Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

R&D Project

Fusing 3d Range Measures with Video Data for Robust Obstacle Detection and Avoidance (Comparative Evaluation, Analysis and Initial Implementation)

Gautam Kumar Jain

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

Supervised by

Prof. Dr. Erwin Prassler
Msc. Sebastian Blumenthal

January 2020

I, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely my own work.

Date	Gautam Kumar Jain

Abstract

The combination of different visual sensors play a major role in applications like obstacle detection in autonomous driving, robot navigation, capturing the 3D representation of the surrounding environment, motion tracking. In all of the mentioned applications, the main functionality is the creation of depth image that is acquiring the depth information from the surroundings in terms of a depth image. There are two types of sensors that are used to measure depth cues active and passive sensors. Active sensors process the reflected light from the objects that they emit into the surroundings, whereas passive sensors rely on external light sources to capture depth. Depth images can be obtained by using active range sensors like Lidar, TOF (Time of Flight) cameras, structured light cameras or passive sensors like Stereo cameras. Passive sensors can not measure depth in textureless surfaces and repeated patterns whereas active sensors are prone to light saturation, noisy and sparse depth measurements. Active sensors like Lidar when it is employed in outdoor applications give a point cloud that is sparse in nature, as point cloud is resampling of all the 3d points in the scene, leads to missing information about the scene. When that point cloud is projected into an image plane it gets even sparser having some of the pixels with no depth values. To complete this missing information we can use external information in the form of RGB color images from a high-resolution color camera. And this task of filling missing information in a depth image is called depth completion. Similarly, active and passive sensors are fused to complete and upsample, which is to increase the resolution of depth image. This project will provide the reader with an in-depth literature survey for the task of depth estimation and throws light on the state of the are methods for the task of depth completion for a single view. The main aim of this project is to evaluate the performance of combining low-cost active range sensors with RGB color information for the task of depth completion.

Acknowledgements

I would like to express my gratitude and thanks to Prof. Dr. Erwin Prassler for his guidance and support throughout this RD project and for providing me the opportunity to work on this topic. I would like to thank Mr. Sebastian Blumenthal for the constant discussions and patiently listening to my doubts for the project, which further helps me to complete this project successfully. I would also like to thank my colleagues at Hochschule Bonn Rhein Sieg for the long discussions that helped me a lot in the long run. In the end, I would like to thank my family and friends for their support and love.

Contents

List of Figures

List of Tables

1

Introduction

"In 3D computer graphics, a Depth map is an image or image channel that contains information relating to the distance of the surfaces of scene objects from a viewpoint " [1]. Depth imaging is an important branch of computer vision that has a lot of applications in fields like Autonomous driving, microbiology, automation and many robotics tasks like obstacle detection, mapping, localization, and 3d reconstruction. With the development in the field of optics capturing depth through sensors can be done in real-time.

Figure 1.1: A color image of a scene and its respective depth image from NYU Depth V2 dataset[65]

In simple words, an RGB image has three color channels(Red, Green, Blue) with values ranging from 0-255, in a depth image each value of the pixel will denote some

[1] https://en.wikipedia.org/wiki/Depth_map

values of depth in meters or millimeters depending on the sensor we are using and each sensor has a particular scale by which you have to divide with all the pixels in the depth image while processing a depth image.

With sensors like Time of flight Cameras, Lidar, Stereo cameras, Structured light cameras depth information can be obtained with varying accuracy and resolution. Such sensors are selected depending on the application and environmental conditions in which sensors are used in. Sometimes combination of multiple sensors is used to capture depth and fused to counter the limitation of one sensor by other, resulting in more refined depth outputs. Depth images obtained by sensors like Lidar and TOF (Time of flight) cameras in outdoor conditions are sparse. Basically for the application in outdoor conditions, not all the objects are fully reflective, the light waves emitted from the sensors sometimes does not come back to the sensor as it is absorbed by the objects in the surroundings leading to incomplete outputs in terms of depth image and in case of a lidar this problem is really visible. When a point cloud from a lidar is projected back to the image plane it results in incomplete depth images with holes in it, therefore we can not design our algorithms on incomplete information, as it will not be able to detect meaningful features like object boundaries, shapes and to counter this situation we use a combination of sensors like Time of flight cameras with a high-resolution RGB(color) camera or a Lidar with a high resolution RGB(color) camera which will in result give us a dense depth image.
For indoor applications, 3d cameras are used like Intel Realsense [2] which gives a dense depth image but when we take 3d cameras in outdoors it will give us bad results because of saturation. 3d cameras work on IR(Infrared rays) and in the conditions like bright sunlight the sunlight will be confused by the sensor as infrared rays which is emitted from the sensor leads to saturation of the sensor.

[2]https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html

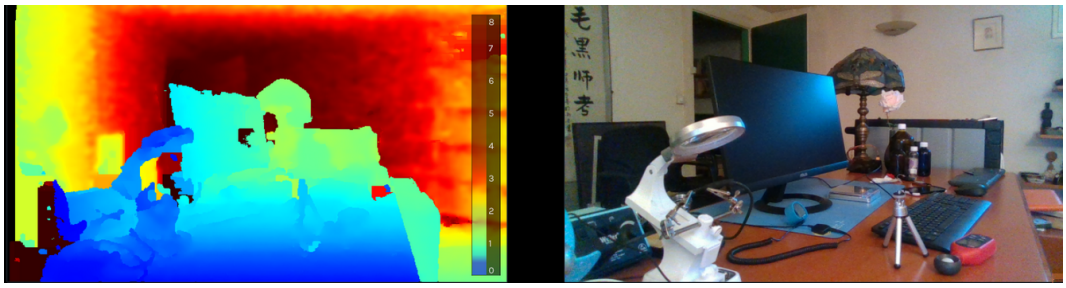


Figure 1.2: A depth image from a Intel Realsense its respective RGB image

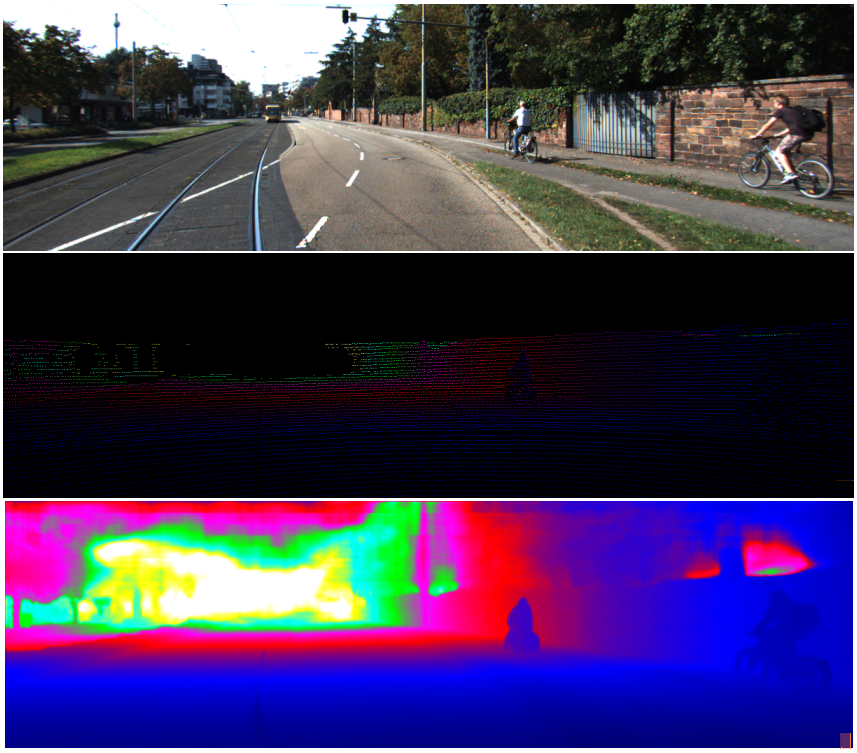


Figure 1.3: (a) RGB image and sparse depth image (b) of an outdoor scene from KITTI depth Completion benchmark[85] and (c) completed dense depth image by [62]

1.1 Motivation

According to a research report by Technavio, smarts sports equipment market is expected to rise by 14.49 billion USD between 2019 and 2023 [3]. In contrast with that, other application domains like autonomous vehicles the market size is worth 54.23 billion USD in 2019, and is expected to rise by 556.67 billion USD by 2026 [4]. The autonomous vehicle industry is looking for an alternative for Lidar as it is expensive and provides noisy depth data. In autonomous driving depth imaging is used for 3d object detection, driver assistance systems and as the application requires real-time operation with fewer chances of error therefore we need a refined or a high-resolution depth image. Moreover as mentioned above depth imaging has a wide variety of applications like robotics, microbiology, etc. and the availability of high-resolution depth images facilitates the improvement of the performance of these applications.

1.2 Challenges and Difficulties

There are only two categories of methods[91] that can be used to obtain the depth image. The first one is active methods in which source of light is thrown on to the objects in the environment and the light source reflected by the objects in the surroundings travel back to the sensor and the depth information is measured from the reflected light. The second method is the passive method where multiple images from more than one view are used or sometimes only one RGB color image is used for obtaining a depth image. The sensors which fall in the active sensors category which are used for the construction of depth maps are, for example, Lidar, Kinect, Time of flight cameras, Infrared cameras. Whereas passive sensors, like stereo cameras, monocular cameras. Active sensors give the sparse depth maps, thus depth image provided by them has some pixels with missing values due to the presence of some non reflecting objects. Thus the depth maps provided by these sensors have holes that is why they are called sparse and are not comparable in terms of resolution as a comparison to high-resolution color images. Depth maps produced by passive sensors like stereo cameras are very dense that is high in resolution but can not fully

[3] https://www.finanznachrichten.de/nachrichten-2019-04/46512183-global-smart-sports-equipment-market-2019-2023-increasing-demand-for-robotics-in-the-market-to-boost-growth-technavio-004.html

[4] https://www.alliedmarketresearch.com/autonomous-vehicle-market

produce the depth maps for textured surfaces and repetitive patterns and also they suffer from the problem of occlusion in which an object is more visible from one view and not visible fully from other views. For outdoor applications obtaining a dense depth, map is done by fusing a passive and an active sensor thus taking the advantages from both of the sensors.
All the state of the art approaches that deal with the creation of depth maps using sparse depth inputs can produce dense, accurate depth maps with resolution that is comparable to high-resolution color images and most of these approaches are based on deep neural networks. But they lack generalizability in terms of the level of the sparsity of depth input and in the cases where the input sparse data is provided from different types of depth sensors. Fast and accurate creation of depth maps in dynamic environments with variation in lighting conditions is also a major concern which is an issue that is not completely solved yet. The high computational complexity of the deep neural network architectures that are used to solve the task is also an issue.

1.3 Problem Statement

In this project, we aim at constructing a dense depth image by fusing 3d range measures from sensors like Lidar or a 3d camera with a high-resolution RGB image. Our solution is focused towards a sports robot that will operate in an outdoor environment with varying illumination conditions and we are using low-cost sensor setup consists of a Hypersen solid-state Lidar [5] , an Intel Realsense 3d camera [6] and a high-resolution Logitech webcam [7]. A survey and state of the art analysis is carried out on the problem of depth completion that is obtaining a dense depth image from a sparse depth image followed by a comparative evaluation of the state of the art approaches on outdoor depth completion dataset called KITTI depth completion benchmark [85] and also on two indoor datasets collected by combination of Hypersen solid-state Lidar and Intel Realsense 3d camera with high-resolution webcam.

[5]https://en.hypersen.com/product/detail/10.html
[6]https://www.intelrealsense.com/depth-camera-d435/
[7]https://www.ebay.de/itm/132644526498

2

Theoretical Background

2.1 Depth Imaging

As discussed above depth imaging has been one of the major tasks in the field of robotics, computer vision, autonomous driving where the major emphasis was given on obtaining information about 3d geometry and structure from the scene. 3d Structure of a scene is represented in different forms like meshes, point clouds, Volumetric models, depth maps, etc. And to capture this information methods like Structure from motion (Sfm), multi-view geometry like stereo matching, SLAM(Simultaneous Localisation and Mapping) have been used, sometimes single view or frame is also taken into account for capturing the depth and the structure and our problem, in this case, is focused on densifying the sparse depth cues which are captured from a single view.

2.1.1 Representation of Depth Image

As briefly explained in the introduction a depth image is an image channel where all the pixels in the channel contains depth information from a single viewpoint in our case we can say that is the position of the camera. Normally we can easily confuse a depth image with a depth map as if you have depth map and a depth image of a scene we can not spot any difference between both of them by looking at it. The depth image also contains depth information where pixels in the image

channels represent the depth information with respect to a 2d image plane to the objects in the scene. A depth image is usually generated by projecting 3D points in the world to a 2D image plane using a pinhole camera model we generally call it as perspective transformation and human vision also works on the same principle. Let us assume a 3D point $P = (x, y, z)$ in the world and when we project the point on 2D image plane using perspective transformation it will give us the value of z coordinate along the optical axis ie. depth of the point from the camera hole where the hole is located according to pinhole camera model. Sometimes other geometrical encodings are also used for representing depth like inverse depth where the pixel values are $(1/z)$ or it contains the value of disparity for a certain pixel in terms of stereo vision. To understand how perspective transformation works we need to understand the camera geometry.

2.1.2 Camera Geometry and Single View Geometry

The projection of 3d world point to an image plane is mapped by a (3x4) matrix called camera matrix which maps the 3D world coordinates into 2D points onto the image plane. This camera matrix has 11 degrees of freedom and is comprised of a two-parameter matrix called intrinsic and extrinsic. The intrinsic parameters manage the situation of the focal point of the camera, focal length in x and y direction and more. Whereas extrinsic parameters will give you the position of the camera with respect to the world frame of reference. We will go into more detail about these parameter matrix. Later we will see that all camera parameters like center of cameras called the focal center, focal lengths, translation and rotation of the camera with world coordinate system can be found by matrix operations.

There are two types of camera models which are categorized based on the distance of the camera from the 3d point, the first type is based on the fact that when the camera center lies at finite distance and the other specialized models are applicable when our camera lies at an infinite distance. Our task belongs to the first case so, we have to understand the most basic camera model assuming that our camera center lies at a finite distance called the pinhole camera model.

As in the above figure we have to project that point $X = (X_w, Y_w, Z_w)^T$ in the world coordinates onto the image plane and as per the pinhole camera model the

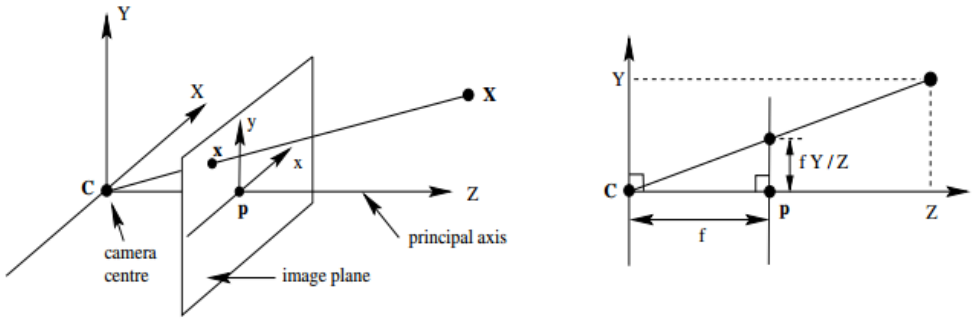


Figure 2.1: Pinhole camera geometry taken from [27] where C is the camera center and p is the focal point where the image plane cuts the principal axis and the camera center is placed at the origin of the world coordinate system.

point on the image plane is the point formed the line joining the point and the camera center cuts the image plane. And with the help of similar triangles we can say that X can be represented as $(fX_w/Z_w, fY_w/Z_w, f)^T$ therefore we can say that final coordinate of the 2D point on the image plane is $X = (fX_w/Z_w, fY_w/Z_w)^T$. We can also represent the projection as mentioned above in the form of matrix transformations.

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} fx \\ fy \\ z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$\text{And } K = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix}$$

Therefore we can rewrite the above equation into $X' = PX$ where X' is the 2D point on the plane and P is 3X4 homogeneous matrix known as camera projection matrix and X is the homogeneous vector of the 3D world point.
Initially we have assumed that camera center lies at the origin of the world coordinate system. Generally, the camera have some rotation and some translation with respect to the origin of world coordinate system. So in taking into account the rotation and translation (4X4) homogeneous matrix is introduced which is called extrinsic camera

parameters.

Figure 2.2: Converting the world coordinates into camera coordinates with the help of extrinsic camera parameters [63]

The resulting camera matrix is made up of intrinsic and extrinsic parameters of the camera.

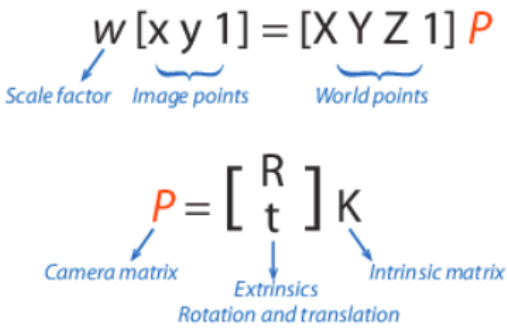


Figure 2.3: transformation of a 3D world point onto an image plane by camera matrix and structure of a camera matrix. [63]

The above-derived model assumes that the image plane as equal scale in both x and y directions, but practically it is not always true, assuming image coordinates are measured in the form of pixels to counter the problem we will introduce a scale factor m_x and m_y here assuming the number of pixels per particular distance and we will also introduce a calibration factor called skew factor in case when exists skew in the pixel ie. when x and y-axis of the pixel is not perpendicular to each other. Thus our intrinsic camera matrix will look like this.

$$K = \begin{bmatrix} \alpha_x & s & c_x \\ 0 & \alpha_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Where $\alpha_x = f * m_x$, $\alpha_y = f * m_y$, c_x and c_y is the position of the optical center and s is the skew coefficient.

