. 2018 [9] propose a method that consists in the pixel smoothing in order to eliminate unimportant fine-scale details with the maintaining of the main principal structure of the image. This technique could be used in many graphical applications like tone mapping, image abstraction and our main objective, the image enhancement and edge preserving. See Figure 4.3 The loss function in Equation 4.4 is composed of three different parts:

- $\mathcal{E}_d$ as data term. It minimizes the difference between the input image and smoothed image to ensure a kind of similarity. see Equation 4.5
- $\mathcal{E}_f$ as regulation term to remove unwanted image details by penalizing the color differences between adjacent pixels. see Equation 4.8
- $\mathcal{E}_e$ as edge-preserving term by minimizing the quadratic difference of their edge responses between the guidance edge $E(I)$ and $E(T)$. see Equation 4.7
- λ_f and λ_e as constant balancing weights

In equation 4.7, E_i corresponds to the gradient magnitude from the edge response.

$$\mathcal{E} = \mathcal{E}_d + \lambda_f \times \mathcal{E}_f + \lambda_e \times \mathcal{E}_e \tag{4.4}$$

$$\mathcal{E}_d = \frac{1}{N} \sum_{i=1}^{N} ||T_i - I_i||_2^2 \tag{4.5}$$

$$\mathcal{E}_i(I) = \sum_{j \in \mathcal{N}(i)} |\sum_{c} (I_{i,c} - I_{j,c})| \tag{4.6}$$

$$\mathcal{E}_e = \frac{1}{N_e} \sum_{i=1}^{N} B_i ||E_i(T) - E_i(I)||_2^2 \tag{4.7}$$

Figure 4.3: Results of Qinqnan Fan and al. smoothing method

Where I is the input image, T the output image, N the total number of pixel and $\mathcal{N}_i$ the neighborhood of point i, i the pixel index and c the color channel, B is a binary map and $B_i = 1$ corresponds to an edge and $\frac{1}{N_e}\sum_{i=1}^{N} B_i$ corresponds to the total number of edge points.

$$\mathcal{E}_f = \frac{1}{N}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_h(i)} \tag{4.8}$$

We estimate the weight of pixel pairs $w_{i,j}$ by its color affinity $w^r_{i,j}$ in Equation 4.9 or by its spatial affinity $w^s_{i,j}$ in Equation 4.10 and the determine of p_i, depending on the values of the edge response from the input image $E_i(I)$ and the output image $Ei(T)$. See Equation 4.11

$$w^r_{i,j} = exp(-\frac{\sum(I_{i,c} - I_{j,c})^2}{2\sigma[2}_r) \tag{4.9}$$

$$w^r_{i,j} = exp(-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{2\sigma[2}_s) \tag{4.10}$$

$$p_i, w_{i,j} = \begin{cases} p^{large}, w^s_{i,j} & if\ E_i\ <\ c_1\ and\ E_i(T) - E_i(I)\ >\ c2 \\ p^{small}, w^r_{i,j} & otherwise \end{cases} \tag{4.11}$$

Where $p^{large} = 2$ and $p^{small} = 0.8$ and are conditional to the output image.
By this method, in comparison to $L_{0.8}$ norm smoothing method, the pixels are correctly smoothed with the great respect to the most important edges and details in the image. There are no big damages on the structure, or losing a lot of information from the input image and the proposed detail enhancement application magnifies very well the details from the input image in comparison to the other proposed methods and so could be a good base of studying or coding

Figure 4.4: Detail magnification results of the proposed method compared with previous image smoothing algorithms LLF, WLS, L0 and FGS. In this example, the top row shows the smooth base layers obtained via image smoothing, while the bottom one demonstrates the enhancement results. As can be seen, our algorithm does not over-sharpen the image structures in the smooth image and achieves visually pleasing detail exaggeration effects.

on non-conventional disparity images in regard to the Woodford cost function.

