Paris Saclay University

Master thesis

Machine Learning for Image Based 3D Reconstruction

Tarek BOUAMER

August 29, 2019

Advisors: Prof. Dr. Friedrich Fraundorfer
Institute of Computer Graphics and Vision, TU Graz

Abstract

In 3D reconstruction; depth and normal maps estimates based on multi view stereo Patch Match has shown a great results in terms of accuracy; however; in textureless surfaces; photo consistency based correspondence fails to match and provide pixel-wise depth estimate for large portion in the scenes.

In this master thesis, we assume that textureless areas are smooth and piecewise planar, we combine between low-level segmentation and Plane masks generated using Convolutional Neural Network (CNN) to construct local planes. The plane estimates are used to fill the missing depth and normal surfaces.

Finally; we evaluate our algorithm in terms of accuracy and completeness on the ETH 3D high resolution dataset.

Acknowledgement

Firstly, I would like to express my sincere gratitude to my advisor Prof. Friedrich Fraundorfer for the continuous support in this interesting research topic, motivation, and immense knowledge.

Besides my advisor, I would like to thank Dr. Andrea Kuhn for his feedbacks and encouragement, and my project partner Christian Sormann for his support the great discussions.

I thank also all members of Computer Graphic and Vision (ICG) for their support, knowledge sharing and for all the fun we have had in the last four months. Many thanks also go out to Graz Technical University for the great welcoming and making this internship possible.

Lastly, I want to thank my family members for their support and helping me through my education.

Table of content

Chapter 1

I. Introduction

Multi-View Stereo (MVS) Reconstruction aims to recover a dense 3D representation of the scene from a set of views; in last decade, several successful MVS algorithms have been proposed with variety of datasets.

Among these algorithms; Patch Match based algorithms [1] are still the best and outperforming learning methods; for instance, Colmap [2] produces an accurate depth maps estimates and better than the recently developed DeepMVS [3] Multi view stereo network. The resulting depth and normal maps are then fused geometrically into scene point clouds. However, Patch Match based methods fails to estimates depth in textureless surfaces for their smoothness and similarity appearance.

The problem of completion in textureless surfaces has been recently addressed in TAPA-MVS [4] and PCF-MVS [5] assuming that textureless surfaces are often smooth piecewise planar. Both method uses low level segmentation superpixels to generate plane hypothesis based on their photo consistency metric. In this thesis, we propose a new plane segmentation method based on instance plane segmentation CNN model; combined with low-level superpixel similarity appearance segmentation as in [4] [5]. As in PCF-MVS, a fitted plane parameters for each plane-superpixel are used to estimate depth for missing pixels in Colmap depth maps.

To validate our method; we perform an evaluation on the recently published training dataset ETH 3D for its scene variety, high resolution view with a precise ground truth measures.

This master thesis text is structured as follows:

- Chapter 2: Principles and fundamentals of Structure of Motion (SFM) and Multi View Geometry.
- Chapter 3: 3D Reconstruction (sparse, dense).
- Chapter 4: Depth Completion process and Plane fitting.
- Chapter 5: Evaluation on ETH 3D benchmark.
- Chapter 6, Conclusion and future work are given.

Chapter 2

II. Principles

In this chapter, we introduce the main principles of multi view stereo (MVS) and the general structure of our 3D Reconstruction pipelines.

1. Image Description

In multi view stereo system, the main interest is to extract locally the geometrical association between a multiple of views to reconstruct the 3D model of the scene.

Typically, each view is described by its local **features** with high colour gradient as an edges, corners, …; and Ideally, these points features should be sparsely detected and robust against the photometric (illumination, noise) and geometric (rotation, scale) variations. The **Scale Invariant Feature Transform (SIFT)** [6]; most popular and widely used descriptor; encodes the spatial information of an image using local interest points and neighbouring pixels as gradient (orientation and magnitude). Features **matching** finds the pixel level correspondence between the features in the source image and others images that points to the same object in terms of correlation.