So, the end result we can write the camera model as,

$$\begin{bmatrix} x \\ y \\ x \end{bmatrix} = KM_{ext} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

2.2 Taxonomy of Depth Measuring Methods

Depth measurement can be done by many techniques which are classified as contact and non-contact based methods. We will focus more on the non-contact based methods in which we just emit a light wave and process the wave which is emitted from the surface of the object to obtain depth. A brief classification of the depth measurement methods is done below.

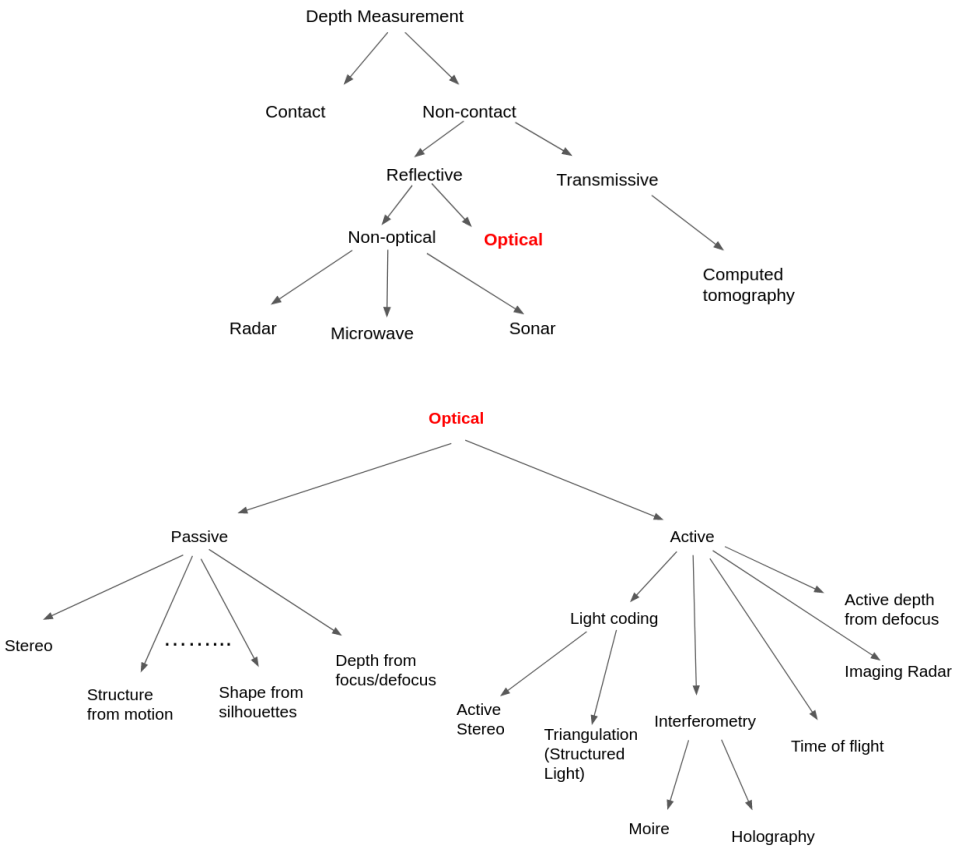


Figure 2.4: Taxonomy for the methods to measure depth (derived from [38],[5])

Further from the taxonomy above, non-contact based methods are divided into reflective and transmissive methods use x-rays for computed tomography, so it is not preferable to use in an environment dealing with human interaction. Later comes the reflective methods which rely on the processing of the reflective light for measuring depth, in these methods sonar, the microwave has used a medium to measure depth. In the case of the microwave, it passes through most of the objects, therefore, it is not an effective solution to measure depth, on the other hand, sonar uses ultrasonic waves which provides inaccurate depth measurements in comparison to using microwaves. So the major development is done on the light spectrum that is visible to human eyes, leading to optical methods. Optical methods are classified into passive and active methods. Active optical methods emit a light wave and process the light which is reflected back from the surface of the object, whereas passive methods don't have their own source of light, they rely on the illumination conditions of the environment. On the same principle sensors used in those methods are also divided into active and passive methods. Some examples of active sensors like Lidar, TOF camera, Structured light cameras, and passive sensors like normal color camera, stereo camera.

2.3 Stereo Vision

Humans can perceive depth from the surroundings with the help of two binocular eyes where both eyes capture two different scenes and brain processes them to create depth perception. Similarly in computer vision depth can be captured using two cameras capturing a scene with a particular distance between them, they are arranged in the same way our eyes are originally there in the human body. The camera used can rely on the light available in the environment or sometimes a projector is used alongside the camera setup differentiating the techniques into active and passive stereo vision.

In Passive Stereo Vision light available in the surroundings are used to capture the scene, previously we discussed how a 3d point can be associated with a pixel in a single image and further we calculated extrinsic and intrinsic parameters for a camera with the help of perspective transform. But we can derive all the properties of a 3D world from a single image, because of the intrinsic ambiguity of the 3D to 2D mapping [28]. For example in the image below you can see that a man is holding

the Leaning Tower of Pisa and we are not able to figure out the depth ambiguities of the scene. So a single image can not tell you about the real 3D ambiguities in a scene, for that we need more than one single view.

Figure 2.5: Single image is not able to clear out the depth ambiguities, if the man is in front of the Leaning Tower of Pisa or standing beside it.

Depth perception from stereo cameras is established on the principle of triangulation. In the figure below two cameras are placed at a certain distance such there is overlapping of the scene from both of the cameras.For a 3d point in the scene, two rays will connected to the focal center of the camera and to capture the 3d structure of the scene we need to do two tasks. First, we need to find the correspondences ie. finding where each pixel in the left image is located on the right image this is called stereo matching. The second thing is the knowledge of the exact camera geometry of both the cameras so that the point can be re-projected into 3D space.

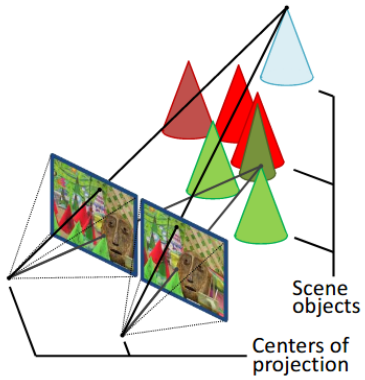


Figure 2.6: Imagination of a 3d scene with paper cones in background taken from Middlebury Stereo Dataset [76]

To understand stereo matching first we need to understand the concept of epipolar geometry. In stereo vision, 3d points are projected on two image planes of both the camera, epipolar geometry helps to provide the relation between the 3D points , projection on the camera image planes.

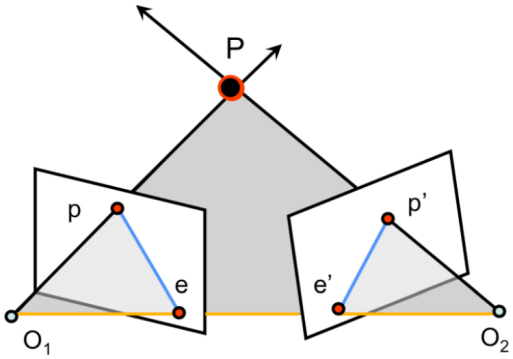


Figure 2.7: Epipolar geometry of a 3D point P for two cameras.

In the figure above, there exist two cameras with focal center at $O1$ and $O2$ which are seeing the same 3d point P and the line connecting both the centers is called baseline. And the plane formed by joining the centers and the point P is called epipolar plane. The projection of P on both camera image planes is p and $p^{'}$. The point e and $e^{'}$ where the epipolar plane cuts the baseline is referred to as epipoles and the line joining the projection of P and the epipoles is called epipolar lines.

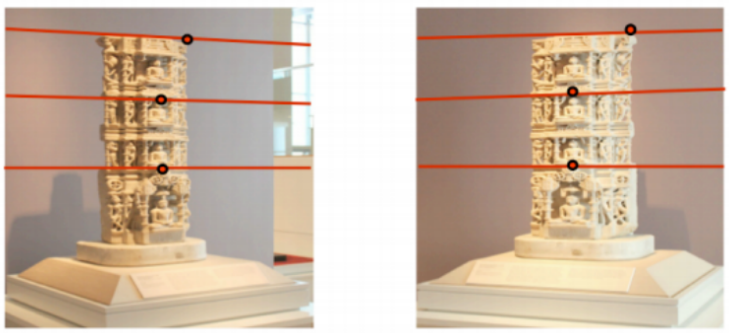


Figure 2.8: Epipolar lines in two images from a stereo camera setup and corresponding points in both of the images [28]

In real-world scenarios we don't know the exact location of the 3D point P in the

scene, however, we know the projection of that point in the camera's image plane. By the process of camera calibration, we will know the exact location and orientation of the stereo camera pairs. So, by that, we can define epipolar planes for all the 3D points observed in both the images and joining the projections and the epipoles are able to find the epipolar lines in both the images. Further during the process of calibration a fundamental matrix is computed which takes care of intrinsic and extrinsic parameters for both the cameras. This fundamental matrix is employed to find the epipolar lines for a 3D point projected into both the images. Thus we are able to establish a connection between the projection of a 3D point on the image plane of both the cameras without knowing the 3D structure of the scene because of that the search for the corresponding points in both the images are restricted to one dimension that is one epipolar lines.

To capture depth from a stereo camera pair we usually perform three processing steps, rectification, matching, and re-projection or conversion of disparity into depth. In the un-calibrated camera images the corresponding epipolar lines for a set of 3D points is curved because of the distortions in the camera lens and further, the images are taken from different orientation as the camera image plane for both the camera is not parallel identical. So the rectification step is carried out that, first makes the curved or distorted epipolar lines straight and after that, the perspective transformation is done on both the images to align the epipolar lines. This is a necessary step to speed up the process of stereo matching.

As the epipolar lines in both of the images are straight and aligned in both the images, every pixel in the right image can be matched with the pixels on the left image that lie on the same epipolar lines. Further single pixel value is not enough to find the exact correspondence a small matching window ($4x4$) is placed at a pixel in the right image against the different window sizes in the same row of the left image. In case of finding a suitable match, we store the correspondence in the form of a disparity map which gives the depth information in terms of offset of a particular pixel in right and left image. This method of matching is called local stereo matching as we are using the neighboring pixels to find the correspondence in the left and right images as this method will not work well in regions with repetitive textures. There exists one more method called global stereo matching where neighboring pixels are considered but correspondences are carried out all at once for all the pixels, these

methods assume the surfaces in the scene are smooth and all are at similar depth and are computationally expensive but more accurate when processed on smooth surfaces.

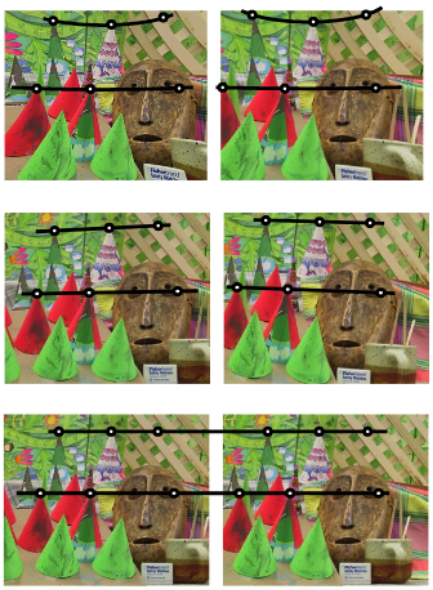


Figure 2.9: rectification process on pair of stereo images [24]

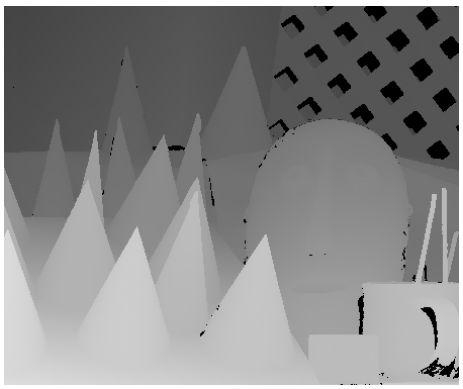


Figure 2.10: A disparity map from Middleburry stereo dataset. [76]

The method used to get depth from disparity is called triangulation and depth is inversely proportional to the disparity and we can also convert these disparity values into X, Y, and Z values for a particular pixel using the camera geometry calculated during calibration giving us point cloud of the particular scene.

In the cases where exists, texture fewer surfaces passive stereo vision does not give good results in that case we generally use active stereo, where a light is projected onto the surface of the scene and patterns are created. In that case, those patterns are matched in stereo image pairs giving better results.

2.4 Structured Light

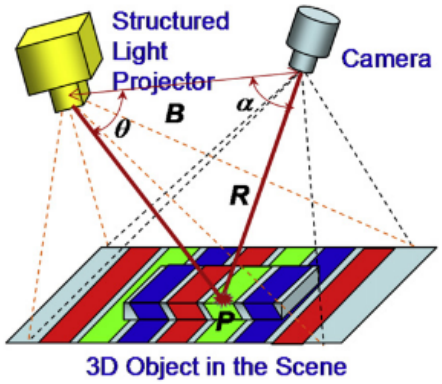


Figure 2.11: An illustration of a structure light camera setup

Passive stereo does not show any results in textureless surfaces to counter those we project a pattern on the 3d object and use that pattern for stereo matching. But instead of using two cameras for that we can use a single camera and a projector which illuminates the scene with spatially varying intensity pattern. Microsoft Kinect[1] is a good example of a device working on the same principle.

The intensity of the projected illumination is represented by a signal like ($I_{ij} = (i, j)$) where (i, j) represents the coordinates of the projected pattern. A video camera or a normal color camera is used to capture the 2d images of the scene. The image is similar to the pattern illuminated by the structured light projector when the scene has no 3D objects but in the case of the 3d objects in the scene where the shape of the pattern is distorted. This is the basic principle of structured light where the information about the 3d surface is collected by the help of distortion in the projected pattern. So, by the principle of triangulation, the geometric relation between a camera and a light projector and the respective illuminated scene is given by

[1]https://en.wikipedia.org/wiki/Kinect

$$R = B\frac{sin(\theta)}{sin(\alpha+\theta)}$$

The major factor for the effective working of this technique is to match the coordinates of the illuminated pattern with the pixels that are captured by the color camera so that depth values can be allocated to that particular pixel. The most simple way of doing this can be illuminating a single-pixel or a row of pixels onto the scene which will speed up the matching problem but in that, we have to take a lot of successive images of the scene which is not possible in terms of real-time applications. To counter this problem many encoding strategies have been proposed and mainly categorized by three important properties[5] as given in the figure below.

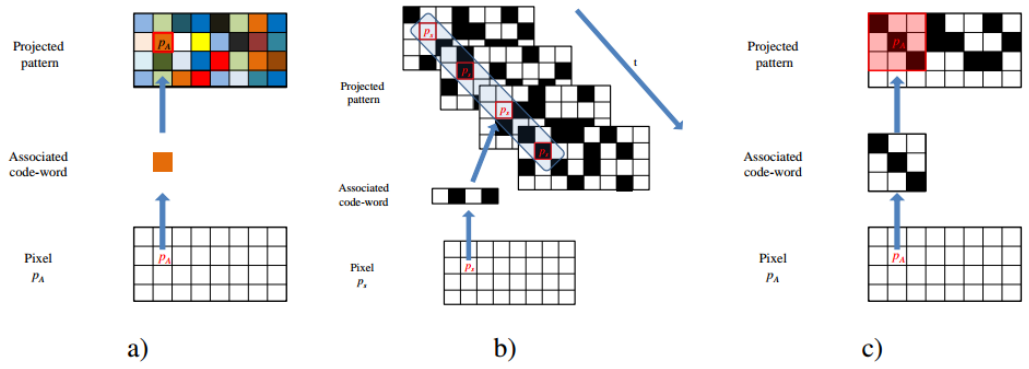


Figure 2.12: Encoding strategies (a) color encoding; (b) time-multiplexing encoding; (c) Spatial encoding [5]

The first encoding strategy used is known as color encoding, where when the number of encodings is more the more information can be stored into the image, but this does not work well when surfaces in the scene have different reflectance in that case binary encodings are used[84]. The next one is the number of images that have to be captured of the particular scene to measure the depth correctly and the last one is the number of pixels that are used for the encoding of a single coordinate of the pattern captured in the image.

Various methods for capturing depth through surface light is there like a sequential method where multiple shots of the scene are taken from the camera to evaluate the matching and depth accurately, single-shot methods are also there. In below figure there is a complete classification of these methods presented by [23].