4.1.3 New method

Zehua Fu and al. May 2020, in her Computer Vision thesis about "Confidence Measures in Deep Neural Network based Stereo Matching", propose a new method of Recurrent disparity refinement to refine disparity maps stage by stage and recover the high-frequency details to solve stereo matching problems. The Network is composed of gated recurrent units (GRUs) (which memorize information from the last refinement procedure and improve more details in disparity maps), and a dilated convolution refinement module, "using information from the previous stage to guide the refinement in the later stage".
The inputs are the reference RGB images and their related disparity maps produced from image pair by PSMNET. This two images are fused and the output features are improved by applying "dilated convolution layers with Squeeze-and-Excitation modules". The refined disparity map can be presented as:

$$D_f = f_{RRM}(\hat{D}, T_{ref}) + \hat{D} \tag{4.12}$$

with D_f as refined disparity maps, f_{RRM} the Recurrent Refinement module, $\hat{D}$ the estimated disparity map, I_{ref} the reference RGB image pair. In their feature fusion module, the Reference image pairs (left image) and the estimated disparity maps produced by PSMNET are combined and fed to several dilated Convolutional layers with the followed smooth loss:

$$L(d, \hat{d}) = \frac{1}{N} \sum_{i=1}^{N} smooth_{L_1}(d - \hat{d}) \tag{4.13}$$

$$smooth_{L_1} = \begin{cases} 0.5x^2, & if|x| < 1 \\ |x| - 0.5, & otherwise \end{cases} \tag{4.14}$$

With $smooth_{L_1}$ corresponds to the Smooth L_1 loss and more robust that L_2 loss, $\hat{d}$ is the predicted disparity obtained as the sum of all disparities range from 0 to D_{max} weight by their respective probabilities, c_d the predicted cost, σ the Softmax function and d_i the sub-pixel estimation obtained by a soft argmin function and $C_i(d)$ represent the matching cost over disparity range D

$$d_i = \underset{d}{argmin} C_i(d) \tag{4.15}$$

$$\hat{d} = \sum_{d=0}^{D_{max}} d \times \sigma(-c_d) \tag{4.16}$$

The disparity computation $\hat{d}$ is performed by a winner-take-all strategy, which is a traditional method used in neural Networks in which neurons in a layer compete with each other for activation. Only the neuron with the highest activation stays active while all other neurons shut down.
The implementation of the algorithm was done with Pytorch, optimized with Adam method and learned with KITTI 2012 and 2015 dataset with 200 images, over 1000 epochs, a learning rate of 0.002 and a Batch size set to 4 to get the final model. On the tests, it surpassed the PSMNET by 0.12% and SegStereo by 0.26% on all 3-pixel-error.

This new presented method clearly improve the disparity maps in stereo matching by fusing multi-modal features extracted from disparity maps and the related reference RGB images. It is performed without losing important information (like traditional smoothing methods) and improving step by step can help to keep useful information from the former stage.

4.1.4 Results and Conclusion

In our project, to enhance the disparity images by the fusion of RGB images with their disparity equivalence, we chose the Residual Network to be learned with the smoothing cost function from Qinqnan Fan and al. 2018 [9]. We can't compare the results obtained by this training because the enhancement of disparity images with Deep fusion technique is totally new in the research flow, so it could only be compared with Ground Truth images.