2. Multi View Geometry

Before moving to Multi View Geometry and discussing the general problem of pose calculation using the detected point features, it is necessary to present the **camera model** and **perspective projection** that describes the image formation through a linear mapping from 3D homogenous world frame $X \in \Re^3$ to 2D representation $x \in \Re^2$ in image frame. Thus; this projection is formulated as:

$$x = K\,[R \quad T]\,X$$

$$x = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.1}$$

In 2.1, K represents the *intrinsic* parameters that describes the properties of the camera; while $R \in SO(3)$ and $T \in \Re^3$ defines the Euclidean transformation “The 3×3 Rotational and Translation vector” from the world to the camera coordinates system; known as *extrinsic* parameters.

2.1. Camera Calibration

Camera calibration is the process to determine the intrinsic and extrinsic parameters of the camera model for an uncalibrated camera; one of the most used camera calibration techniques is the one proposed by Tsai [6]; that requires more than six 3D points $n_p > 6$ to identify the 12 unknown parameters of camera model.

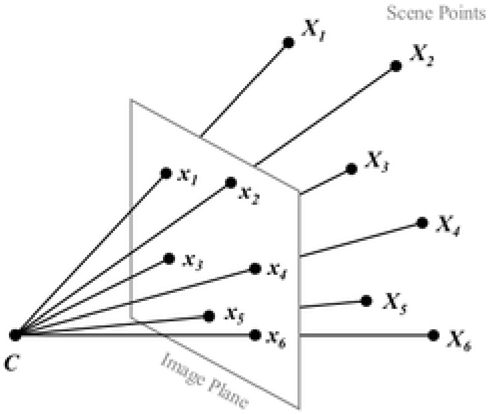


Figure 2.1: Camera Calibration from 2D-3D correspondence.

Given a 2D observations x and its corresponding 3D point X, the camera parameters are estimated as:

$$x = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim M\,X = \begin{bmatrix} {m_1}^T \\ {m_2}^T \\ {m_3}^T \end{bmatrix} . \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad => \quad \begin{cases} ({m_1}^T - u_i\, {m_3}^T).X_i = 0 \\ ({m_2}^T - v_i\, {m_3}^T).X_i = 0 \end{cases} \tag{2.2}$$

The problem can be then re-arranged to obtain a system of equation with n observations (figure 2.1):

$$\begin{bmatrix} {X_1}^T & 0^T & -u_1{X_1}^T \\ 0^T & {X_1}^T & -v_1{X_1}^T \\ \cdots & \cdots & \cdots \\ {X_n}^T & 0^T & -u_n{X_n}^T \\ 0^T & {X_n}^T & -v_n{X_n}^T \end{bmatrix} . \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix} \quad => \quad Q.M = 0 \tag{2.3}$$

A solution is to minimize $||Q.M||^2$ subjected to the constraint $||M||^2$. It can be solved through Singular Value Decomposition (SVD) where the solution is the eigenvector corresponding to the smallest eigenvalue of the matrix $Q\ \ T\ Q$; since it is the unit vector x that minimizes$||Q.M||^2 = x^T Q^T Q\ x$.

The reverse process, **Triangulation**, is to determine 3D point X by intersecting multiple visual rays from the corresponding 2D projections x_i. With known camera model of at least two observations; 3D pose is estimated using Direct Linear Transformation (DLT) as in camera calibration.

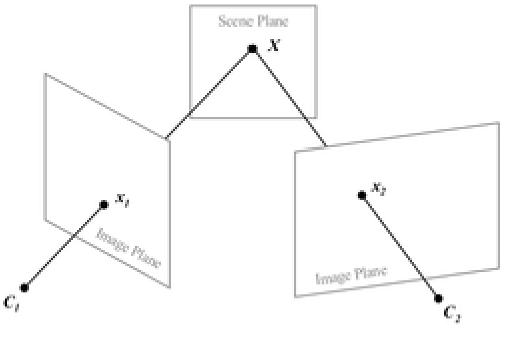


Figure 2.2: Two View triangulation for homography.