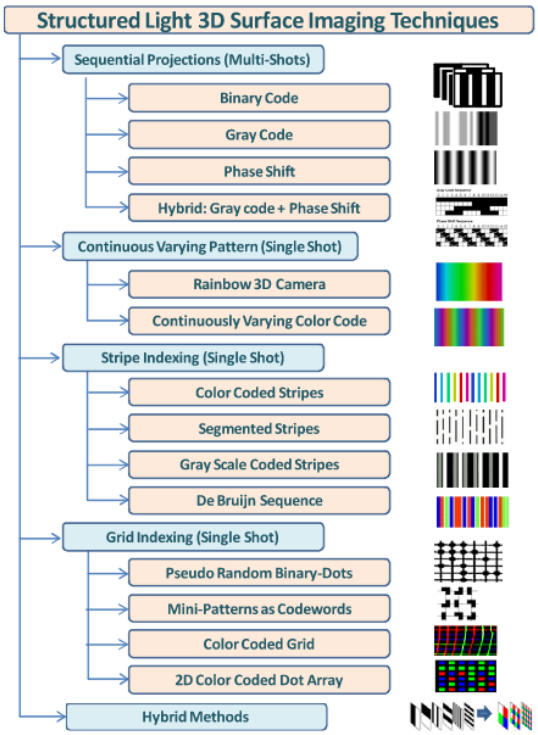


Figure 2.13: Road map of methods for structured light 3D surface imaging techniques derived by [23]

2.5 Time of Flight

Time of flight cameras is also known as Photonic Mixing Device (PMD). The construction of a time-of-flight camera includes an emitter which are nothing but light-emitting diodes and laser diodes which emits light into the surroundings, a receiver which captures the light made up of photodetectors arranged in the form of an array and this principle of illuminating the surrounding objects with a beam of parallel rays, followed by detecting it is to estimate the range or depth is called time of flight(TOF).

One of the most popular ranges measuring sensors is LIDAR (Light Imaging Detection And Ranging) and this also works on the same principle of time-of-flight and they use fine laser beams as illumination medium and because of that they can capture depths with high resolution and can target various objects like metal and non-metal objects which makes them different from normal 3d TOF range sensors.

Most of the range sensing devices need to collect the array of depth measurements and do not rely on single depth value. Lidar is equipped with a scanning mechanism like a rotating mirror to capture the row of adjacent range values sometimes two-axis rotating mirrors are used to capture the vertical depth values. In the same way there exist scannerless devices also that illuminated the while space by emitted, where the reflected light is captured by the array of photodetectors and this is how a normal TOF depth camera work.

So basically there are two ways to measure time-of-flight, one is pulsed light are other one is continuous-wave.

2.5.1 Pulse-Based Method

In the pulse-based method, the emitter emits light pulse which is reflected by the scene and the reflected light is captured by the camera. The photodetectors detect the light which is reflected and the time of flight for the emission to reflection is measured. Usually, the return signal reflected by the scene is very weak to counter that these days a particular form of diodes are used called as Single-photon avalanche diodes (SPAD)[32] are used. As we know the time of flight of the light pulse from emitting to receiving part we can measure the distance by the formula

$$D = \frac{c*T}{2}$$

Where T is the time taken by the light to reach the receptor in seconds and c is the speed of light.

Figure 2.14: Pulsed based time-of-flight measurement.

2.5.2 Continuous Modulation Based Method

In continuous modulation based method time-of-flight is measured in the same way as the previous methods but in this case phase shift of the reflected light is measured and the light used is in the form of a sinusoidal signal. As in the figure below we can see that a sinusoidal signal is used and after the reflection from the scene there is a phase shift in the light signal.

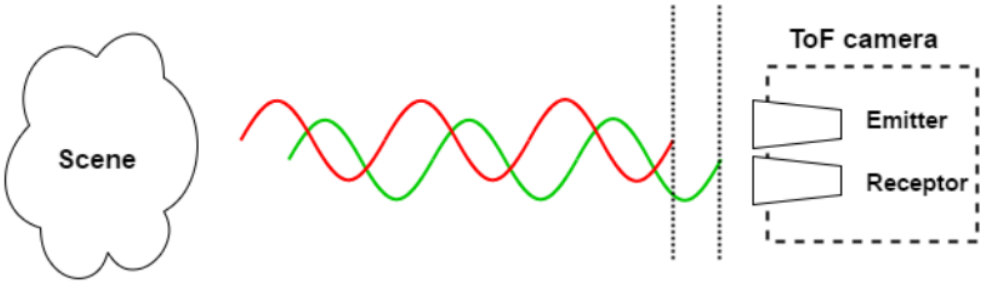


Figure 2.15: Continuous modulation based time-of-flight measurement.

2.6 Fusion of Multiple Sensors

Sometimes combination of multiple sensors is used to measure depth. The term fusion is described as the combination of multiple measurements from particular sensors or a combination of measurements from two different sensors that operate on two different principles like an active and a passive sensor. Taking an example of passive and active fusion combination of passive stereo with structured infrared light thus dealing with all the shortcomings in passive stereo. Fusion of stereo camera and a TOF camera is proposed for the first time by [48] thereby increasing the speed of matching the correspondences called stereo matching by resolving the stereo ambiguities and it was shown that both the sensory inputs handle the shortcomings of each other and in the end able to obtain a dense depth map suited for real-time applications [18], [20], [2], [79]. [64] proposed a survey of TOF-stereo fusion.

Similarly with the development in time-of-flight (TOF) cameras new active systems for depth measurement using TOF cameras emerged [2] [3]. As TOF cameras

[2]https://3dvsystems.com/

[3]https://www.pmdtec.com/html/pdf/pmdPhotonICs$_1$9k_S3.*pdf*

have low resolution and they can capture depth even in low textured areas and they are cheaper than laser scanners. Further, one more type of sensor called Lidar (Light Detection and Ranging) which also operates on-time flight principle is fused with high-resolution RGB images from a color camera, where incomplete depth images are completed. [67] argues why the fusion of depth measurements with RGB information of the scene is beneficial for the task of object detection. Many recent approaches [70], [21] show that Lidar and RGB fusion leads to good results viable for real-time applications in outdoor conditions like autonomous driving.

In the end, we can say that the inability of the sensors to measure the depth of transparent objects or not ideal Lambertian surfaces [4] leads to the fusion of one or more sensor modality to counter the results. Some approaches [92], [88] has worked in that particular matter but due to lack of high-resolution arrays for sensors still there exist no practical solution which can be applied to the applications that utilizes depth information in real-time and outdoor environments.

[4] https://en.wikipedia.org/wiki/Lambertian_reflectance

3

Literature Review

3.1 Single View Depth Estimation

The literature review of this project will revolve around the problem of single view depth estimation. To explain this scenario just imagine we are using a time of flight pulsed light-based Lidar to capture the depth measurements from lidar the depth measurements are represented in the form of depth image. While the observation of the depth image we see that it is sparse in nature that is only a particular set of pixels have values and other remaining pixels have zero values. This means that those pixels have no information about depth because of many reasons mostly in outdoor applications not all the objects in the scene are fully Lambertian [1] this means the light emitted from the sensor does not come back to the sensor, therefore there is no reading for that particular line of scan results in sparsity in point cloud and when that point cloud is projected onto an image plane via perspective transform which we talked about in section 2.1.2 results in even more sparsity. Therefore the main motive to solve this task is to acquire a dense depth image with incomplete sensor measurements and information. So the task of completing the missing information in a sparse depth map is called depth completion and depth inpainting using with and without color images as surplus information used to complete depth. But the case when only color images are used to complete depth then it called depth completion. This project revolves around the problem of depth completion but we will take a

[1] https://en.wikipedia.org/wiki/Lambertian_reflectance

look at other variants also which enhances depth in other ways in later sections.

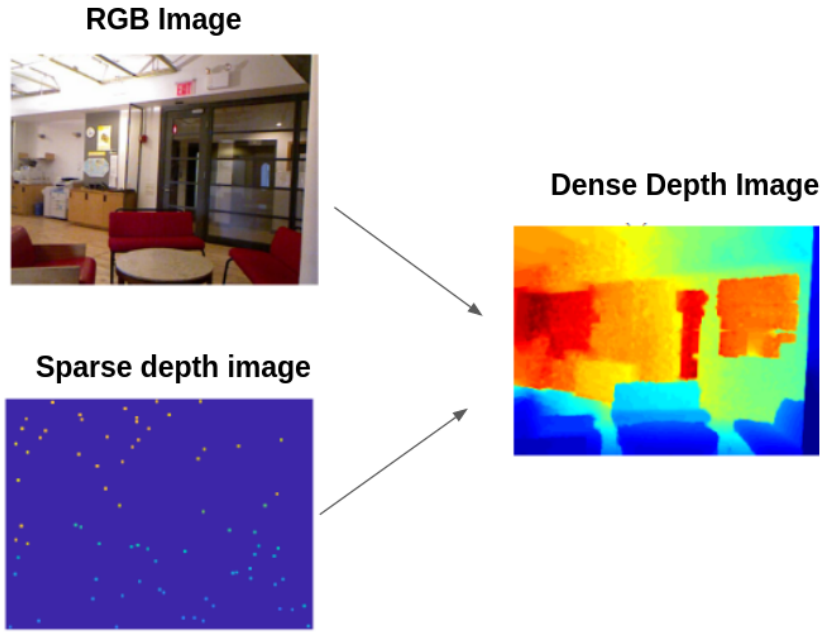


Figure 3.1: single view sparse depth is completed using RGB image of the same scene and output is a dense depth derived from [61].

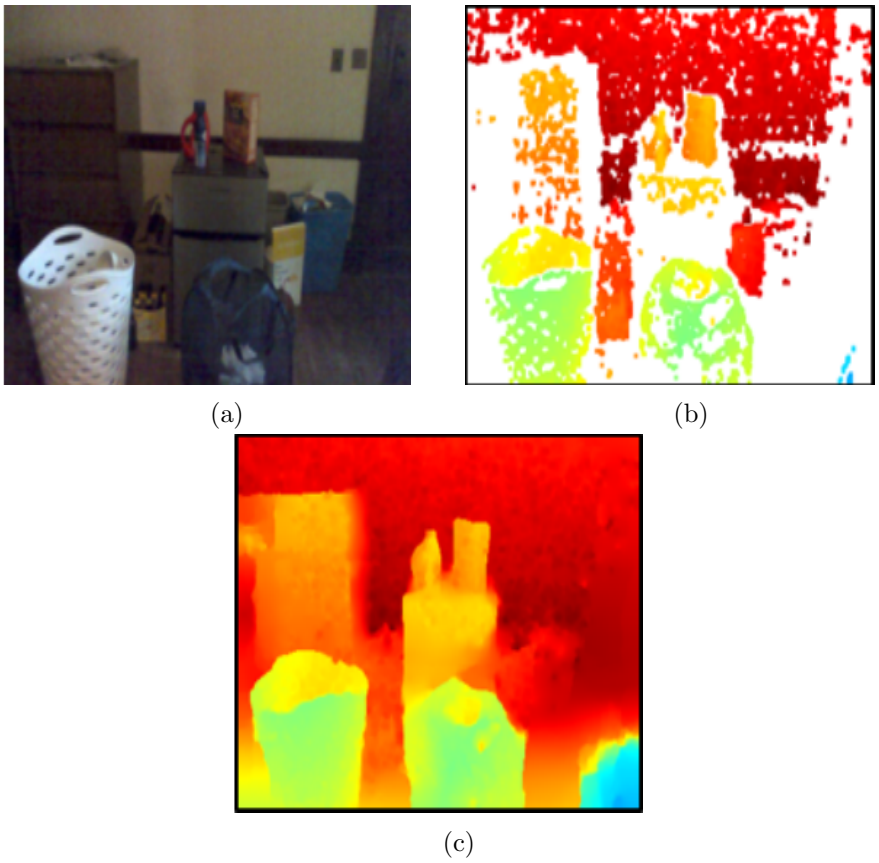
3.2 Overview of Applications of Single View Depth Estimation

As already discussed depth estimation is an important task in computer vision. So the range of its applications areas is also significant.

3.2.1 Depth Sensor Improvement

Talking about the applications of the single view depth estimation one the major contribution of this task is the depth sensor enhancement, by the definition of depth estimation we are filling the incomplete sensor measurements thus enhancing the depth sensor measurements. Like in figure 1.3 it is can be seen that initial sparse depth image from Velodyne HDL-64 is completed using single view depth estimation algorithm thus giving dense outputs. [95] is obtaining a dense depth image from a

structure light 3d cameras like Microsoft Kinect [2] as the depth map generated from such sensors have a lot of missing pixels as you can see in the figure below.



(a) (b)

(c)

Figure 3.2: Depth completion example from [95] (a) is the RGB image of the scene (b)raw depth from Kinect and (c)completed depth with the help of RGB image.

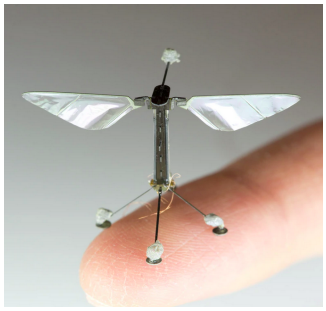
3.2.2 Cost Effective Solutions and Hardware Flexibility

Along with enhancing the depth sensors measurements single view depth estimation is also providing a way to reduce the setup cost of the sensors in applications like autonomous driving. Generally, the sensors used on 3d reconstruction are costly and have their own limitations as mentioned above, these days companies working on autonomous driving or 3D mapping or SLAM (Simultaneous Localisation and Mapping) are using cheaper variants of these sensors. Like instead of using a 64

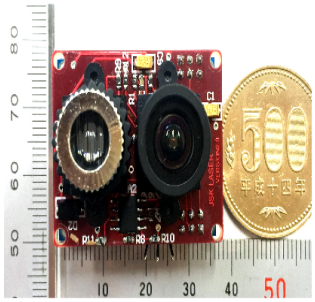
[2]https://en.wikipedia.org/wiki/Kinect

scan line Lidar, they generally use a 16 line scan lidar and use single view depth estimation algorithms to enhance the depth image thus reducing the setup cost and computational challenges of the data processing pipelines to carry out such applications.

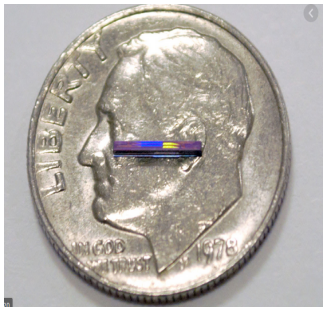
Miniature robotics has been one of the recent interest for researchers and systems like [87], [53] is a good example of that. Swarm robotics or multi-robot systems are employed for applications like disaster relief, 3d mapping of unknown places and more. The system employed for these applications have onboard sensing as the researchers try to make the systems more compact, so does the size of sensors decreases [7], [69], [1]. And the newly developed sensors for the same has not much computational resources and power in comparison to the real version of those used for depth sensing. The depth measurements from these sensors are incomplete and sparse and not sufficient to carry out applications like 3D reconstruction and mapping and therefore single view depth estimation techniques play a major role in providing dense depth measurements.



(a)



(b)



(c)

Figure 3.3: (a) An example of miniature robotics platform called Robobees [87] (b) An ultra tiny laser range sensor[7] (c) Lidar on a chip [69]

3.3 Depth Estimation Methods

Depth estimation or depth reconstruction problems as mentioned above has been an important talk these days among the researchers and significant amount of work has been carried out in this particular branch of research and the existing work may vary according to the type of sensor modality used, the application conditions, whether we need to preserve true depth or we just want an accurate and robust result, kind of input depth the algorithms are dealing with. On the basis of all this literature in-depth estimation is divided into some categories like depth completion and image inpainting, depth super-resolution, depth prediction, RGB image inpainting also has similarities with depth estimation techniques. Later more detailed discussion is carried out on these categories and the existing work which is done in the respective fields.

3.3.1 Depth Completion and Image Inpainting

Depth Completion [86], [85] and depth image inpainting [3] deals with the problem of completing the incomplete depth information in a depth image. Both of these terms are mixed up quite easily but fundamentally there is a difference in both of them on the basis of the input depth these approaches work on and nature of truthness of the real depth after filling the incomplete information. And further, these approaches are classified into guided [78], [61], [13], [91] and non-guided [85] ones, where in guided approaches external sensor modalities are used to complete the incomplete depth and where are in non-guided ones no external sensory information captured from the scene is used.

Broadly we categorize both depth completion and depth image inpainting ?on the basis of density of the input these approaches work on. 3d cameras or structured light cameras like (Microsoft Kinect) produced dense depth images with some irregular holes in it, the depth completion, in this case, can be referred to as depth image inpainting [3] and in the case when the noise of the sensors are considered it is called as depth enhancement [78], [84], [59]. This task of completing the depth information is considered to be easy as more than 80% of the pixel in the depth image has values. And because of that, there exist some filtering based techniques [84], [13] that are able to give good results in the same.

Now talking about the case when sparse inputs are available, such kind of problems are really challenging as near about 5% of the pixels have depth value, in that case, filtering based techniques does not work properly. Works like this [57], [29] where dense disparity map is estimated from sparse features correspondences in stereo matching.[29] is able to provide dense depth even using 5% of the entire depth image by representing the depth image into the wavelet domain. They have represented the problem of dense reconstruction in the form of the optimization problem and proposed a conjugate sub-gradient method for the same. Whereas [57] shows that combining wavelets and contourlets to represent sparse disparity maps gives better accuracy for dense reconstruction. Similarly, other mainstream sensors for indoor applications like Time-of-flight 3d cameras and structured light cameras a significant amount of work is done to recover depth from these sensors. As TOF cameras give a depth map with lower resolution than a color image and it is also prone to noise leads to sparse depth. Earlier work by Diebel and Thrun [12], proposed the use of Markov random fields for the correlation in the range measurements and solved Markov random field optimization problem by conjugate gradient algorithm and combines low-resolution range images with high-resolution camera images, but the resulting dense depth results were over smooth. [26] considered the amplitude values captured by the TOF camera and formed an additional MRF to counter the over smoothness problem. Advanced filters like joint bilateral filter [46] and non-local means filters are also used for the depth completion problems for TOF cameras [46], [36]. Generally filtering based methods for the depth completion and upsampling might have better accuracy than MRF based approaches but they lack in terms of computational competence.