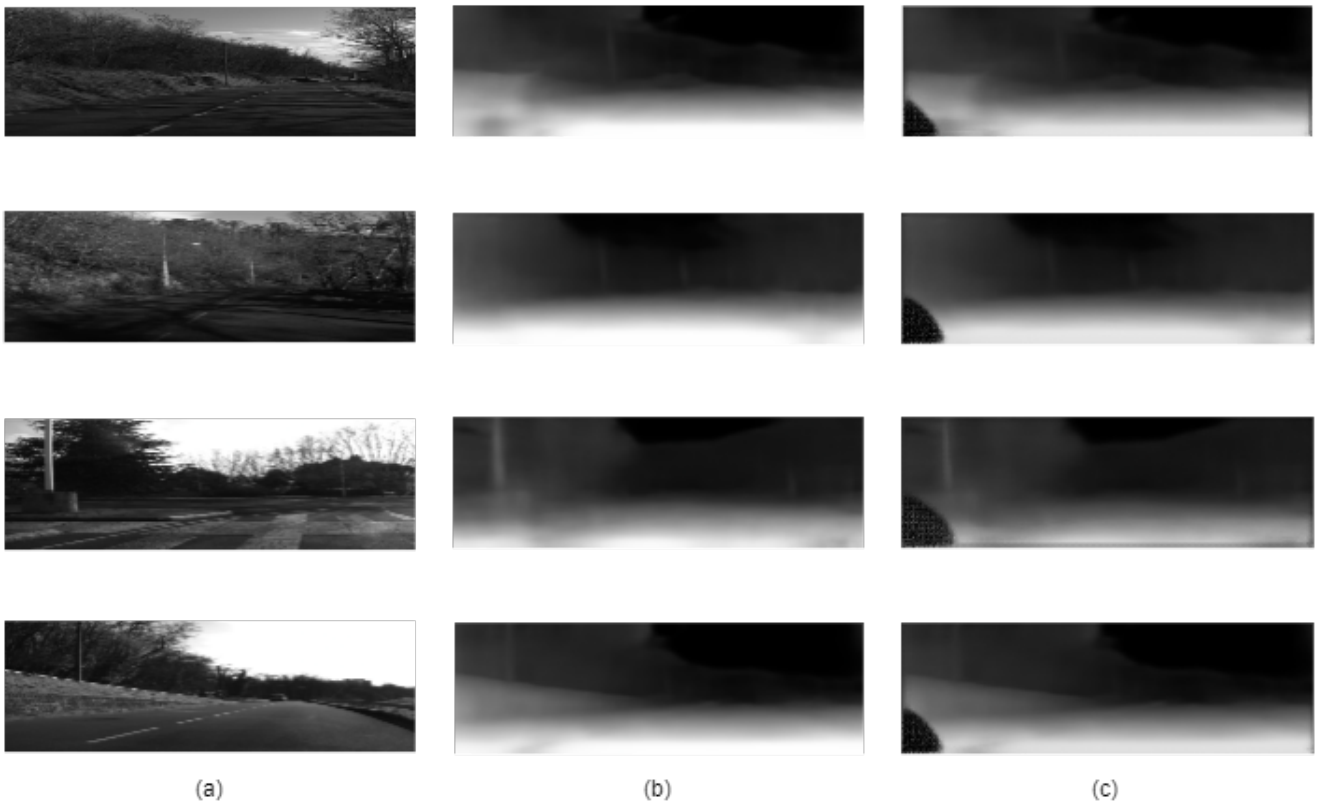


Figure 4.5: Image board showing the reference RGB images (a) transformed into Disparity maps (b) then enhanced with our method (c)

Conclusion

The goal of this project is to find an algorithm to enhance disparity maps by using Fusion with preconceived deep convolutional neural network method. This technique, initially used to smooth region pixels without any big damages in edges and to preserve the most important details from an RGB image, is attempted to be used with disparity maps. The smoothing is one of the most common technique to fuse data, but each type of image needs its own cost function or method to take all the advantages of the fusion process. The result shows some pixel regions are smoothed, but the enhancement is not performed especially on specular areas and dangerous artefacts appears on the down-left image corner: this may due to cost function which has be modified to correspond to our expectations, the network may needs much more images to improve the accuracy and the smoothing power and other pre-processing could has to be done before, like region segmentation/clustering, other methods than smooth could be tested to fuse non-conventional data.
But thanks to this first approach, we could go further, and improve or completely change the algorithm by another one, closer to the reality of the data we want to exploit.
This could be possible by the studying on the new thesis of Zehua Fu and al. May 2020 and her innovative non-destructive method which could represent a new area to exploit for non-conventional images.

Appendix A

The first appendix

Marker coordinates	
Type of marker for first pattern	Data
Number of board used	2
Length (s1)	50
Breadth (s2)	50
Border_width_along_length (b1)	1.75
Border_width_along_breadth (b2)	1.75
Edge_length_of_ArUco_marker (e)	17.1
Type of marker for second pattern	
Length (s1)	50
Breadth (s2)	50
Border_width_along_length (b1)	1.75
Border_width_along_breadth (b2)	1.75
Edge_length_of_ArUco_marker (e)	17.1

Table A.1: Table of marker coordinates file

Figure A.1: Bad point cloud pattern reconstruction due to bad marker estimation

Camera/Lidar configuration file	
Configuration parameter	Data
Resolution	640x480
-x/+x	-1000 1000
-y/+y	-1000 1000
-z/+z	-1000 6.5
cloud_intensity_threshold	0.0003
number_of_markers	2
use_camera_info_topic	0
fx 0 cx 0	1534.64741 0.00000 330.45098 0.0
0 fy cy 0	0.00000 1530.89503 156.22079 0.0
0 0 1 0	0.00000 0.00000 1.00000 0.0
MAX_ITERS	100
initial_rot_x initial_rot_y initial_rot_z	1.57 -1.57 0
lidar_type	0