2.2. Two View Geometry

In uncalibrated multiple view geometry; a reconstruction of both cameras and scene structure can be computed from point feature correspondence.

a. Homography

Homography is projective transformation that relates pixel coordinates between two images $x_2 = h(\, x_1 \,)$ if both images are viewing the same plane (parallax); we say, the homography induced by two image planes (figure 2.2); therefore, $Z = 0$ and the system of equation in camera calibration 2.3 can be rewritten as follows:

$$x = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim H\,X = \begin{bmatrix} {h_1}^T \\ {h_2}^T \\ {h_3}^T \end{bmatrix} . \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \quad => \quad H.M = 0 \tag{2.4}$$

The model matrix H has 8 degrees of freedoms; we use DLT to estimate the homography and we require at least 4 non-collinear points.

b. Epipolar

For general scenes; we introduce the epipolar concept in modelling two view geometry. We define the epipolar geometry between two views as the intersection of the image planes with pencil of planes joining the centres of the cameras with a baseline line. The epipole is the point of intersection of the line joining the camera centres with epipolar plane as shown in figure 2.3.

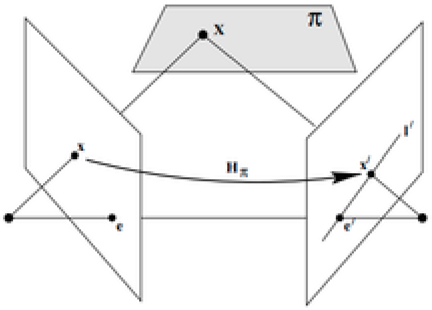


Figure 2.3: Two View geometry and epipolar with necessary condition for correspondence.

We have $x' = H\,x$ and $l' = [e']_\times\, x'$, implies that $l' = [e']_\times\, (H\,x) = F\,x$; where F is defined to be the *fundamental matrix*. If x and x' corresponds, then x' lies in the epipolar line $l' = F\,x$. In other words, $x'^T\, l' = 0$ implies that $x'^T\, F\,x = 0$ as necessary condition for correspondence. In case of calibrated cameras, we define $x'' = K^{-1}\,x$ as inverse point to x in normalized coordinates; and by rewriting the condition for correspondence formula $(K'x')^T\, F\,(K\,x) = 0 \;=>\; x'^T\, E\,x = 0$ with $E = \; K'^T\, F\,K$. The fundamental matrix can be estimated from 8 points correspondences up to a scale factor enforcing the rank to 2; for more details, refer to [7] page 279.

2.3. N-View Geometry

In large number of views, linear estimation methods are not sufficient and robust to produce an accurate estimation due to redundancy and outliers (noise, occlusions, …); therefore, we wish to estimate the projection matrix P and 3D points X assuming that the measurements follows Gaussian distribution $\hat{x} = \hat{P}\,\hat{X}$ in minimizing cost function 2.5 between the projected points and measured image points in every view; known as *Bundle Adjustment (BA)*:

$$\min_{\hat{P}^i, \hat{X}_j} \sum_{i\,j} d\left(\hat{P}^i\, \hat{X}_j, \hat{x}_j^i\right)^2 \tag{2.5}$$

Where $d(x', x)$ is the distance between point x and reprojection x'. BA refinement is used as final step in reconstruction and it requires a good initialization of X and P which can be done using DLT as described previously.

RANSAC (Random Sample Consensus) is another robust model fitting algorithm in presence of outliers; that can be applied to estimate the camera calibration, Structure of motion, Homography, Primitives …; so that the estimate is unaffected by outliers.