Depth completion for sparse depth samples for Lidar is also a current research topic these days because of the use of the Lidars in outdoor conditions for applications like Autonomous driving. Again the approaches used are classified into guided and non-guided on the basis of external sensory input used to interpolate or complete the missing depth information. This class of depth completion problem is really difficult because of the lack of initial depth information as input is very sparse. As in case, for KITTI depth completion dataset [85] depth images are generated by projecting the Velodyne lidar onto camera image space and it can be seen that less than 5% of all pixels contains depth information and to fill that is a challenging task. [47]

proposed a fast classical image processing algorithm that outperforms even the deep learning-based solutions [85] at the time of submission and performed really well in the KITTI depth completion benchmark [85] and runs on CPU. Uhrig et al. [85] they first proposed a novel sparsity invariant CNN's that are able to handle sparse inputs by introducing input normalizations to the standard convolutional neural network that can work on sparse input data giving deep learning approaches a new pathway to explore depth completion. Inspired by this [17] introduced algebraically constrained normalized convolution layer for confidence propagation and reduce the number of parameters to learn than sparsity invariant CNN's. [9] used compressing sensing techniques Alternating Direction Neural Networks for depth completion. [6] proposed a multi-modal auto-encoder that takes input as three different sensor modalities RGB images, depth images, semantic labels to complete sparse depth inputs. Depth estimation using deep learning is responsible for per-pixel depth prediction, Therefore most of the recent works focused on deep learning are using state of the art deep learning architectures used in object detection and semantic segmentation [66], [54], [68], [72] just by altering a little structure and changing the last layers a bit and mostly the multiscale encoder-decoder architectures are used for depth completion as using encoder-decoder style architectures we are increasing the effective receptive field. As encoder-decoder architectures use transposed convolutions and subsampling [15] and according to [60] introducing such things increases the effective receptive field in CNN's. [35] introduced three operations on sparsity invariant CNN's so that they are able to work with encoder-decoder structures and using multiscale per pixel prediction learning architectures [66], [54], [68]. Works like [70], [39], [21] also used encoder-decoder architectures depth completion. Most of the recent works in sparse depth completion uses benchmarking datasets like KITTI depth completion [85] or NYUv2 [65] to learn their models, but [62] argues that the ground truth provided by these datasets is biased and proposed a self-supervised training framework, that only need sequences of color and depth images for depth completion no neimages for depth completion no need of dense depth labels for learning.

3.3.2 Depth Super-resolution

Depth super-resolution [33], [89], [40], [90], [77], [44] is very similar to the task of depth completion and depth image inpainting. It can be seen as the special case of depth completion problem where we are not dealing with sparse depth inputs, in this case, we have a low-resolution depth image and we obtain a high-resolution depth image. [37] A multiscale guidance convolution network called (MSG-Net), high-resolution depth images are obtained from low-resolution depth image and a high-resolution intensity image of the same scene that is they are combing a high frequency and low frequency progressively and they have reduced the training time by using a high-frequency domain training method. Uhrig et al. [85] also proposed a novel sparse convolution that can handle sparse depth inputs and upsamples the depth without any RGB guidance and their approach is robust to different levels of sparsity.

3.3.3 Depth Prediction from RGB

Predicting depth from RGB images is a popular research area in the field of computer vision, robotics or autonomous driving. As researchers are trying to predict a robust and accurate depth value from a single monocular image without using any external sensor measurements. Knowledge about the shape of the scene from a single has applications like computer graphics [43], robot manipulation and object grasping [52], human pose estimation [81] and more. The work done in monocular depth estimation can be represented on the basis of before deep learning era and after that. Works in a deep learning era are again classified into supervised methods and unsupervised methods.

Before the deep learning era, most of the work in computer vision was carried using handcrafted features that are used to detect certain kinds of features in the input data. [75] uses multiscale local and global features from the image and trained discriminatively a Markov Random Field (MRF) to infer depth. Further work exists that is based on non-parametric methods [42], [41], [58], to infer depth.

After the deep learning became prominent wor[16], [51] begin to pop up based on CNN's that are trained on large-scale datasets. Saxena et al. [74] proposed

Make3D which is a patch model in the image is segmented into patches and 3-d location and orientation is predicted for every patch by formulating the problem in the form of Markov Random Field (MRF) and to predict the parameters of patches or plane they used trained a model based on offline datasets of laser scans from the internet. The main limitation of this approach is that they are not able to perfectly model thin structures leading to less realistic outputs. [56] then solves this problem by using a convolutional neural network to learn those features, further [50] used semantic information and introduced in their model to predict depth accurately. Eigen et al. [16] used two-scale Deep Neural Network (DNN) which incorporate RGB images and its depth values and different all the previous approaches they learn the features raw from the image to predict depth without using external information. Soon works like [49], [25] explores the semi-supervised and unsupervised learning-based methods to predict disparity image prediction. Godard et al. [25] proposed an unsupervised method to predict depth exploring epipolar geometry constraints and generate a disparity map on the basis of an image reconstruction loss. [96] proposed an unsupervised learning framework that estimation of depth and ego-motion together using monocular camera images.

Although recent works show decent results in the task of depth prediction from RGB images but the results are not practical and reliable. Tasks like autonomous driving and certain robotics tasks require accuracy and precision, therefore the reliability of the depth image is a major issue and recent works performs well in indoor condition but they do not work up to the mark giving an error of near about 4 meters on outdoor datasets like Make3D [74] and KITTI dataset[22].

3.3.4 RGB Image Inpainting

As color images have different characteristics than depth images therefore the problem of image inpainting is different from depth inpainting or depth completion. The existing work on depth image inpainting does not focus on the true depth of the scene they just work on the filling the incomplete information from the range sensors and color image inpainting focus on textures of the objects in an image and text and structure are two different entities, therefore, the standard color inpainting algorithms will not work for depth estimation tasks. Some of the work that exists for RGB

based inputs like deep learning-based approaches[93], total variation minimization [4], [45] and diffusion-based [11] techniques.

3.4 State of the Art Methods for Single View Depth Completion

Reference	Source code	Framework used	Remarks (RMSE based on KITTI depth completion benchmark)
[82] "Learning Guided Convolutional Network for Depth Completion". Jie Tang, Fei-Peng Tian, Wei Feng, Jian Li, Ping Tan, IEEE TRANSACTIONS ON IMAGE PROCESSING, 2019. arXiv:1908.01238	https://github.com/kakaxi314/GuideNet		* RMSE - 736.24 * Inspired from image-guided filtering [30] * Employed encoder-decoder architecture for increasing the effective receptive field with the proposed guided convolution layer. * Model compression techniques like Mobile Net[34] is used for reducing the capacity for GPU. * Provides the generalization capability in different lighting and weather conditions , and point densities. * Applicable in indoor and outdoor scenes, evaluation on KITTI depth completion benchmark [85], Virtual KITTI [19], NYUv2 dataset [65],
[70] "DeepLiDAR: Deep Surface Normal Guided Depth Prediction for Outdoor Scene From Sparse LiDAR Data and Single Color Image" Jiaxing Qiu, Zhaopeng Cui, Yinda Zhang, Xing Zhang, Shuaicheng Liu, Bing Zeng, Marc Pollefeys; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3313-3322	https://github.com/JiaxiongQ/DeepLiDAR	Python, Pytorch	* RMSE - 758.38 * Surface normals are used to create an intermediate representation for dense depth prediction * Surface normal pathway contributes towards the nearby objects, color pathway (RGB) handles the depth of faraway objects. * Confidence mask is introduced for the problems of occlusion. * Attention-based integration is carried out to rule out the errors in-depth because of surface normals. * Works well for indoor and outdoor * Approach is robust against different levels of sparsity. * Evaluation carried out on KITTI depth completion benchmark. [85]

Reference	Source code	Framework used	Remarks (RMSE based on KITTI depth completion benchmark)
[21] "Sparse and Noisy LiDAR Completion with RGB Guidance and Uncertainty, Wouter Van Gansbeke", Davy Neven, Bert De Brabandere, Luc Van Gool (Submitted on 14 Feb 2019)arXiv:1902.05356[cs.CV]	https://github.com/wvangansbeke/Sparse-Depth-Completion	Python, Pytorch	* RMSE - 802 * Uses encoder-decoder architecture. * Used two branches for fusion -local and global * Global branch acts as a prior to correct the local branch. * Local branch, lidar input is fused with the guidance map from the global branch. * Data is upsampled on the local branch. * Meets real-time requirements. * An evaluation carried out on the KITTI depth completion benchmark. [85]
[62] "Self-supervised Sparse-to-Dense: Self-supervised Depth Completion from LiDAR and Monocular Camera" by Fangchang Ma, Guilherme Venturelli Cavalheiro,and Sertac Karaman,ICRA,2018	https://github.com/fangchangma/self-supervised-depth-completion	Python, Pytorch	* RMSE - 814.73 * Criticizes the benchmark datasets as it is semi-dense or sparse. * Does not train on those semi-dense annotations. * RMSE error might be more, but the quality of depth maps is much better. * Proposed a self-supervised framework for depth completion. * Evaluation proves that approach handles the highly sparse data well. *Robust towards different levels of sparsity. *An evaluation carried out on the KITTI depth completion benchmark. [85]

Reference	Source code	Framework used	Remarks (RMSE based on KITTI depth completion benchmark)
[17] "Confidence Propagation through CNN's for Guided Sparse Depth Regression", Eldesokey, Abdelrahman and Felsberg,Michael and Khan, Fahad Shahbaz, arXiv preprint arXiv:1811.01791,2018	https://github.com /abdo-eldesokey/ nconv	Python, Pytorch	* RMSE - 908.76 * Proposes a normalized convolution layer that requires less number of parameters to learn that the previously proposed methods. * Novel strategies via which can propagate the confidence information to next layers. * Proposed approach learns only 5% of the parameters as comparison to other SOA. *Less runtime and good for embedded devices.
[39] "Sparse and Dense Data with CNNs: Depth Completion and Semantic Segmentation", Maximilian Jaritz, Raoul de Charette, Emilie Wirbel, Xavier Perrotton, Fawzi Nashashibi,31 Aug 2018, arXiv:1808.00769			* RMSE - 917.64 * This approach negates the idea of using the validity masks for depth completion. * They propose it is not necessary to use validity masks. * Uses NASnet [97] inspired encoder-decoder architecture for increasing the effective receptive field. * Robust towards different levels of sparse depth data. * Evaluation data sets Synthia [73], KITTI depth completion benchmark [85], Cityscapes [10]. * The approach works well with low-cost Lidars.

Reference	Source code	Framework used	Remarks (RMSE based on KITTI depth completion benchmark)
[35] "HMS-Net: Hierarchical Multi-scale Sparsity-invariant Network for Sparse Depth Completion" Zixuan Huang, Junming Fan, Shuai Yi, XiaogangWang, Hongsheng Li, 27 Aug 2018, arXiv:1808.08685			* RMSE - 937.48 * This approach able the sparsity invariant convolutions. * This approach proposed three operations by which sparsity invariant CNN's are able to work with encoder-decoder structures. * Network architecture inspired from SOA object detection architectures that uses multi-scale feature combination. Like FPN's [54], U-net [72], FRRN [68]. * Robustness of the approach is tested inducing different sensors noises. * Robust to different levels of sparsity. * Evaluated on KITTI depth completion benchmark [85] and NYU-depth-v2 dataset [65].
[96] "DFineNet: Ego-Motion Estimation and Depth Refinement from Sparse, Noisy Depth Input with RGB Guidance", Zhang, Yilun, and Nguyen, Ty and Miller, Ian D and Shivakumar, Shreyas and Chen, Steven, and Taylor, Camillo J and Kumar, Vijay, and others,arXiv:1903.06397	https://github.com/ShreyasSkandanS/DFuseNet		* RMSE - 943.89 * Criticizes the state of the art benchmarks for evaluations as the ground truth is semi-dense, leading to biased results. * Utilizes pose prediction CNN architecture and depth prediction CNN architecture for dense depth prediction. * Applicable for different type of noisy sensors * Evaluation on KITTI depth completion benchmark [85].

Reference	Source code	Framework used	Remarks (RMSE based on KITTI depth completion benchmark)
[] J. Ku, A. Harakeh and S. L. Waslander, "In Defense of Classical Image Processing: Fast Depth Completion on the CPU," 2018 15th Conference on Computer and Robot Vision (CRV), Toronto, ON, 2018, pp. 16-22.	https://github.com/kujason/ip_basic	Opencv, Python	* RMSE - 1288.46 * This is a nonlearning based approach for lidar depth upsampling. * Uses different image processing algorithms for depth completion without any RGB guidance. * Able to run on CPU and in real-time. * Has provided better performances that CNN based architectures like sparsity invariant CNN's. * Evaluation carried out on KITTI depth completion benchmark [85].
[47] J. Uhrig, N. Schneider, L. Schneider, U. Franke,T. Brox and A. Geiger, "Sparsity Invariant CNNs, "2017 International Conference on 3D Vision (3DV), Qingdao,2017, pp. 11-20	https://github.com/PeterTor/sparse_convolut‘ion	Tensor -flow, Python	* RMSE - 1601.33 * Proposed a novel sparsity invariant CNN's that enables the CNN's to tackle the sparse input data. * Proposed their own novel data set for training generating the dense ground truth which is the current KITTI depth completion benchmark [85]. * Approach basically based on upsampling the sparse data without any guidance. * Robust towards different levels of sparsity. * Evaluation of the proposed KITTI depth completion benchmark [85].

Reference	Source code	Framework used	Remarks (RMSE based on KITTI depth completion benchmark)
[8] "Learning Depth with Convolutional Spatial Propagation Network", Xinjing Cheng, Peng Wang, Ruigang Yang, arXiv:1810.02695	https://github.com /XinJCheng/CSPN	Python , Pytorch	* RMSE - 743.69 * One variant of this approach ranks second on KITTI depth completion benchmark [85]. * Proposed a convolutional spatial propagation network which learns an affinity matrix for the guidance image. * This network can be appended to any SOA architectures for increasing their performance. * Architecture is inspired by SPP(spatial pyramid pooling)[] for increasing the effective receptive field. * Evaluated the multiscale spatial pyramids networks[] for dense pixel prediction. * Evaluation is carried out on KITTI depth completion benchmark [85].

3.5 State of the art Bench-marking Datasets

3.5.1 KITTI Depth Completion

KITTI depth completion benchmark [85] is one of the state of the art benchmarking datasets for depth completion. This data set was proposed by [85]. The dataset contains a set of Lidar scans taken in outdoor environments and are projected into image coordinates for obtaining the depth image. These Lidar scans are projected into the image plane using camera calibration matrices which are provided for every scan using perspective transform we talked about in section 2.1.2 and the resolution of the depth image is equivalent to RGB image. Lidar scans are present in only lower regions because of the angle of lidar scan lines. And the only 5-7% of the pixels contain depth values. Every corresponding image is provided with a depth image with the same name of the scene which can be used for guided depth completion. Further 1000 corresponding images of depth and RGB are given in validation and test set.

In the ground, truth depth image near about 30% of the pixels have depth values and it is created by merging 11 Lidar scans using pose estimates that are given in [85]. The sparse projected images are corrected using stereo depth estimation algorithms and the outliers were removed. As this dataset provides a large number of images to train the models for depth completion but it has some drawbacks as it does not contain depth in upper regions of the depth image and sometimes depth values are missing along the object boundaries leading to blurred results for the approach which are trained on this dataset.

The performance of the algorithms that are trained on this dataset is measured with the help of Root Mean squared Error (RMSE) , Mean Average Error(MAE), inverse Root Mean Square Error(iRMSE) and inverse Mean Average Error (iMAE). RMSE is considered as the prominent error for the ranking of the best algorithm on the benchmark. A brief description of the error metrics defined above.

RMSE (Root Mean Square Error)- it takes into consideration the depth errors for faraway distances and is measured in mm.

$$RMSE = (\frac{1}{|V|} \sum_{u,v \in V} |o(u,v) - t(u,v)|^2)^{0.5}$$

MAE (Mean Average Error) - takes into consideration the depth errors for farther away objects

$$MAE = (\frac{1}{|V|}\sum_{u,v \in V} |o(u,v) - t(u,v)|)$$

iRMSE (inverse Root Mean Square Error) takes into consideration the depth errors for nearby objects.

$$iRMSE = (\frac{1}{|V|}\sum_{u,v \in V} |\frac{1}{o(u,v)} - \frac{1}{t(u,v)}|^2)^{0.5}$$

iMAE(inverse Mean Absolute Error) takes into consideration the depth errors for nearby objects.

$$iMAE = \frac{1}{|V|}\sum_{u,v \in V} |\frac{1}{o(u,v)} - \frac{1}{t(u,v)}|$$

In all of the metrics above o represents the output pixel from the approach and t stands for the ground truth pixel value of the same pixel.

3.5.2 Virtual KITTI

"Virtual KITTI [19] is a synthetic video dataset designed to learn and evaluate computer vision models for several video understanding tasks: object detection, multi-object tracking, depth estimation, optical flow and more." [3] The dataset contains around 21260 images and depth frames that are synthetically generated with simulated lighting and weather conditions and the maximum depth range provided by the data are around 655m. This dataset is not a real representation of the real-world data, but this provides a large training corpus for the training and it can be seen that approaches like [80] have used this dataset to show that their approach is robust to different scenarios and sparsity levels.