Table A.2: Table of Camera/Lidar configuration file

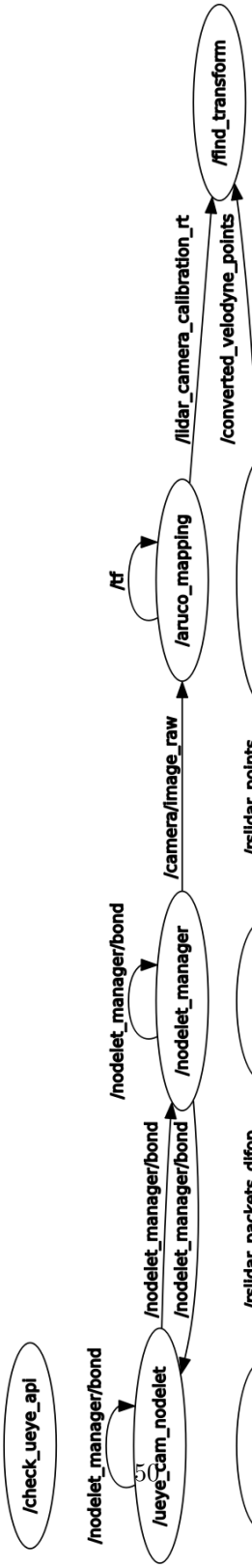


Figure A.2: The graph of the Camera_lidar calibration algorithm

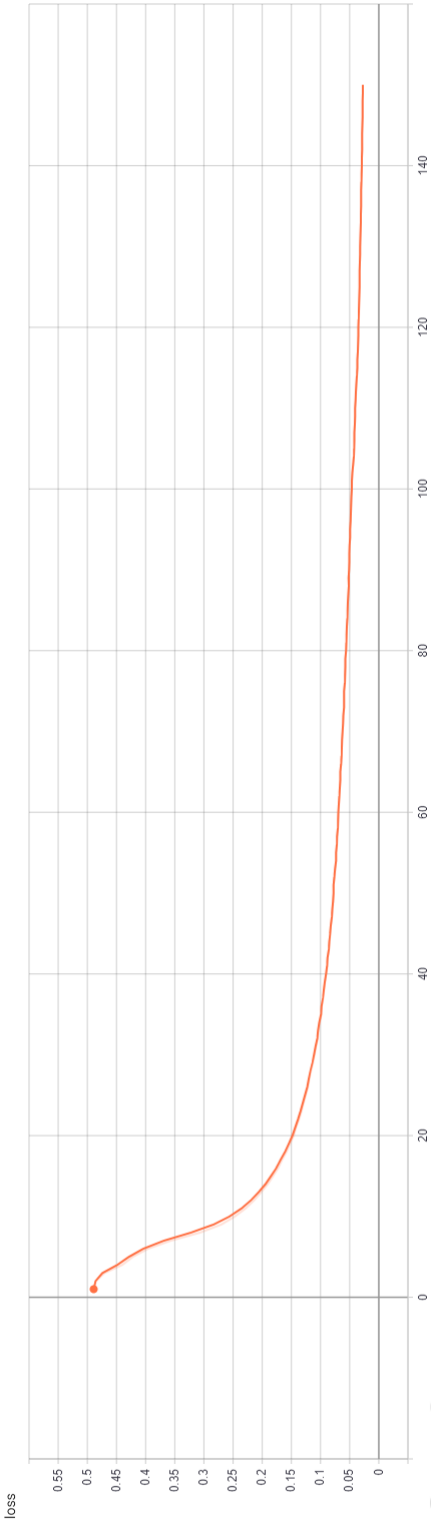


Figure A.3: Loss curve from our learning method

Bibliography

[1] Saeed Anwar, Chongyi Li, and Fatih Porikli. Deep underwater image enhancement. *arXiv preprint arXiv:1807.03528*, 2018.

[2] Konstantinos Batsos and Philippos Mordohai. Recresnet: A recurrent residual cnn architecture for disparity map enhancement. In *2018 International Conference on 3D Vision (3DV)*, pages 238–247. IEEE, 2018.

[3] Romil Bhardwaj, Gopi Krishna Tummala, Ganesan Ramalingam, Ramachandran Ramjee, and Prasun Sinha. Autocalib: automatic traffic camera calibration at scale. *ACM Transactions on Sensor Networks (TOSN)*, 14(3-4):1–27, 2018.

[4] J.-Y. BOUGUET. Camera calibration toolbox for matlab. *http://www.vision.caltech.edu/bouguetj/calib$_d$oc/index.html*, 2004.