RANSAC algorithm is described as follows:

- *Randomly select* n *sample of set A that have Npoints.*
- *Fit the model using the* n *points.*
- *Determine the set of points S which are within distance threshold* d_t *to the fitted model (number of inliers).*
- *Compute the distances of all other points in S from this model.*
- *We choose the model that has maximum of inliers after* k *iterations.*

According to the previous description, RANSAC aims to maximize the number of inliers points with $\sigma = 0.95$ confidence level; mathematically, we define this optimization problem with the following robust cost function (Huber):

$$D = \sum_i \gamma(d_i) \quad with \quad \gamma(d_i) = \begin{cases} {d_i}^2 & {d_i}^2 < {d_t}^2 \quad Inliers \\ {d_t}^2 & {d_i}^2 \geq {d_t}^2 \quad Outliers \end{cases} \tag{2.6}$$

A typical stopping criterion k for RANSAC with p confidence level, ε inlier ratio and s samples is set to:

$$k = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^s)} \tag{2.7}$$

Chapter 3

III. 3D Reconstruction

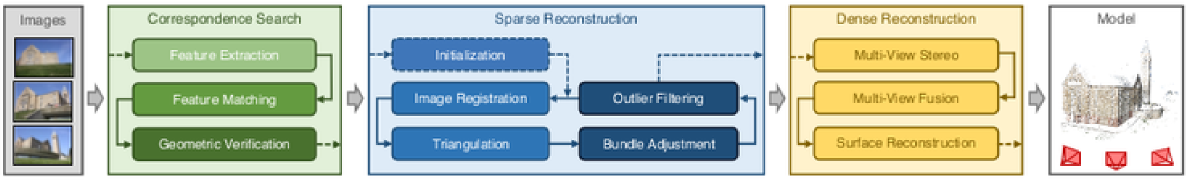


Figure 3.1: Colmap 3D reconstruction pipeline.

1. Correspondence Search

The first stage in 3D reconstruction, is the correspondence search that processes set of input images aiming to generate a 3D point clouds and construct a visibility graph, it consists of the following steps:

- Local features extraction from the source images and match them with different images using SIFT, a hierarchical indexing approach is used to search for correspondences in most similar views.
- The matched features are geometrically verified if they point to the same scene point or not by estimating the transformation between two views using RANSAC. The geometrically verified features are registered in database with their images indexes constructing a graph.

2. Sparse Reconstruction

In the second stage, we use the visibility graph and the triangulated correspondences in database to estimate the camera parameters in incremental reconstruction approach that repeatedly add a new image to the existing reconstruction followed by triangulation and refinement BA. The two pair views has to be carefully selected for a good initialization in estimating the intrinsic and extrinsic parameters of camera; likewise, to register a new view it has to be well chosen to produce an accurate pose estimation in the presence of outliers.

In the other hand; to register a new image; it has to share part of the reconstructed scene points and shares seen points that are not constructed yet with at least one registered view; therefore, the new set of 3D point points will be extended and decisive in selecting next best view.