[3] https://europe.naverlabs.com/research/computer-vision/proxy-virtual-worlds/

3.5.3 NYUv2 Dataset

NYUv2 [65] is an indoor dataset that is recorded in homes and offices and a Kinect RGBD sensor is used for collecting the dataset. It contains 1449 aligned and labeled pairs of RGB and dense depth images. Class labels are given for each object in the scene. It also contains the raw RGB and Kinect dataset for the respective scenes before preprocessing. This dataset has been used by the approaches to show that their depth completion algorithms can work in different conditions and they are robust towards the level of the sparsity of the input data from different range sensors.

3.5.4 Synthia Dataset

Synthia dataset [73] is built on a unity game engine and provides an RGB, semantics, and depth of urban cities and highways, with different scenarios of pedestrians and cars. It is a large dataset that contains more than 200,000 high resolution images from video streams. It provides synthetic real-life scenarios with different lighting conditions and seasons like summer, winter, spring. The ground truth is available for semantic segmentation, depth, and car ego-motion.

3.5.5 Cityscapes Dataset

Cityscapes dataset [10] is a large scale dataset that provides stereo disparity from 50 different German cities, it includes 20000 weakly annotated and 3000 fine pixel-level semantic annotations. The dataset is taken in different weather conditions in urban scenarios in the daytime in different seasons of the year like spring, summer, winter.

4

Methodology

This section contains the description of the task of the depth completion that has been employed on a test rig composed of an Intel Realsense 3d camera[1] and a low-cost solid-state lidar Hypersen[2] and a color camera setup. Later explains the collection of dataset through the designed rig and the state of the art approaches that are selected to be evaluated from the section 3.4.

4.1 Design of Test Rig

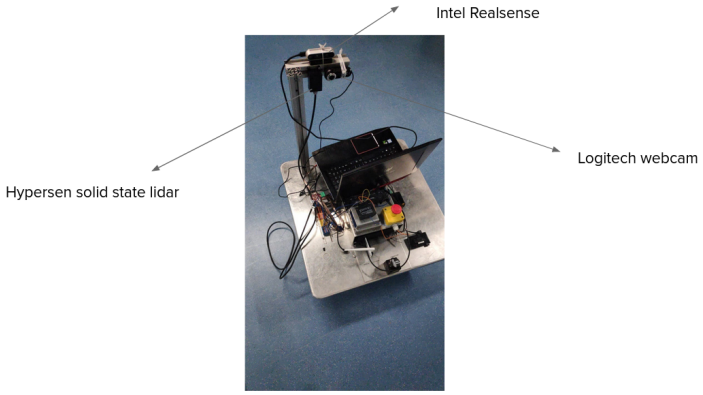


Figure 4.1: Test rig setup composed of a low cost solid state Lidar, a color camera, and a 3D camera

[1]https://www.intelrealsense.com/depth-camera-d435/
[2]https://en.hypersen.com/product/detail/10.html

The setup of test rig is mounted upon the mobile robot platform with the help of metallic mounts and fixed at a place with screws. The mobile robot platform has its own Linux operating system has ROS is installed on it. The sensors are controlled by the ROS framework and the data from the sensors are collected in the form of ROS bag files.

4.2 Sensors Used

(a)

(b)

(c)

Figure 4.2: (a) Intel Realsense D435 3d camera (b) Logitech webcam carl zeiss tessar (c) Hypersense Solid-state Lidar

Table 4.1: Specification of Intel Realsense d435 3d camera

RGB Resolution	1920 x 1080
Frame Rate	30fps
Field of View	69.4^o x 42.5^o x 77^o
Communication Interface	USB-C* 3.1 Gen 1

Table 4.2: Specification of Logitech webcam carl zeiss tessar

RGB Resolution	1920 x 1080
Frame Rate	30fps
Communication Interface	USB

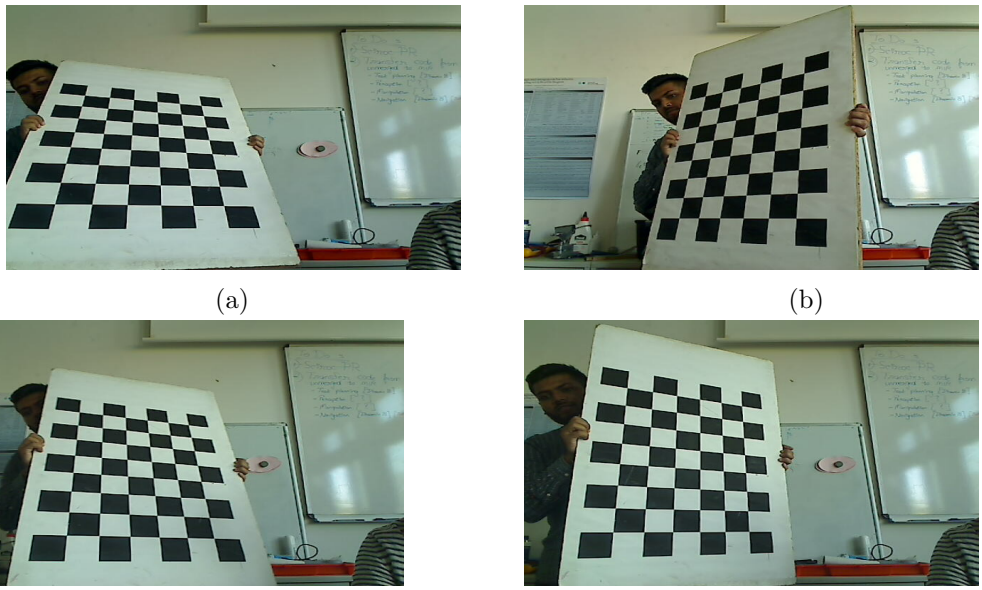
Table 4.3: Specification of Hypersense Solid-state Lidar

Spatial Resolution	60 x 60
Frame Rate	35fps
Field of View	76^o x 32^o
Communication Interface	USB, LAN, RS232
Power Supply	12-24V

4.3 Camera Calibration

In the test rig we have a camera and lidar setup. The camera is calibrated with the help of Opencv python camera calibration tutorials [3] where a set of pictures of a checkerboard pattern is taken from different perspectives and using perspective transformation as mentioned in section 2.1.2 the camera parameters are calculated and placed in an XML file format. The camera parameters consists of intrinsic and extrinsic parameters. We have assumed the point where the camera is mounted to be the origin of the world coordinate frame. The XML file is then placed into the ROS camera driver which further corrects the distortion and other deformities in the RGB data collected.

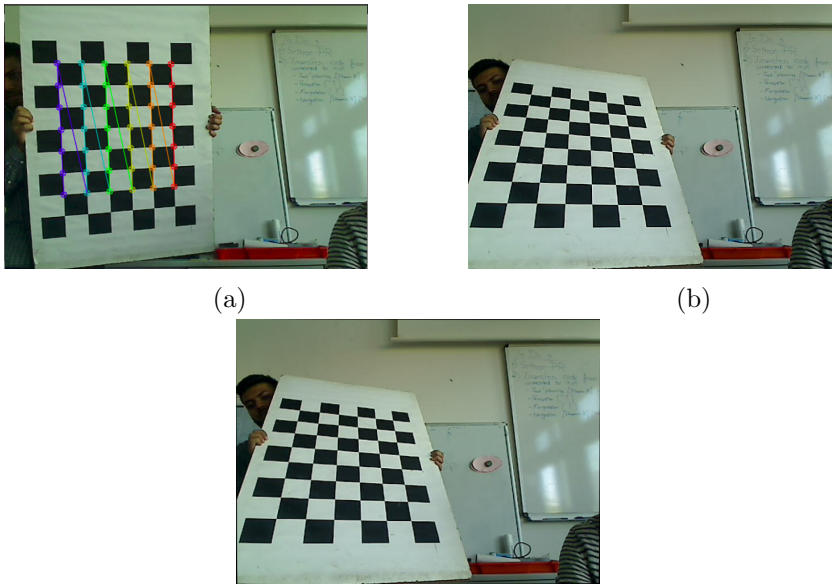
[3]https://opencv-python-tutroals.readthedocs.io/en/latest/py$_t$*utorials*/*py*$_c$*alib*3*d*/*py*$_c$*alibration*/*py*$_c$*alibration*



(a) (b)

(c) (d)

Figure 4.3: (a), (b), (c), (d) pictures of checkerboard pattern with the non calibrated camera.



(a) (b)

(c)

Figure 4.4: (a)Detection of corners of the checkerboard pattern, (b) not calibrated image, (c) calibrated image

4.4 Dataset Collection

To evaluate some of the state of the art depth completion approaches from section 3.4, two indoor datasets were taken with the help of the test rig.

4.4.1 Intel Realsense Dataset

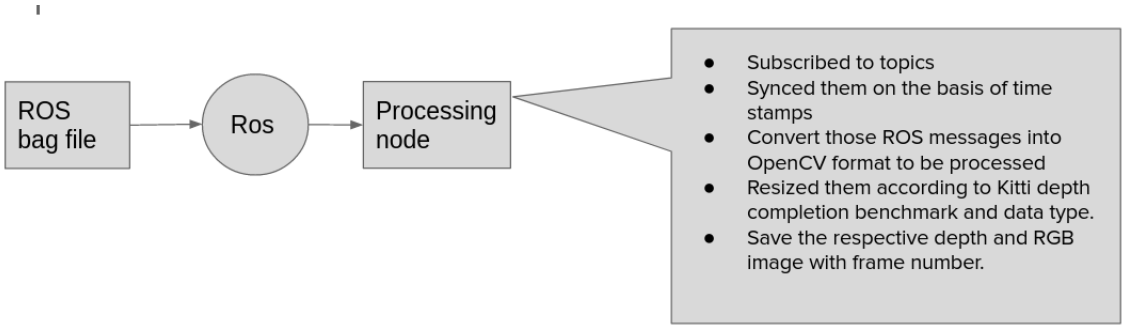


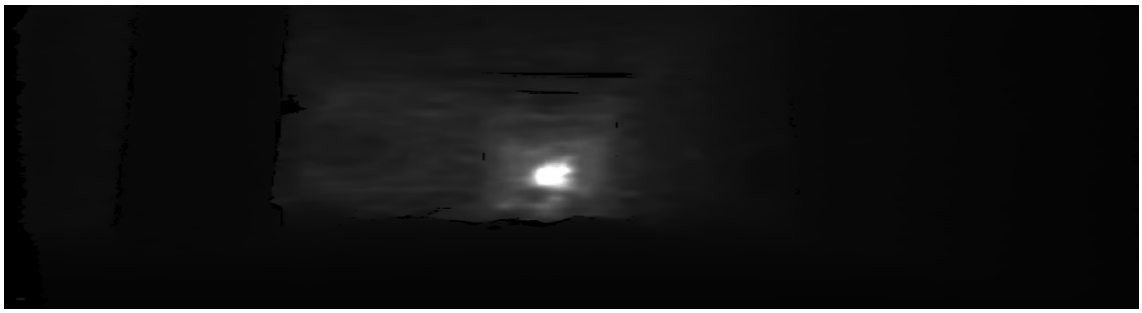
Figure 4.5: Pipeline for realsense data collection

ROS[4] driver for the respective sensor was installed on the robot platform and the driver is executed. The robot is teleoperated with the joypad and moved around and the data is recorded in a ROS bag file. To make the data set ready for evaluation some preprocessing steps were performed on it. A preprocessing ROS node is written in python which is executed and the bag files is played in the ROS master. Preprocessing node subscribe to the topics that contain data for aligned RGB and depth image which are already synced as done already by the sensor manufacturer. Further, the ROS messages are converted into OpenCV format so that it can be saved. But before saving, RGB and depth images are resized in a similar fashion with the KITTI depth completion benchmark[85] as the evaluated approaches selected were trained on that. After resizing the images they are given a similar frame number for every matched timestamp and saved.

As to evaluate the dataset we need a ground truth depth image. As done by the creators of KITTI depth completion dataset[85] in which they create their ground truth by using stereo matching techniques and removing the outliers as originally

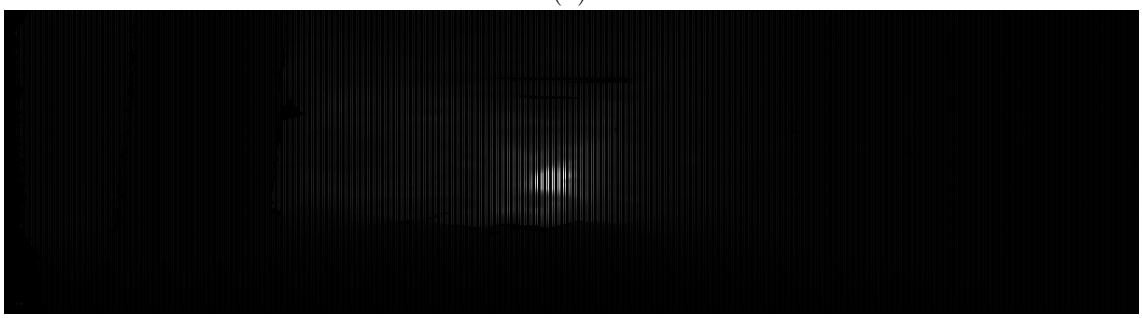
[4]https://www.ros.org/

the depth images have less than 5% of the pixels have values and after creating their ground truth near about 30% of the pixels have depth values in-depth image. Inspired from them we created our ground truth where we artificially make the depth images acquired from the 3d camera sparse by removing the values of some pixels randomly and trying to fill those missing information. The dataset contains 1560 depth and RGB images and 1560 synthetically generated sparse depth images.



(a)

(b)



(c)

Figure 4.6: (a) is the depth image acquired from the 3d camera which will act as ground truth (b) is the RGB image from the 3d camera (c) is the synthetically made sparse.

4.4.2 Lidar and Camera Dataset

As the main purpose of this project is to test low cost lidar and camera setup for the task of depth completion. Therefore to test the performance of state of the art depth completion approaches we took an indoor test dataset with the low cost lidar and a color camera setup that is mounted on the test rig.

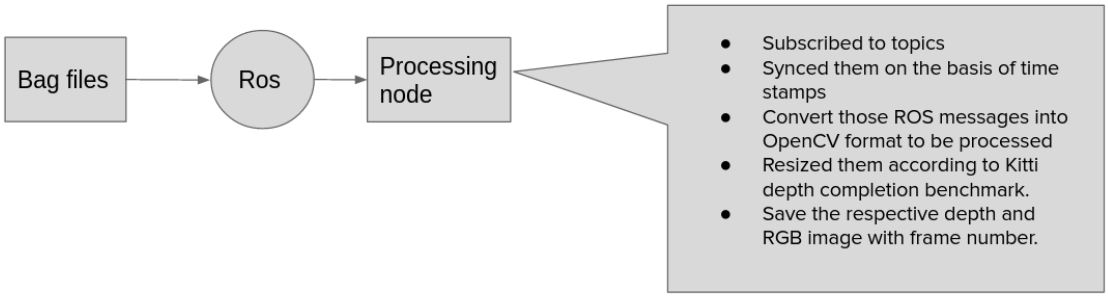


Figure 4.7: Pipeline for realsense data collection

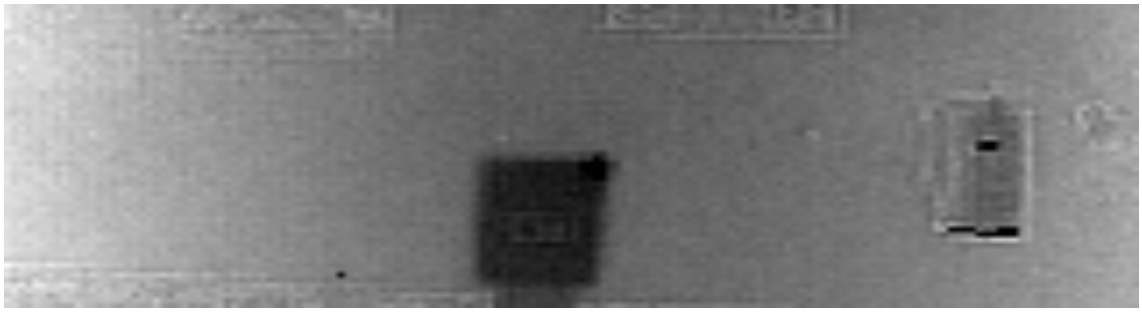
The drivers are executed in the ROS framework which turns on the camera and the lidar and both of the sensors start publishing the data in the ROS master and the robot is moved with teleoperation in indoors and the data is recorded in the form of bag files. To make a dataset ready for evaluation we need to preprocess it as the frame rate of both the sensors are different so we need to sync them we use a ROS framework to sync the data from both two sensors that when the publishing time that is timestamp in our case is very near to each other like a difference of 1 millisecond we record that data as one frame. The data in the selected frame is still in the form of ROS messages therefore we have to convert the messages into a format that we can process. We convert those ROS messages into OpenCV format and resize them to the resolution similar to Kitti depth completion benchmark [85] that is (352 x 1216) and after the same frame is saved with the frame number.

The data set consists of 232 depth and RGB images with the same frame number with the spatial resolution of 1216 width and height are 352.