[5] Juan Castorena, Ulugbek S Kamilov, and Petros T Boufounos. Autocalibration of lidar and optical cameras via edge alignment. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2862–2866. IEEE, 2016.

[6] Yu-Sheng Chen, Yu-Ching Wang, Man-Hsin Kao, and Yung-Yu Chuang. Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6306–6314, 2018.

[7] Jason Deglint, Andrew Cameron, Christian Scharfenberger, Hicham Sekkati, Mark Lamm, Alexander Wong, and David A Clausi. Auto-calibration of a projector–camera stereo system for projection mapping. *Journal of the Society for Information Display*, 24(8):510–520, 2016.

[8] Ankit Dhall, Kunal Chelani, Vishnu Radhakrishnan, and K. M. Krishna. Lidar-camera calibration using 3d-3d point correspondences. *CoRR*, abs/1705.09785, 2017.

[9] Qingnan Fan, Jiaolong Yang, David Wipf, Baoquan Chen, and Xin Tong. Image smoothing via unsupervised learning. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018.

[10] Vincent Fremont, Philippe Bonnifait, et al. Extrinsic calibration between a multi-layer lidar and a camera. In *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 214–219. IEEE, 2008.

[11] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956, 2017.

[12] Yukang Gan, Xiangyu Xu, Wenxiu Sun, and Liang Lin. Monocular depth estimation with affinity, vertical pooling, and label enhancement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 224–239, 2018.

[13] Pedram Ghamisi, Bernhard Höfle, and Xiao Xiang Zhu. Hyperspectral and lidar data fusion using extinction profiles and deep convolutional neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(6):3011–3024, 2016.

[14] Xiaojin Gong, Ying Lin, and Jilin Liu. 3d lidar-camera extrinsic calibration using an arbitrary trihedron. *Sensors*, 13(2):1902–1918, 2013.

[15] Alexander Hanel and Uwe Stilla. Iterative calibration of a vehicle camera using traffic signs detected by a convolutional neural network. In *VEHITS*, pages 187–195, 2018.

[16] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Jonathan Eisenmann, Matthew Fisher, Emiliano Gambaretto, Sunil Hadap, and Jean-François Lalonde. A perceptual measure for deep single image camera calibration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2354–2363, 2018.

[17] Vaishnavi Hurakadli, Sujaykumar Kulkarni, Ujwala Patil, Ramesh Tabib, and Uma Mudengudi. Deep learning based radial blur estimation and image enhancement. In *2019 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–5. IEEE, 2019.

[18] Sei Ikeda, T. Sato, and N. Yokoya. High-resolution panoramic movie generation from video streams acquired by an omnidirectional multi-camera system. pages 155 – 160, 01 2003.

[19] David Jarron, Derek D Lichti, Mozhdeh M Shahbazi, and Robert S Radovanovic. Multi-camera panormamic imaging system calibration. 2019.

[20] Junho Jeon and Seungyong Lee. Reconstruction-based pairwise depth dataset for depth image enhancement using cnn. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 422–438, 2018.

[21] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang. Enlightengan: Deep light enhancement without paired supervision. *arXiv preprint arXiv:1906.06972*, 2019.

[22] Luyang Jing, Taiyong Wang, Ming Zhao, and Peng Wang. An adaptive multi-sensor data fusion method based on deep convolutional neural networks for fault diagnosis of planetary gearbox. *Sensors*, 17(2):414, 2017.

[23] Otto Korkalo, Tommi Tikkanen, Paul Kemppi, and Petri Honkamaa. Auto-calibration of depth camera networks for people tracking. *Machine Vision and Applications*, 30(4):671–688, 2019.

[24] Xiaodong Kuang, Xiubao Sui, Yuan Liu, Qian Chen, and Guohua Gu. Single infrared image enhancement using a deep convolutional neural network. *Neurocomputing*, 332:119–128, 2019.

[25] Kiho Kwak, Daniel F Huber, Hernan Badino, and Takeo Kanade. Extrinsic calibration of a single line scanning lidar and a camera. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3283–3289. IEEE, 2011.

[26] Chiman Kwan, Joon Hee Choi, Stanley Chan, Jin Zhou, and Bence Budavari. Resolution enhancement for hyperspectral images: A super-resolution and fusion approach. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6180–6184. IEEE, 2017.

[27] Chongyi Li, Saeed Anwar, and Fatih Porikli. Underwater scene prior inspired deep underwater image and video enhancement. *Pattern Recognition*, 98:107038, 2020.