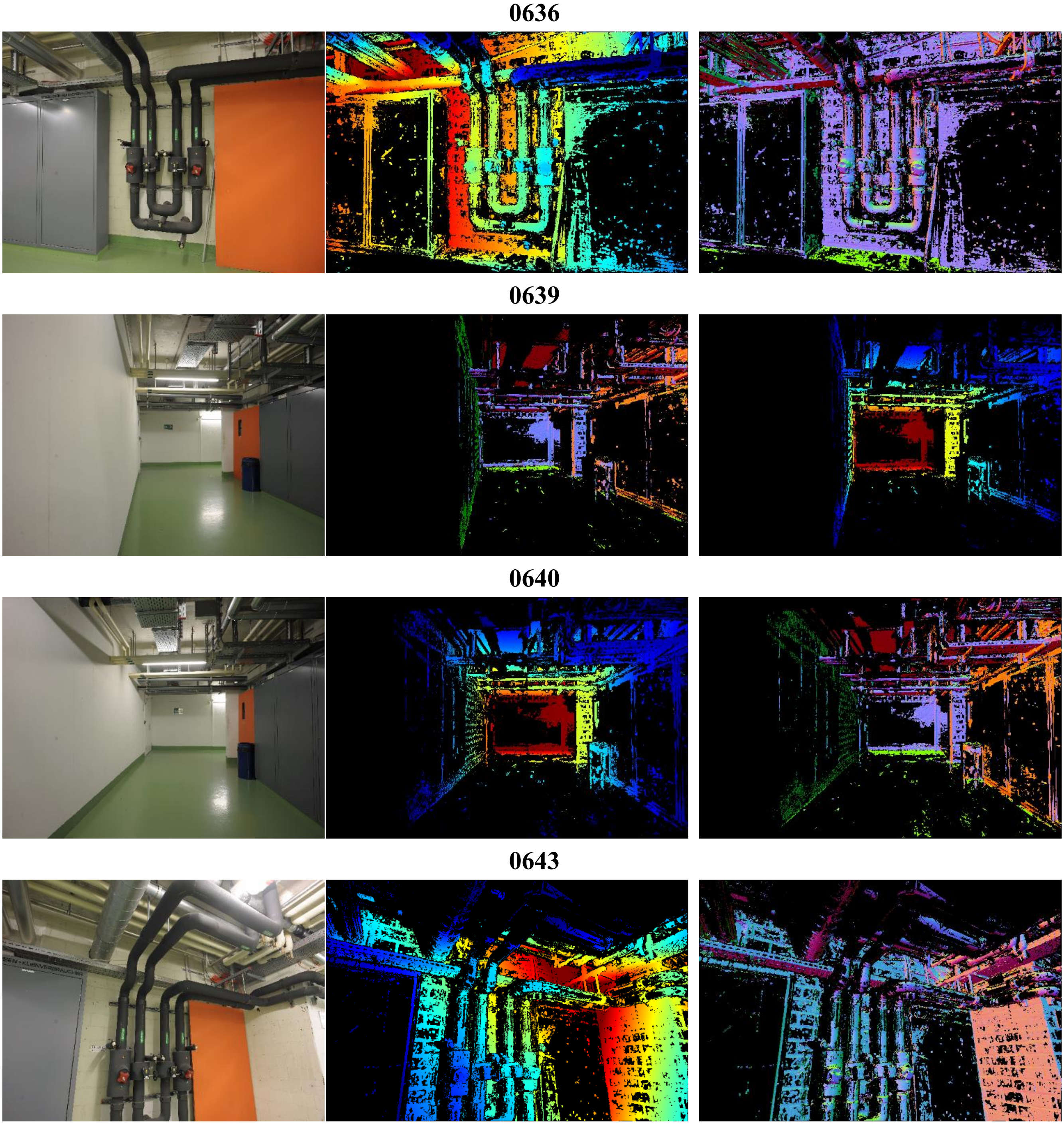
To mitigate the accumulated drift in in incremental reconstruction; Full BA is necessary to refine the camera parameters and 3D point poses jointly as introduced in [43]. Some outliers may survive in the robust estimation leading to non-valid triangulation; in multi view, redundancy is key to strictly remove the triangulation mismatches.

3. Dense Reconstruction

In dense reconstruction; we aim to densify the captured wold representation using a dense correspondence starting from depth/normal maps estimation, fusion in scene point clouds then surface meshing step.

Once the camera parameters are estimated; Depth and normal maps can be recovered by constructing epipolar geometries between the corresponding images to aggregate the appearance similarity between the views. Therefore; the depth is estimated through stereo comparison among patches from different views referred as photometric measures as Normalized Cross Correlation (NCC); where the normal is deduced directly from the recovered depth map as gradient information.

The depth and normal estimates will be then fused to the scene as a dense point clouds. To improve the quality and accuracy of fusion; we perform depth/ normal consistency check (figure 3.1) between the view to filter out the outliers and invalid information.