There is a need for extrinsic calibration of the lidar depth images to align the depth images with the camera images as there is an offset between the camera and the lidar. As the position where the camera is mounted is defined as the origin

of the world coordinate system, and the lidar is translated in the x-direction from the origin. Therefore using the perspective transform and filling the values of the extrinsic calibration parameters as explained in section 2.1.4 which is nothing just composed of rotations and translations the depth images are aligned with the RGB images.

(a)



(b)

(c)

Figure 4.8: (a) RGB image from the camera (b) is the depth image from our low cost Lidar (c) aligned RGB image with depth.

The results for aligned RGB images with depth images from lidar is average because we don't have the exact knowledge of the center of the mounting point of

the camera.

4.5 Evaluated Methods for Single View Depth Estimation

The approaches that have been used to evaluate the task of depth completion on the collected data set are [47], [70], [62], [21].

4.5.1 Approach 1

[47] is one of the nonlearning approaches that give the outperforms the learning-based techniques like [sparsity invariant CNN's] and able to run in real-time and on CPU. This approach is non-guided as it uses no RGB images just relies on the sparse LIDAR data for depth completion using classical image processing approaches and runs on CPU and it ranked first on the KITTI depth completion benchmark [85] at the time of submission. We were interested in the methodology followed by the authors for the task of depth completion and using only basic classical image processing techniques to solve the task.

In this authors have used the approach of filling the incomplete depth maps using 8 steps of classical image processing which are carried out using OpenCV. Different scales of dilations are carried, followed by Gaussian and Bilateral blur. Their approach revolves around the concept of using the larger pixel values to fill the lower pixel values. First, the values of the pixels are inverted by an offset such that when the dilation is carried out the lower pixel values will not be overlapped by larger pixel values that is larger depth values will not overlap lower depth values and if it happens that will affect the boundary of the objects in closer ranges. In the next step, the first dilation is carried out which is based on the idea that the nearby pixels to valid pixels are upsampled at first and those pixels are most legitimate to have the same depth as them, for that they have decided a kernel which is designed on the basis, in keeping with the structure of lidar scan lines and the sparsity level and the values for the kernel is selected so that, pixels with the similar values are dilated to the same value. They have defined some shapes and sizes and in the end, they went for 5x5, diamond binary kernel. After the first dilation, there exist some regions where depth has not been upsampled and again a 5x5 binary kernel is used to pinpoint the depth and to fill the remaining holes that are left there. They build

upon the idea that nearby patches of the depth can be joined to create the boundaries for the objects. After this big to medium holes are filled, and to fill the small holes a 7x7 kernel dilation is used such that filling of only the empty pixels takes place and leaving already filled ones. As the method was employed on the KITTI depth completion benchmark[85], and in the benchmark, there exists no point clouds for tall objects, and to fill for those they extrapolated the top value of each column in the depth image to the top pixel. The remaining steps takes into account the larger holes that are not filled in the prior dilations as these areas might not contain any point cloud information, therefore, no depth values in the depth image so to fill those they used a 31x31 dilation kernel leaving again the remaining pixels unchanged. After filling the holes they remove the outliers by the process of denoising and used Gaussian and bilateral blur one by one and checked which ones give them the least RMSE value on the ground truth. And in the end, inverted depth values are shifte back to their original scale by removing the offset that was done in the first case resulting in the completed depth image in the end.

4.5.2 Approach 2

[70] is one of the learning-based state of the art guided depth completion methods and the performance of this algorithm lies in the first half of the KITTI depth completion benchmark[85]. This approach is different from the other state of the art guided depth completion in such a way that they are using surface normals from the RGB image as a guidance cue for depth completion. They have divided their learning architecture into two pathways surface normal pathway and color pathway. Both pathways take as an input RGB image of the scene and sparse depth image and a binary mask that gives us the information where pixels contain depth values and not. In the surface normal pathway firstly the surface normals are calculated from the RGB images and then are combined with sparse depth and the confidence mask that has been calculated from the color pathway and gives a dense depth map and in the same, we get dense depth from the color pathway. In the end, an attention-based integration is carried out where an attention map is calculated from the results from both the pathways and calculate a weighted average depth maps based on both the pathway. As calculating depth using surface normals is not accurate in outdoor areas

and noises in surface normals can affect the depth estimation.

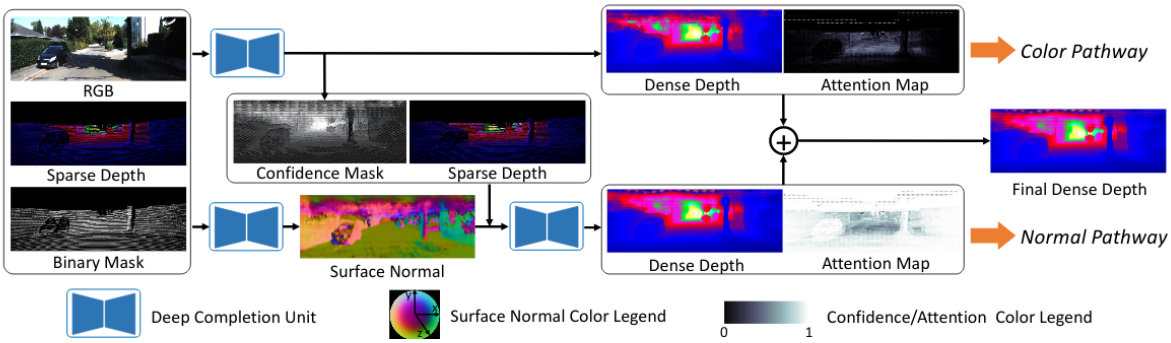


Figure 4.9: Pipeline of [70]

In both the pathways they have used [70]DCU (Deep Completion Units) by taking inspiration from the traditional color image inpainting techniques [83], [94], which presented that dense depth is dependent on sparse depth and surface normals for a particular scene. [70]DCU is composed of encoder-decoder structure with skip connections, the encoder is responsible for learning the local affinity from the provided RGB and sparse depth whereas in the decoder part they fused by means of interpolation, features learned by the two different encoders at different scales. In the architectural point of view, the encoder is composed of Resnet blocks [31] where the decoder is composed of up-projections or we called transpose convolutions. For the attention-based integration propsed by them, they preedict a score map from the last features learned before the output layer and fed through three convolutions via Relu and, the two score maps calculated from the pathways are then passed into a softmax layer, and transformed into combined weight and at the end, the final dense map is this a weighted combination of the depth maps of the two above mentioned pathways. They have introduced confidence masks that is responsible for handling occlusions. These confidence masks are usually learned from color pathway and that helps in obtaining the reliability of the depth input cues at the object boundaries and close ranges. For the training data, authors have generated their own training data because of the lack of training data for the surface normals. They use an open-source urban driving simulator Carla [14], where they synthetically rendered 50K training samples of RGB images, sparse depth images, and dense depth images. For the real-world data, they use surface normal ground truth for the KITTI dataset

[22]. As ablation study is carried out and it shows that the contribution of each and every element that they have introduced in their approach reduces the RMSE on the ground truth and it was concluded in the end that, surface normal is responsible for close-range features and on the other hand the color pathway capture accurate details in long-range. They also conduct an evaluation in which they make the data more sparse in different ratios and it was concluded that this approach generalizes well for different levels of sparsity.

4.5.3 Approach 3

This approach is one of the states of the art methods for guided depth completion.[62] It is similar to the approach described in section 4.5.2 they are also using two different pathways for the fusion of RGB and Lidar data but they are not using surface normals as guidance cues instead of that they are relying only on the feature maps from depth and RGB to upsample and complete depth.

They have proposed two pathways that they named local and global branch. Local branch takes in input as only sparse lidar input and densify the depth image, whereas in global branch takes in input sparse depth image and RGB image.

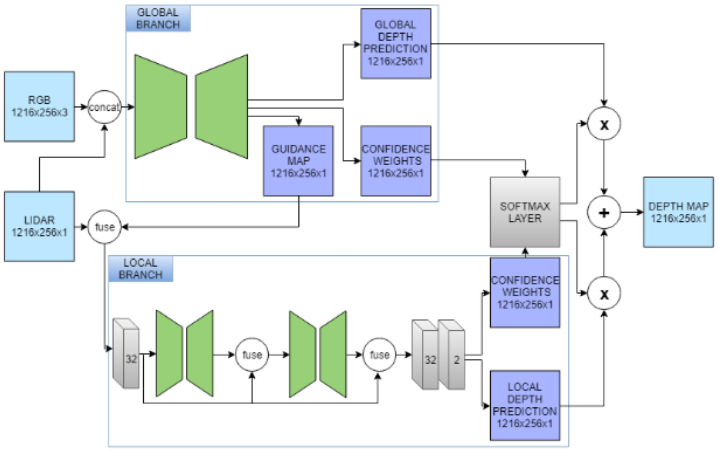


Figure 4.10: Architecture proposed by [62]

They have proposed architecture on the basis of early and late fusion, of RGB and depth information. The global branch corrects the features predicted by the

local branch and act as regularizer as there are mistakes in the Lidar input and global information in the form of guidance map helps the local branch to find the mistakes in the Lidar input and predict accurately a dense depth image. The global network has been put up to detect the moving objects and it is also able to give a better prediction if two objects in the scene have similar depth. As sparse Lidar input is not able to find the borders of the objects, therefore from global network RGB information can be utilized to detect the borders of the object easily and this information is given to the local network in the form of global guidance map which is obtained from global branch as it tell the local network about the confident and accurate Lidar points as global network is able to extract features about objects edges, boundaries in the scene.

In a similar fashion to [70], they are combining the outputs from both the branches using an uncertainty. As both branches predict a confidence map and it used as a weight map for the two branches to be fuse together. Basically using this method we are giving attention to certain type of input towards the prediction of a dense depth map like in some cases the model learns global information in some regions than the local information there as mentioned above the confidence will be more where accurate Lidar measurements are available and local network which is taking sparse Lidar data as input will complete the depth image with more confidence and global information in local network will be used where there are no accurate Lidar points are available. The architecture of global branch is an encoder-decoder network that is inspired by [71] and a local network is made up of a stacked hourglass network. The loss functions they have used are inspired by [55], [51]. Evaluation of this approach is carried out on the KITTI depth completion benchmark [85] and lies in the top part of the KITTI depth completion leaderboard and runtime of this approach is 20ms, therefore it qualifies real-time requirements.

4.5.4 Approach 4

Existing work on the depth completion is carried out using the densely annotated ground truth which in the case is not densely annotated as even the acquisition of semi-dense annotations is challenging. The existing ground truth available at the KITTI depth completion benchmark ie. given by the paper sparsity invariant CNN's

[85] is also semi-dense. Where they create the, annotated ground truth using stereo vision algorithms, GPS and additional manual inspections. This approach [62] is special because it not relying on those semi-dense annotations, which can lead to not good results on validation set of the KITTI depth completion benchmark [85]. But the resulting dense depth maps by this approach is very accurate but might have lower RMSE values than the other existing works on guided depth completion.

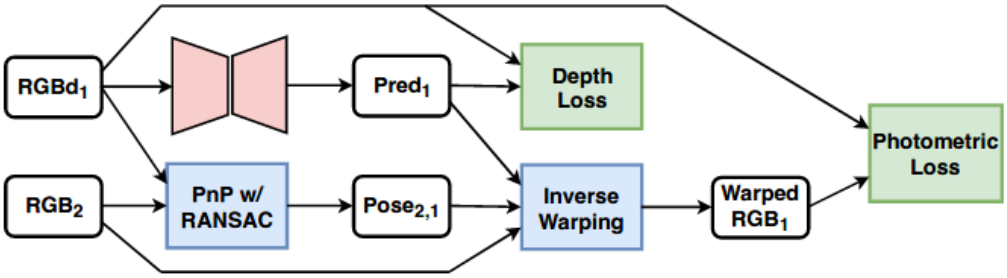


Figure 4.11: Proposed unsupervised learning framework for depth completion by [62].

The authors have proposed a model-based self-supervised framework in which they are using the synchronizes sequence of monocular images and depth data from Lidar. Their architecture is basically divided into two streams in which they are taking the current frame of the RGBD data and an encoder-decoder network is implemented basically consists of Resents blocks [31] with skip connections which at the end of the decoder predicts the depth maps. In the second stream, they taking the consecutive frames of RGB image and perspective-n-point algorithm is employed to capture the pose of the camera using the 3d points and 2d projections of those 3d points that are the RGB images. Further they have used RANSAC to remove the outliers so that estimation can be more robust. Then the predicted pose from frame 1 to 2 and the RGB image from the second frame with the predicted depth from the current frame is used to produce a wrap image using inverse warping which is further used to calculate the photometric loss. Thus to conclude about the architecture we can say that current frame data from the depth camera and the RGB image from the consecutive frame is used to provide self-supervised signals.

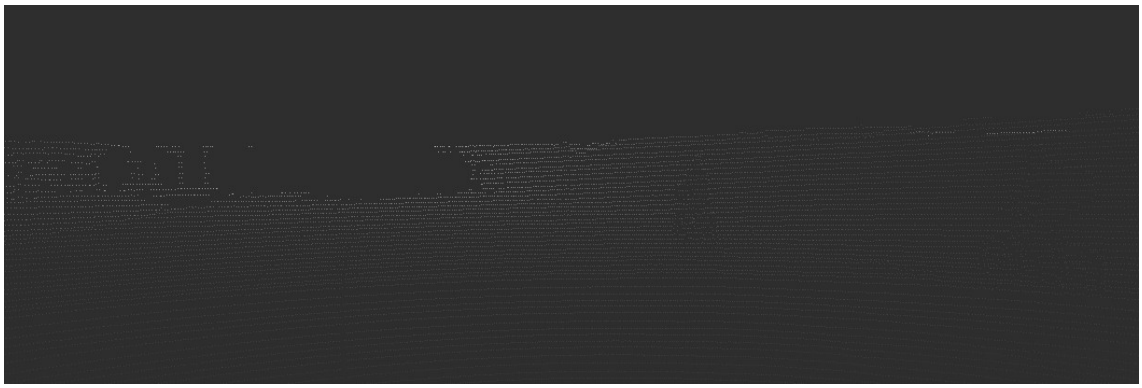
Talking about the loss functions that they have used to govern their training of the model, first loss function used in the first stream while prediction depth in which

they are not using any ground truth as described above, therefore the sparse depth input is used here as supervised signal. The second loss function that they have used is at the combination of both of the streams called photometric loss function which is accounting, for warped image generated from the predicted depth and the image from the previous frame, here the predicted depth is not used in the loss function, therefore for the calculation of this loss function, no pixels associated with depth are used. Reducing the photometric loss results in the reduce of the depth loss, when the predicted depth is similar to the ground truth means that when the warped images are nearly equal to the real one. Takeaway from this approach is that apart from the self-supervised architecture that they have proposed, the quality of dense depth maps from this approach is better than the existing state of the art approaches for depth completion which is considered better in terms of RMSE on the KITTI depth completion benchmark [85].

5

Results

5.1 Evaluation of Results on KITTI Depth Completion Dataset



(a)

(b)

(c)

Figure 5.1: (a), (b) and (c) are sparse depth image from image plane projection of raw velodyne scan, ground truth depth image and the Respective RGB image of the scene derived from [85]

Figure 5.2: Depth completion results on the sparse depth image, figure 5.1 (a) using RGB color image (b) by [70]

Figure 5.3: Depth completion results on the sparse depth image, figure 5.1 (a) using RGB color image (b) by [21]

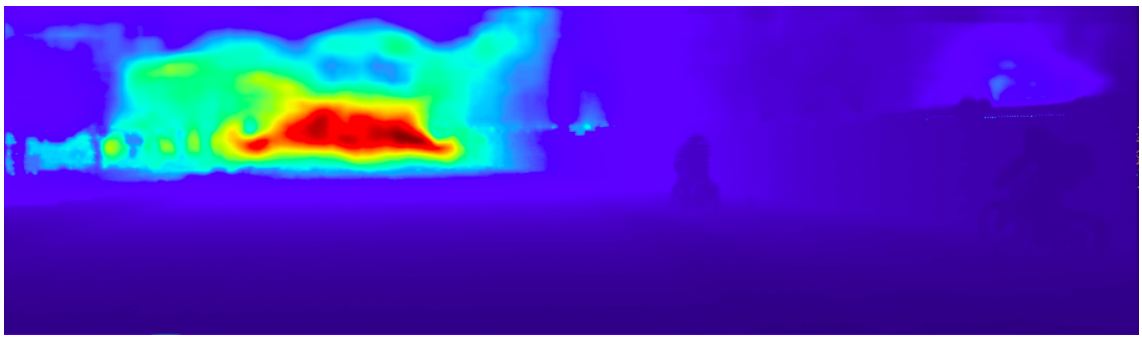


Figure 5.4: Depth completion results on the sparse depth image, figure 5.1 (a) using RGB color image (b) by [62]

Figure 5.5: Depth completion results on the sparse depth image, figure 5.1 (a) using RGB color image (b) by [47]

Table 5.1: Evaluation on KITTI depth completion benchmark [85]

Approach	**RMSE[mm]**	**MAE[mm]**	**iRMSE[1/km]**	**iMAE[1/km]**
[70]	**758.38**	**226.50**	2.56	1.15
[21]	772.87	215.02	**2.19**	**0.93**
[62]	814.73	249.95	2.80	1.21
[47]	1288.46	302.60	3.78	1.29

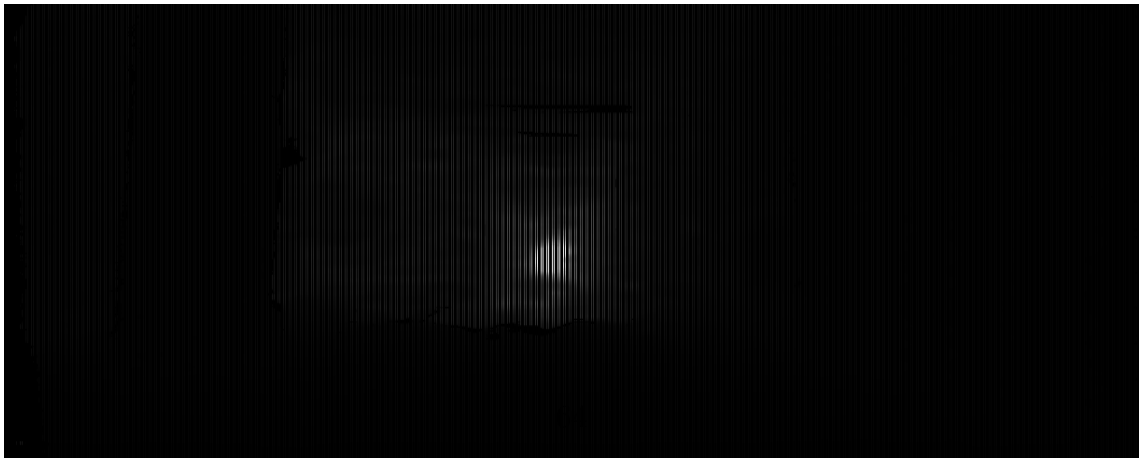
As we can see that [70] has the least RMSE and MAE values on the KITTI depth completion benchmark [85] where as [21] has the least values of iRMSE and iMAE. This means that [70] completes depth more accurately for faraway objects where as [21] completes depth more accurately for the objects lying at less depth. Thus shows that the surface normal criteria followed by [70] works as the noise in the surface normal increase with distance and in figure 4.9 we can see that they have followed two pathways to correct the errors caused by noisy surface normals, and to correct that they have introduced a confidence mask which is obtained by combining sparse depth, RGB and binary mask of that sparse depth. Whereas [21] uses the similar pipeline for fusion but their local branch as in figure 4.10 takes in input early fusion of sprase lidar and guidance map which is predicted from the global branch. Thus this early fusion of guidance map in local branch helps it to concentrate on correct and confident depth values as global branch takes in input RGB and sparse depth, this makes the global branch to learn about the boundaries and shapes of the objects in the scene.