[28] Shutao Li, Xudong Kang, Leyuan Fang, Jianwen Hu, and Haitao Yin. Pixel-level image fusion: A survey of the state of the art. *information Fusion*, 33:100–112, 2017.

[29] Zhaodong Liu, Yi Chai, Hongpeng Yin, Jiayi Zhou, and Zhiqin Zhu. A novel multi-focus image fusion approach based on image decomposition. *Information Fusion*, 35:102–116, 2017.

[30] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar. Llnet: A deep autoencoder approach to natural low-light image enhancement. *Pattern Recognition*, 61:650–662, 2017.

[31] Kaiyue Lu, Shaodi You, and Nick Barnes. Deep texture and structure aware filtering network for image smoothing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 217–233, 2018.

[32] Steve McGuire, Christoffer Heckman, Daniel Szafir, Simon Julier, and Nisar Ahmed. Extrinisic calibration of a camera-arm system through rotation identification. *arXiv preprint arXiv:1812.08280*, 2018.

[33] Faraz M Mirzaei, Dimitrios G Kottas, and Stergios I Roumeliotis. 3d lidar–camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization. *The International Journal of Robotics Research*, 31(4):452–467, 2012.

[34] Aditi Mittal. Introduction to u-net and res-net for image segmentation. `https://towardsdatascience.com/introduction-to-u-net-and-res-net-for-image-segmentation-9afcb432ee2f`, 2019.

[35] Naveed Muhammad and Simon Lacroix. Calibration of a rotating multi-beam lidar. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5648–5653. IEEE, 2010.

[36] Ashley Napier, Peter Corke, and Paul Newman. Cross-calibration of push-broom 2d lidars and cameras in natural scenes. In *2013 IEEE International Conference on Robotics and Automation*, pages 3679–3684. IEEE, 2013.

[37] Frosti Palsson, Johannes R Sveinsson, and Magnus O Ulfarsson. Multispectral and hyperspectral image fusion using a 3-d-convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 14(5):639–643, 2017.

[38] Gaurav Pandey, James R McBride, Silvio Savarese, and Ryan M Eustice. Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[39] Yoonsu Park, Seokmin Yun, Chee Sun Won, Kyungeun Cho, Kyhyun Um, and Sungdae Sim. Calibration between color camera and 3d lidar instruments with a polygonal planar board. *Sensors*, 14(3):5333–5353, 2014.

[40] Zoltan Pusztai and Levente Hajder. Accurate calibration of lidar-camera systems using ordinary boxes. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 394–402, 2017.

[41] S. N. Raza, H. Raza ur Rehman, S. G. Lee, and G. Sang Choi. Artificial intelligence based camera calibration. In *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, pages 1564–1569, 2019.

[42] 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, MFI 2008*, pages 214–219, South Korea, August 2008.

[43] Johannes Schneider and Wolfgang Förstner. Bundle adjustment and system calibration with points at infinity for omnidirectional camera systems. *Photogrammetrie - Fernerkundung - Geoinformation*, 2013:309–321, 08 2013.

[44] Huihui Song, Qingshan Liu, Guojie Wang, Renlong Hang, and Bo Huang. Spatiotemporal satellite image fusion using deep convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(3):821–829, 2018.

[45] Afonso M Teodoro, Jose M Bioucas-Dias, and Mario AT Figueiredo. A convergent image fusion algorithm using scene-adapted gaussian-mixture-based denoising. *IEEE Transactions on Image Processing*, 28(1):451–463, 2018.

[46] Oliver Woodford, Philip Torr, Ian Reid, and Andrew Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *IEEE transactions on pattern analysis and machine intelligence*, 31(12):2115–2128, 2009.

[47] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 636–651, 2018.

[48] Yunlong Yu and Fuxian Liu. Aerial scene classification via multilevel fusion based on deep convolutional neural networks. *IEEE Geoscience and Remote Sensing Letters*, 15(2):287–291, 2018.

[49] Wenda Zhao, Huimin Lu, and Dong Wang. Multisensor image fusion and enhancement in spectral total variation domain. *IEEE Transactions on Multimedia*, 20(4):866–879, 2017.

[50] Zhiqin Zhu, Hongpeng Yin, Yi Chai, Yanxia Li, and Guanqiu Qi. A novel multi-modality image fusion method based on image decomposition and sparse representation. *Information Sciences*, 432:516–529, 2018.