0636

0639

0640

0643

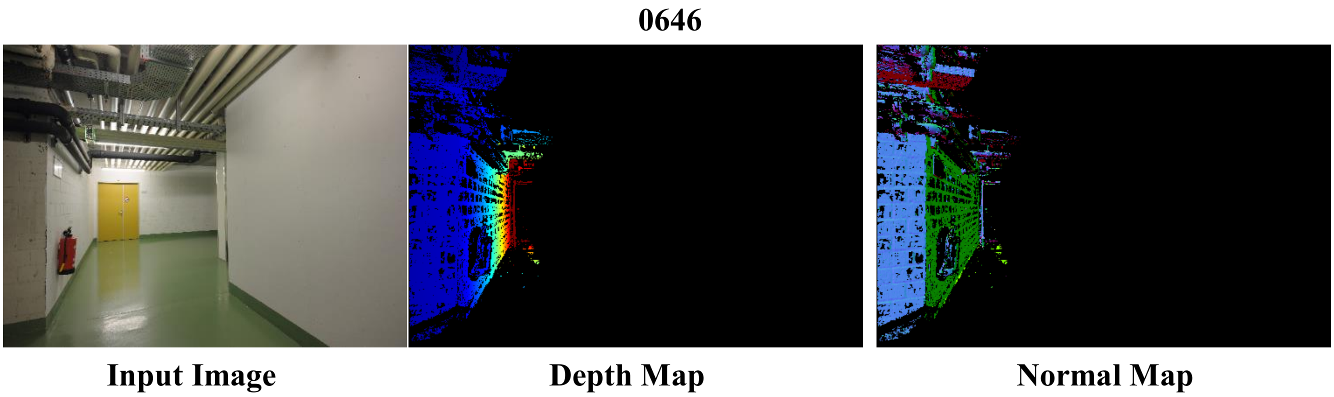


Figure 3.1(a): 3D Dense Reconstruction for *pipes* scene in ETH 3D dataset
(left) RGB image (middle) depth map (right) normal map