5.2 Evaluation and Results on Collected Dataset

5.2.1 Depth Intel Realsense Dataset

(a)

(b)



(c)

Figure 5.6: (a) the ground truth depth image, (b) synthetically created sparse depth image and(c) is the respective RGB image of the scene all taken by the Intel Realsense

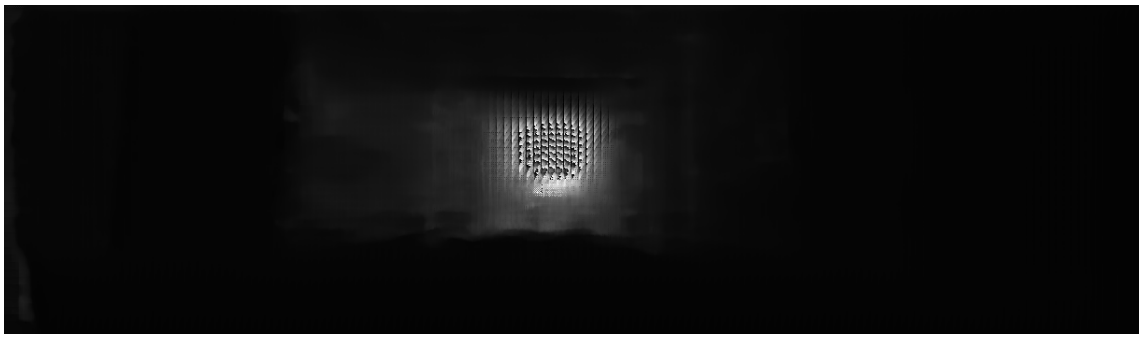


Figure 5.7: Depth completion results on the sparse depth image, figure 5.6 (c) using RGB color image (b) by [70]

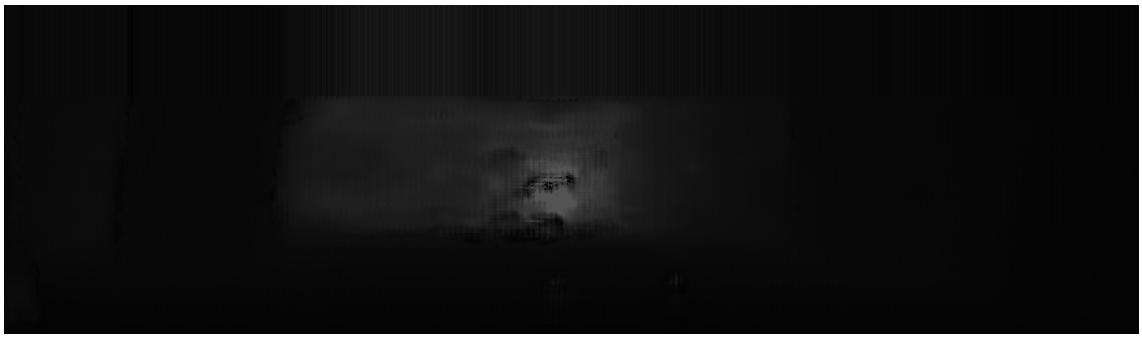


Figure 5.8: Depth completion results on the sparse depth image, figure 5.6 (c) using RGB color image (b) by [21]

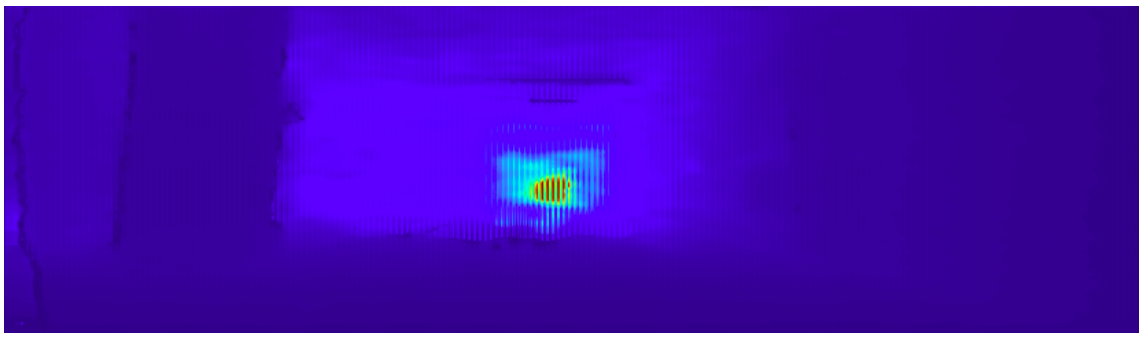


Figure 5.9: Depth completion results on the sparse depth image, figure 5.6 (c) using RGB color image (b) by [62]

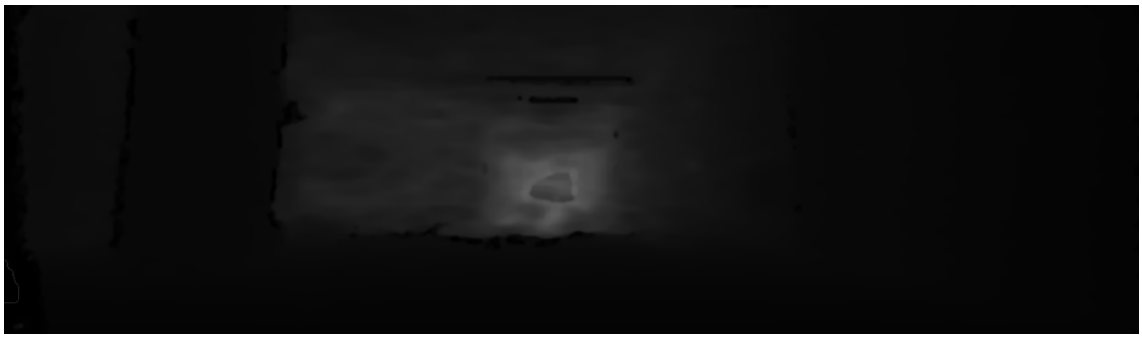


Figure 5.10: Depth completion results on the sparse depth image, figure 5.6 (c) using RGB color image (b) by [47]

Table 5.2: Evaluation on dataset collected with Intel Realsense

Approach	**RMSE[mm]**	**MAE[mm]**	**iRMSE[1/km]**	**iMAE[1/km]**
[70]	3794.48	2035.29	inf	inf
[21]	2894.16	1574.94	inf	inf
[62]	3730.97	1040.89	inf	inf
[47]	**939.20**	**178.25**	inf	inf

As you can see that [47] performs the best on RMSE and MAE as other other approaches we evaluate the dataset on. [47] is a non learning based approach where as all the other approaches are learning base approaches. Talking about the iRMSE and iMAE errors for the all the approaches in infinity as it was observed that sometimes the errors for some frames is not infinity where as for some frames it is infinite. More investigation about this situation will be carried out in the last section.

6

Conclusion

This work presents the task of dept completion for the sparse depth image that is taken from an active range sensor with the help of RGB image from a High-resolution camera. In the first section, we introduced the topic that explains the problem statement, why it is necessary and explained why this problem is challenging. The seconds sections explains the reader about the background knowledge required to get hold of the task, where we explained representation of depth image, how can we create a depth image from a set of 3d points in the space for that we explained about the camera geometry and later we discussed about the methods by which you can capture depth and presented an in depth road map for that. After that, we discussed some of the prominent methods to capture depth in detail like stereo vision, time of flight and later we discussed the fact that we can also fuse two active-active and active-passive range sensors to measure depth cues.

In the section Literature review, we talked about the task of depth estimation and introduced the reader to single view depth estimation and discussed the possible applications of single view depth estimation. Later in the section we broadly categorized the task into four categories and in-depth literature survey is presented and out of those four categories for depth estimation is depth completion and we gave a state of the art survey for the same as our project is focused on depth completion for a single view.

To evaluate the state of the art depth completion methods we took our own indoor datasets with the help of a low-cost lidar and camera setup and with the

3d camera which were mounted on a test rig which we explained in detail in the next section called methodology. Later we talked in-depth about the pipelines to collect the datasets with the test rig and talked about the characteristics of the dataset. Later in the methodology, we discussed in detail about the state of the art depth single view depth completion methods we chose to evaluate. In the next section called results, we show the results of the evaluated methods on our collected dataset and on the KITTI depth completion dataset [85]. And we briefly discussed the performance of the evaluated methods with the help of error metrics like RMSE, MAE, iRMSE, iMAE.

From the results carried out on the KITTI depth completion benchmark [85] for the evaluated approaches and from table 5.1, we conclude that learning-based approaches outperforms for the task of depth completion on KITTI depth completion benchmark [85]. [70] has the least RMSE and MAE error, whereas [21] has the least iRMSE and iMAE error. From the definition of RMSE and MAE it can concluded that [70] completes depth more accurately for faraway objects whereas [21] completely depth more accurately for nearby objects in the scene, for further investigation you can refer to section 5.1 where we explained clearly why both of those approaches performed better on different error metrics.

But this case is totally opposite for the dataset collected by us on Intel Realsense. In that case, the non learning-based approach [47] outperforms the learning-based evaluated approaches from the table 5.2. [47] has better RMSE and MAE than all the other learning-based approaches. We have concluded two reasons for that, the approaches we evaluated are trained on KITTI depth completion benchmark [85] and the data is recorded by a Velodyne Lidar which is a different sensor modality and in different environmental conditions. In that case, when the evaluated approaches are not trained on the collected dataset it is quite obvious to give average results, whereas in this case the nonlearning approach which is totally based on classical image processing works fine as it has more generalizability than the learning-based approaches. To make them more robust we have to train them with data in different environments having different illumination conditions.

But investigating about the other error metrics like iRMSE and iMAE that are calculated on the results of evaluated approaches on the collected dataset we found that average value of these metric of all the frames is infinity but there exist some

frames which have a finite value.

	image	RMSE	MAE	iRMSE	iMAE
0	depth_image_506.png	3009.544910	1493.201183	0.440342	0.228922
1	depth_image_538.png	2505.104361	1522.092638	0.328317	0.190351
2	depth_image_583.png	3574.134862	1599.513619	0.196068	0.124449
3	depth_image_871.png	3318.427662	1783.501828	inf	inf
4	depth_image_1464.png	2873.849092	1759.489692	inf	inf
5	depth_image_1397.png	2536.386153	1843.366278	inf	inf
6	depth_image_1181.png	2705.209909	1491.723126	inf	inf
7	depth_image_1310.png	3116.913354	1683.510082	0.362492	0.256356
8	depth_image_236.png	3387.963854	1516.399537	inf	inf
9	depth_image_647.png	2053.489718	1251.058355	0.431961	0.276404

Figure 6.1: Error metric table for [21] for depth completion on some frames of the dataset collected with Intel Realsense.

	image	RMSE	MAE	iRMSE	iMAE
0	depth_image_506.png	2890.807145	1501.604035	0.416282	0.275331
1	depth_image_538.png	3209.603856	1424.648561	0.308201	0.202196
2	depth_image_583.png	2704.350832	1389.309637	inf	inf
3	depth_image_871.png	3198.273885	1300.225311	0.305746	0.162387
4	depth_image_1464.png	4317.305912	2668.403933	0.434756	0.327820
5	depth_image_1397.png	4144.558297	2876.019055	0.536335	0.383175
6	depth_image_1181.png	1234.523437	994.945589	0.362135	0.301659
7	depth_image_1310.png	3964.990670	2006.093909	0.873726	0.261643
8	depth_image_236.png	3472.826360	1649.024932	0.335286	0.200444
9	depth_image_647.png	4215.160080	1996.054507	0.606613	0.248683

Figure 6.2: Error metric table for [70] for depth completion on some frames of the dataset collected with Intel Realsense.

	image	RMSE	MAE	iRMSE	iMAE
0	depth_image_506.png	625.946056	146.865630	0.222336	0.038812
1	depth_image_538.png	379.349147	113.528291	0.128588	0.018775
2	depth_image_583.png	467.087218	134.221735	0.121731	0.017705
3	depth_image_871.png	1121.042081	179.766606	0.116825	0.017581
4	depth_image_1464.png	487.862989	145.574049	0.110814	0.015187
5	depth_image_1397.png	609.840985	178.706159	0.114600	0.016025
6	depth_image_1181.png	220.336998	55.670594	0.124892	0.017848
7	depth_image_1310.png	579.917038	149.562276	0.189228	0.033218
8	depth_image_236.png	1809.337756	256.536214	inf	inf
9	depth_image_647.png	179.596453	47.030008	0.110449	0.015028

Figure 6.3: Error metric for [47] for depth completion on some frames of the dataset collected with Intel Realsense.

Further when we look at a frames with finite iRMSE and iMAE values in the datase we found that there exist some objects in the frame that is in the maximum

range of the sensor, where as for frames with infinite values there exist no obstacles in the range range of sensor.So, the sensor in that case is behaving like it is placed at place with no objects in its maximum range, therefore most of the pixels of depth image will have invalid depth values.

Figure 6.4: Frame number 1438 from collected dataset from Realsense

Table 6.1: Error metric table for frame number 1438 of the collected dataset on [70], [47]

Appraoch	**RMSE[mm]**	**MAE[mm]**	**iRMSE[1/km]**	**iMAE[1/km]**
[70]	3883.16	1827.20	0.51	0.25
[47]	327.91	120.84	0.129	0.015

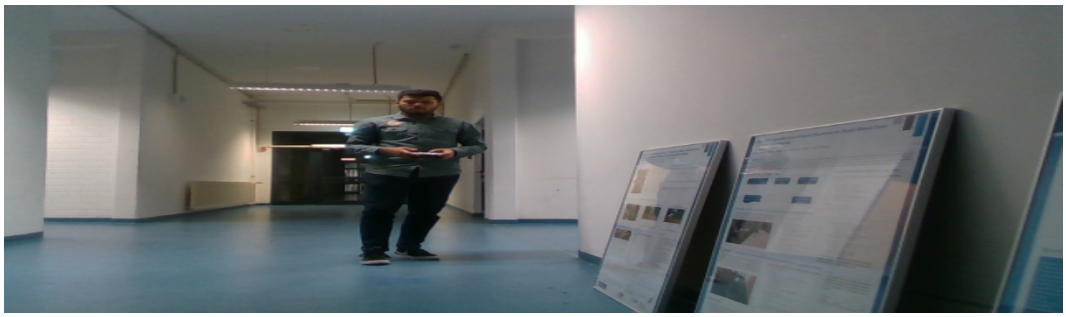


Figure 6.5: Frame number 1438 from collected dataset from Realsense

Table 6.2: Error metric table for frame number 1254 of the collected dataset on [70], [47]

Appraoch	**RMSE[mm]**	**MAE[mm]**	**iRMSE[1/km]**	**iMAE[1/km]**
[70]	4358.11	2394.83	0.51	0.37
[47]	453.20	141.44	0.186	0.033

As it was also observed that error metrics Like RMSE, MAE reduced when there exists an object in the maximum range of the sensor. So, it is clear here that the choice of sensor is really important according to the application.

So, from this, we can conclude two hypotheses. The first one is that the algorithms are designed on the assumption that there will always exist some objects in the maximum range of the sensor so that it can observe the depth of that particular object. As in the case of KITTI depth completion benchmark [85] we always some objects in the vicinity of the maximum range of the sensor which is not the case in the terms of the dataset collected on real sense. The second is the characteristics of the dataset that is captured, like just an example we record the data with a lidar having less range and if the data is recorded in a similar kind of environment what will be the result in that case. Therefore, we need to investigate more about these two things in the future.

References

[1] Teraranger one - the lightweight and low-cost tof distance measurement sensor, 14m, 8 grams. URL https://www.terabee.com/shop/lidar-tof-range-finders/teraranger-one/.

[2] Z. Bao, B. Li, and W. Zhang. Robustness of tof and stereo fusion for high-accuracy depth map. *IET Computer Vision*, 13(7):676–681, 2019. ISSN 1751-9640. doi: 10.1049/iet-cvi.2018.5476.

[3] Jonathan T. Barron and Ben Poole. The fast bilateral solver. *CoRR*, abs/1511.03296, 2015. URL http://arxiv.org/abs/1511.03296.

[4] Kristian. Bredies, Karl. Kunisch, and Thomas. Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010. doi: 10.1137/090769521. URL https://doi.org/10.1137/090769521.

[5] P. Zanuttigh C. Dal Mutto and G. M.Cortelazzo. Time-of-flight cameras and-microsoft kinecttm, January 24 2013. URL https://lttm.dei.unipd.it//nuovo/Papers/ToF-Kinect-book.pdf.

[6] Cesar Cadena, Anthony Dick, and Ian Reid. Multi-modal auto-encoders as joint estimators for robotics scene understanding. 06 2016. doi: 10.15607/RSS.2016.XII.041.

[7] X. Chen, M. Zhao, L. Xiang, F. Sugai, H. Yaguchi, K. Okada, and M. Inaba. Development of a low-cost ultra-tiny line laser range sensor. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 111–116, Oct 2016. doi: 10.1109/IROS.2016.7759042.

[8] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *CoRR*, abs/1810.02695, 2018. URL http://arxiv.org/abs/1810.02695.

[9] Nathaniel Chodosh, Chaoyang Wang, and Simon Lucey. Deep convolutional compressed sensing for lidar depth completion. *CoRR*, abs/1803.08949, 2018. URL http://arxiv.org/abs/1803.08949.

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016. URL http://arxiv.org/abs/1604.01685.

[11] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13 (9):1200–1212, Sep. 2004. ISSN 1941-0042. doi: 10.1109/TIP.2004.833105.

[12] James Diebel and Sebastian Thrun. An application of markov random fields to range sensing. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 291–298. MIT Press, 2006. URL http://papers.nips.cc/paper/2837-an-application-of-markov-random-fields-to-range-sensing.pdf.

[13] J. Dolson, J. Baek, C. Plagemann, and S. Thrun. Upsampling range data in dynamic environments. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1141–1148, June 2010. doi: 10.1109/CVPR.2010.5540086.

[14] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio López, and Vladlen Koltun. CARLA: an open urban driving simulator. *CoRR*, abs/1711.03938, 2017. URL http://arxiv.org/abs/1711.03938.

[15] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. 03 2016.

[16] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *CoRR*, abs/1411.4734, 2014. URL `http://arxiv.org/abs/1411.4734`.

[17] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. Confidence propagation through cnns for guided sparse depth regression. *CoRR*, abs/1811.01791, 2018. URL `http://arxiv.org/abs/1811.01791`.

[18] G. D. Evangelidis, M. Hansard, and R. Horaud. Fusion of range and stereo data for high-resolution scene-modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2178–2192, Nov 2015. ISSN 1939-3539. doi: 10.1109/TPAMI.2015.2400465.

[19] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. *CoRR*, abs/1605.06457, 2016. URL `http://arxiv.org/abs/1605.06457`.

[20] V. Gandhi, J. Čech, and R. Horaud. High-resolution depth maps based on tof-stereo fusion. In *2012 IEEE International Conference on Robotics and Automation*, pages 4742–4749, May 2012. doi: 10.1109/ICRA.2012.6224771.

[21] Wouter Van Gansbeke, Davy Neven, Bert De Brabandere, and Luc Van Gool. Sparse and noisy lidar completion with RGB guidance and uncertainty. *CoRR*, abs/1902.05356, 2019. URL `http://arxiv.org/abs/1902.05356`.

[22] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[23] Jason Geng. Structured-light 3d surface imaging: a tutorial. *Adv. Opt. Photon.*, 3(2):128–160, Jun 2011. doi: 10.1364/AOP.3.000128. URL `http://aop.osa.org/abstract.cfm?URI=aop-3-2-128`.

[24] IDS Imaging Development Systems GmbH. Obtaining depth information from stereo images. URL `https://de.ids-imaging.com/tl_files/downloads/whitepaper/IDS_Whitepaper_3D_Stereo_Vision.pdf`.

[25] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *CoRR*, abs/1609.03677, 2016. URL `http://arxiv.org/abs/1609.03677`.

[26] W. Hannemann, André Linarth, B. Liu, Gabriella Kókai, and Oliver Jesorsky. Increasing depth lateral resolution based on sensor fusion. *IJISTA*, 5:393–401, 01 2008. doi: 10.1504/IJISTA.2008.021302.

[27] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003. ISBN 0521540518.

[28] Kenji Hata and Silvio Savarese. Cs231a course notes 3: Epipolar geometry. URL `http://web.stanford.edu/class/cs231a/course_notes/03-epipolar-geometry.pdf`.

[29] S. Hawe, M. Kleinsteuber, and K. Diepold. Dense disparity maps from sparse disparity measurements. In *2011 International Conference on Computer Vision*, pages 2126–2133, Nov 2011. doi: 10.1109/ICCV.2011.6126488.

[30] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 1–14, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15549-9.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL `http://arxiv.org/abs/1512.03385`.

[32] Radu Horaud, Miles Hansard, Georgios Evangelidis, and Clément Ménier. An Overview of Depth Cameras and Range Scanners Based on Time-of-Flight Technologies. *Machine Vision and Applications*, 27(7):1005–1020, October 2016. doi: 10.1007/s00138-016-0784-4. URL `https://hal.inria.fr/hal-01325045`.

[33] M. Hornácek, C. Rhemann, M. Gelautz, and C. Rother. Depth super resolution by rigid body self-similarity in 3d. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1123–1130, June 2013. doi: 10.1109/CVPR.2013.149.

[34] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. URL `http://arxiv.org/abs/1704.04861`.

[35] Zixuan Huang, Junming Fan, Shuai Yi, Xiaogang Wang, and Hongsheng Li. Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. *CoRR*, abs/1808.08685, 2018. URL `http://arxiv.org/abs/1808.08685`.

[36] Benjamin Huhle, Timo Schairer, Philipp Jenke, and Wolfgang Straßer. Fusion of range and color images for denoising and resolution enhancement with a non-local filter. *Computer Vision and Image Understanding*, 114(12):1336 – 1345, 2010. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2009.11.004. URL `http://www.sciencedirect.com/science/article/pii/S1077314210001712`. Special issue on Time-of-Flight Camera Based Computer Vision.

[37] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[38] P. Debevec M. Levoy S. Nayar J.-Y. Bouguet, B. Curless and S. Seitz. Overview of active vision techniques. siggraph 2000 course on 3d photography. URL `http://www.cs.cmu.edu/~seitz/course/SIGG99/slides/curless-active.pdf`.

[39] Maximilian Jaritz, Raoul de Charette, Émilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. Sparse and dense data with cnns: Depth completion and semantic segmentation. *CoRR*, abs/1808.00769, 2018. URL `http://arxiv.org/abs/1808.00769`.

[40] Jiajun Lu and D. Forsyth. Sparse depth super resolution. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2245–2253, June 2015. doi: 10.1109/CVPR.2015.7298837.

[41] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Transactions on Pattern Analysis*

and Machine Intelligence, 36(11):2144–2158, Nov 2014. ISSN 1939-3539. doi: 10.1109/TPAMI.2014.2316835.

[42] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth extraction from video using non-parametric sampling. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, ECCV'12, page 775–788, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 9783642337147. doi: 10.1007/978-3-642-33715-4_56. URL https://doi.org/10.1007/978-3-642-33715-4_56.

[43] Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig, and David Forsyth. Automatic scene inference for 3d object compositing. *ACM Trans. Graph.*, 33(3), June 2014. ISSN 0730-0301. doi: 10.1145/2602146. URL https://doi.org/10.1145/2602146.

[44] Martin Kiefel, Varun Jampani, and Peter V. Gehler. Sparse convolutional networks using the permutohedral lattice. *CoRR*, abs/1503.04949, 2015. URL http://arxiv.org/abs/1503.04949.

[45] Florian Knoll, Kristian Bredies, Thomas Pock, and Rudolf Stollberger. Second order total generalized variation (tgv) for mri. *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 65:480–91, 02 2011. doi: 10.1002/mrm.22595.

[46] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, 26(3):to appear, 2007.

[47] J. Ku, A. Harakeh, and S. L. Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 16–22, May 2018. doi: 10.1109/CRV.2018.00013.

[48] K. Kuhnert and M. Stommel. Fusion of stereo-camera and pmd-camera data for real-time suited precise 3d environment reconstruction. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4780–4785, Oct 2006. doi: 10.1109/IROS.2006.282349.

[49] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. *CoRR*, abs/1702.02706, 2017. URL http://arxiv.org/abs/1702.02706.

[50] L'ubor Ladický, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, page 89–96, USA, 2014. IEEE Computer Society. ISBN 9781479951185. doi: 10.1109/CVPR.2014.19. URL https://doi.org/10.1109/CVPR.2014.19.

[51] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. *CoRR*, abs/1606.00373, 2016. URL http://arxiv.org/abs/1606.00373.

[52] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *International Journal of Robotics Research*, 34, 01 2013. doi: 10.1177/0278364914549607.

[53] Evan Lerner. Meet piccolissimomeet piccolissimo: The world's smallest self-powered controllable flying vehicle. URL https://penntoday.upenn.edu/spotlights/meet-piccolissimo-worlds-smallest-self-powered-controllable-flying-vehicle

[54] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016. URL http://arxiv.org/abs/1612.03144.

[55] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017. URL http://arxiv.org/abs/1708.02002.

[56] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian D. Reid. Learning depth from single monocular images using deep convolutional neural fields. *CoRR*, abs/1502.07411, 2015. URL http://arxiv.org/abs/1502.07411.

[57] Lee-Kang Liu, Stanley H. Chan, and Truong Q. Nguyen. Sparse reconstruction of depth data: Representation, algorithm, and sampling. *CoRR*, abs/1407.3840, 2014. URL `http://arxiv.org/abs/1407.3840`.

[58] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, June 2014. doi: 10.1109/CVPR.2014.97.

[59] S. Lu, X. Ren, and F. Liu. Depth enhancement via low-rank matrix completion. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3390–3397, June 2014. doi: 10.1109/CVPR.2014.433.

[60] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. *CoRR*, abs/1701.04128, 2017. URL `http://arxiv.org/abs/1701.04128`.

[61] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. *CoRR*, abs/1709.07492, 2017. URL `http://arxiv.org/abs/1709.07492`.

[62] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. *CoRR*, abs/1807.00275, 2018. URL `http://arxiv.org/abs/1807.00275`.

[63] MathWorks. What is camera calibration? URL `https://ch.mathworks.com/help/vision/ug/camera-calibration.html#bu0nh2`.

[64] Rahul Nair, Kai Ruhl, Frank Lenzen, Stephan Meister, Henrik Schäfer, Christoph S. Garbe, Martin Eisemann, Marcus Magnor, and Daniel Kondermann. *A Survey on Time-of-Flight Stereo Fusion*, pages 105–127. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-44964-2. doi: 10.1007/978-3-642-44964-2_6. URL `https://doi.org/10.1007/978-3-642-44964-2_6`.

[65] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[66] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. *CoRR*, abs/1603.06937, 2016. URL `http://arxiv.org/abs/1603.06937`.

[67] Tanguy Ophoff, Kristof Van Beeck, and Toon Goedemé. Exploring rgb+depth fusion for real-time object detection. *Sensors*, 19(4), 2019. ISSN 1424-8220. doi: 10.3390/s19040866. URL `https://www.mdpi.com/1424-8220/19/4/866`.

[68] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. *CoRR*, abs/1611.08323, 2016. URL `http://arxiv.org/abs/1611.08323`.

[69] Christopher V. Poulton and Michael R. Watts. Mit and darpa pack lidar sensor onto single chip. URL `https://spectrum.ieee.org/tech-talk/semiconductors/optoelectronics/mit-lidar-on-a-chip`.

[70] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. *CoRR*, abs/1812.00488, 2018. URL `http://arxiv.org/abs/1812.00488`.

[71] Eduardo Romera, Jose M. Alvarez, Luis Bergasa, and Roberto Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, PP:1–10, 10 2017. doi: 10.1109/TITS.2017.2750080.

[72] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL `http://arxiv.org/abs/1505.04597`.

[73] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, June 2016. doi: 10.1109/CVPR.2016.352.

[74] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, 31(5):824–840, May 2009. ISSN 1939-3539. doi: 10.1109/TPAMI.2008.132.

[75] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1161–1168. MIT Press, 2006. URL `http://papers.nips.cc/paper/2921-learning-depth-from-single-monocular-images.pdf`.

[76] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I, June 2003. doi: 10.1109/CVPR.2003.1211354.

[77] Nick Schneider, Lukas Schneider, Peter Pinggera, Uwe Franke, Marc Pollefeys, and Christoph Stiller. Semantically guided depth upsampling. *CoRR*, abs/1608.00753, 2016. URL `http://arxiv.org/abs/1608.00753`.

[78] Ju Shen and Sen-Ching S. Cheung. Layer depth denoising and completion for structured-light rgb-d cameras. In *CVPR*, pages 1187–1194. IEEE Computer Society, 2013. ISBN 978-0-7695-4989-7. URL `http://dblp.uni-trier.de/db/conf/cvpr/cvpr2013.html#ShenC13`.

[79] Shreyas S. Shivakumar, Kartik Mohta, Bernd Pfrommer, Vijay Kumar, and Camillo J. Taylor. Real time dense depth estimation by fusing stereo with sparse depth measurements. *CoRR*, abs/1809.07677, 2018. URL `http://arxiv.org/abs/1809.07677`.

[80] Shreyas S. Shivakumar, Ty Nguyen, Steven W. Chen, and Camillo J. Taylor. Dfusenet: Deep fusion of RGB and sparse depth information for image guided dense depth completion. *CoRR*, abs/1902.00761, 2019. URL `http://arxiv.org/abs/1902.00761`.

[81] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pages 1297–1304, June 2011. doi: 10.1109/CVPR.2011.5995316.

[82] Jie Tang, Feipeng Tian, Wei Feng, Jian Li, and Ping Tan. Learning guided convolutional network for depth completion. 08 2019.

[83] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 839–846, Jan 1998. doi: 10.1109/ICCV.1998.710815.

[84] Marjan Trobina. Error model of a coded-light range sensor. 05 1998.

[85] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017.

[86] L. Wang, H. Jin, R. Yang, and M. Gong. Stereoscopic inpainting: Joint color and depth completion from stereo images. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587704.

[87] R. Wood. Robobees. URL `https://wyss.harvard.edu/technology/robobees-autonomous-flying-microrobots/`.

[88] Bojian Wu, Yang Zhou, Yiming Qian, Minglun Gong, and Hui Huang. Full 3d reconstruction of transparent objects. *CoRR*, abs/1805.03482, 2018. URL `http://arxiv.org/abs/1805.03482`.

[89] J. Xie, C. Chou, R. Feris, and M. Sun. Single depth image super resolution and denoising via coupled dictionary learning with local constraints and shock filtering. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, July 2014. doi: 10.1109/ICME.2014.6890325.

[90] J. Xie, R. S. Feris, and M. Sun. Edge guided single depth image super resolution. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 3773–37777, Oct 2014. doi: 10.1109/ICIP.2014.7025766.

[91] Jingyu Yang, Xinchen Ye, Kun Li, Chunping Hou, and Yao Wang. Color-guided depth recovery from rgb-d data using an adaptive autoregressive model. *IEEE Transactions on Image Processing*, 23(8):3443–3458, 2014. ISSN 1057-7149. doi: 10.1109/TIP.2014.2329776.

[92] M. Ye, Yu Zhang, R. Yang, and D. Manocha. 3d reconstruction in the presence of glasses by acoustic and stereo fusion. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4885–4893, June 2015. doi: 10.1109/CVPR.2015.7299122.

[93] Raymond A. Yeh, Chen Chen, Teck-Yian Lim, Mark Hasegawa-Johnson, and Minh N. Do. Semantic image inpainting with perceptual and contextual losses. *CoRR*, abs/1607.07539, 2016. URL `http://arxiv.org/abs/1607.07539`.

[94] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Free-form image inpainting with gated convolution. *CoRR*, abs/1806.03589, 2018. URL `http://arxiv.org/abs/1806.03589`.

[95] Yinda Zhang and Thomas A. Funkhouser. Deep depth completion of a single RGB-D image. *CoRR*, abs/1803.09326, 2018. URL `http://arxiv.org/abs/1803.09326`.

[96] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. *CoRR*, abs/1704.07813, 2017. URL `http://arxiv.org/abs/1704.07813`.

[97] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012, 2017. URL `http://arxiv.org/abs/1707.07012`.