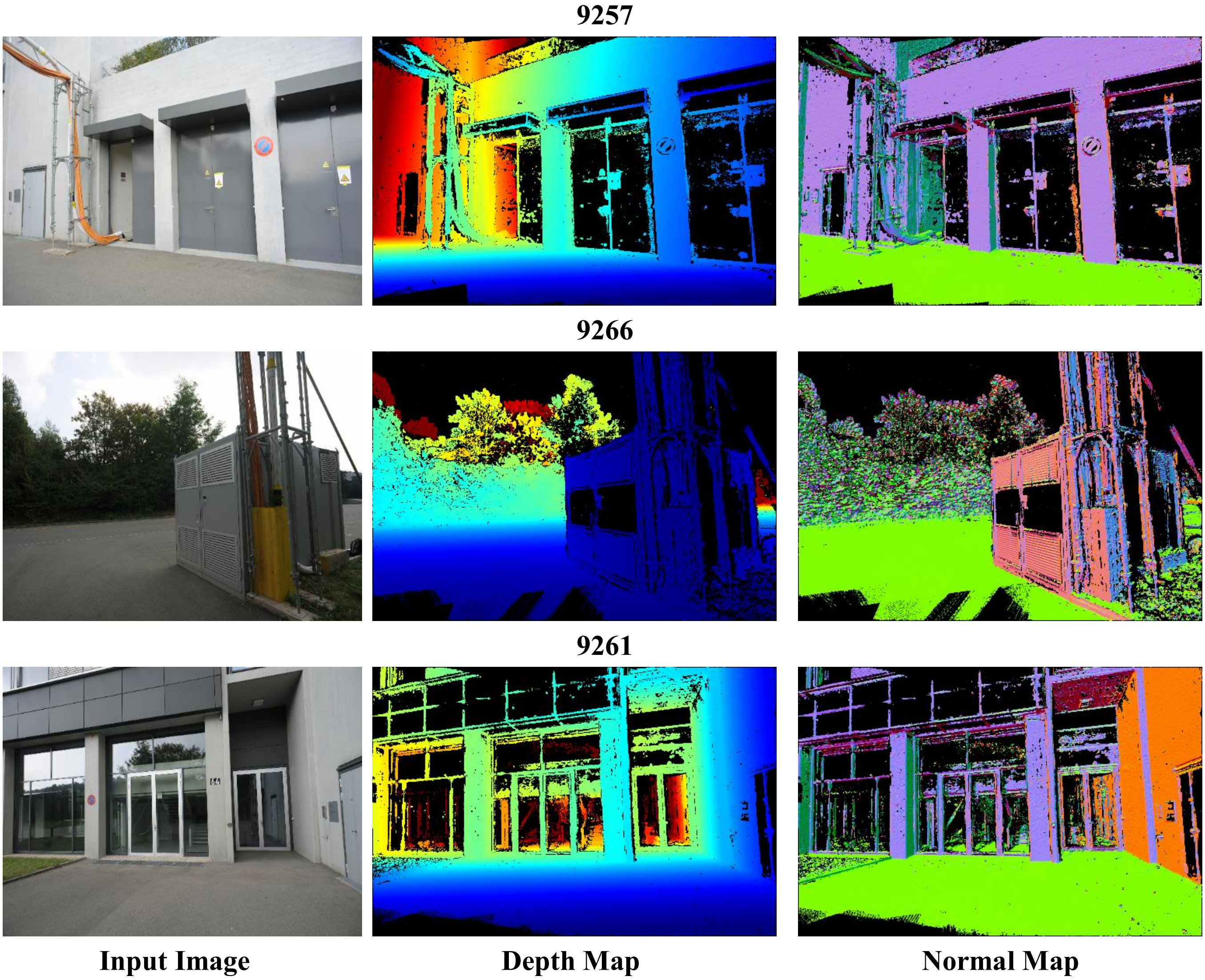


Figure 3.1(b): 3D Dense Reconstruction for *Electro* scene in ETH 3D dataset
(left) RGB image (middle) depth map (right) normal map

Chapter 4

IV. Depth Completion

1. Motivation

The quality of fused scene depends on the accuracy and completeness of depth and normal maps which is critical for robotic and virtual reality applications. For indoor scenes; depth information is missing in large bright and texture less surfaces which can be critical in autonomous navigation where these surfaces represents an obstacles.

So far; our current 3D dense reconstruction pipeline produces an accurate and reliable depth maps, however these information is sparse and missing in large portion in the scenes. Thus through this thesis; we address the problem of the completeness from initial sparse accurate input depth maps.

In upcoming section; we present the related works that addresses the depth completion from sparse depth maps, then we present our depth completion algorithm based on plane detection network and fitting. Last; we evaluate our approach on ETH3D bench mark in terms of accuracy and completeness.

Many methods have been proposed for predicting depth in missing pixels, as depth inpainting with smoothness prior [8], fast marching methods [9], patch based image synthesis [10] and back ground surface extrapolation [11]. In high resolution images, several methods have been proposed to improve the spatial resolution of depth map which is more challenging than low resolution images including Markov random Filed [12], Segmentation [13].

One strategy to enhance depth maps is to leverage scene shape; for indoor scenes for examples; planes and quadratic surfaces fitting has been proposed to improve depth maps completeness and filling local holes as in [14] [15].

In this work, to detect surface regions; we use plane detector network to extract large textureless surfaces as side walls, floor…; additionally, superpixel segmentation is used to cluster the input RGB image into regions. We aim to estimate plane parameters locally in each cluster and perform merging mechanism based on potential relationship between clusters and improve the segmentation mask. The plane parameters will be used to complete the sparse input depth and normal from our 3D reconstruction pipeline.

2. 3D Plane Detection

Human are remarkably effective in using salient global structure such as planes, symmetric and smooth surfaces; taking this global information can an advantage in to produce an accurate and complete reconstruction. However; traditional techniques [16] are computationally challenging and rely on low level features by global optimization procedure makes those methods less robust. Recently, deep learning based methods have shown promising results in detecting planes and room layouts PlanNet [17], Plane Recover [18] and recently PlaneR-CNN [19].

2.1. Plane R-CNN

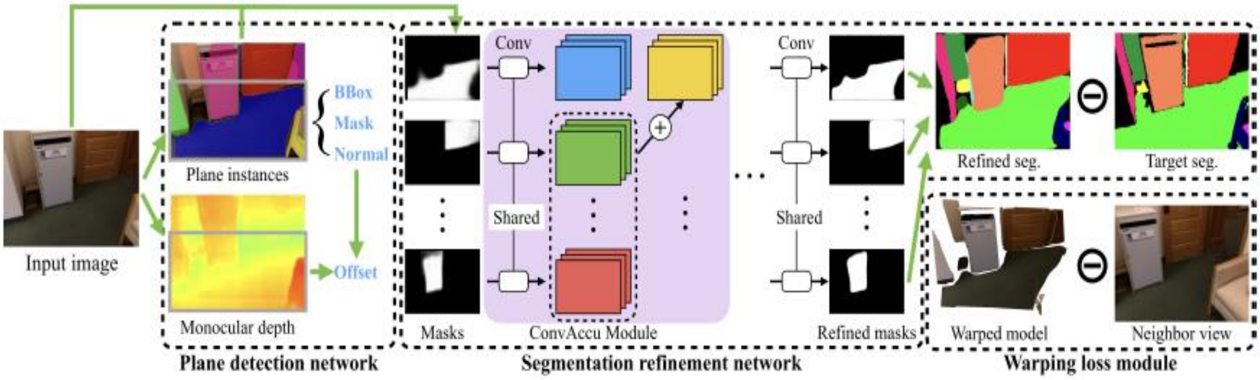


Figure 4.1: Plane R-CNN model framework.

In PlanNet and PlaneRecover, the concept of plane segmentation task was introduced through Convolutional Neural Networks (CNNs); however, the segmentation was in general poor (1) limited to fixed number of planes about 10 plane per view and misses small surfaces. These two main drawbacks were addressed in Plane R-CNN in sophisticated architectures (figure 4.1) that consists of:

a. Plane Detection Network

In the first block, the network segments the scene into planar and non-planar regions in geometric sense using the original Mask R-CNN [20] instance segmentation network. The network jointly predicts the normal and depth of the planar regions where the non-planar regions are represented by their depth only.

The normal is predicted by regressing the residuals of the closest anchor norm (7 fundamental norms) using a L1 cross entropy loss that replaces boding boxes anchors in the original implementation of Mask R-CNN.

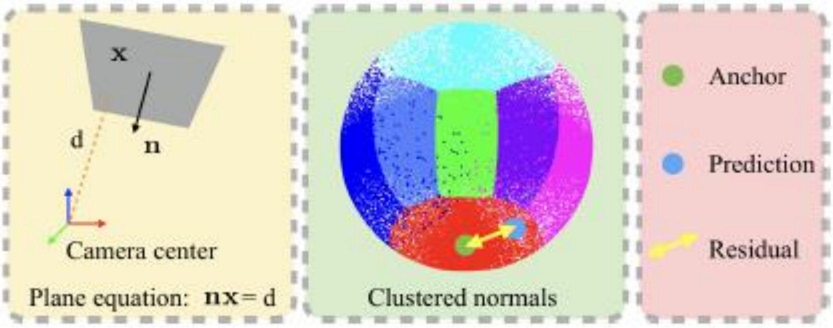


Figure 4.2: Seven (7) fundamental normal anchors.

At the end of Feature Pyramid Network (FPN) [21]; a decoder is used to estimate the pixel wise depth information. The plane offset d is estimated as follows:

$$d = \frac{\sum_i \; m_i(n^T(z_i\, K^{-1}x_i))}{\sum_i \; m_i} \tag{4.1}$$

Where K is the intrinsic parameters, x_i is the pixel coordinates, z_i is the predicted depth, n is the norm of the plane and m_i indicator variable.

b. Mask Refinement Network

The refinement aims to refine jointly all planes masks and maximize the number of detected planes. For this U-Net architecture [22] is proposed, to compare the plane masks against the ground truth planes with cross entropy loss whenever they overlap. As consequence; the depth map and plane normal are also adapted accordingly to the refined the mask.

c. Warping Loss module

The refined planes parameters are adapted between the nearby views using 3D point maps in minimizing the re-projection error L2 distance norm. This differentiable module enforces the geometrical consistency in depth estimation network and boosts the global plane detection accuracy.

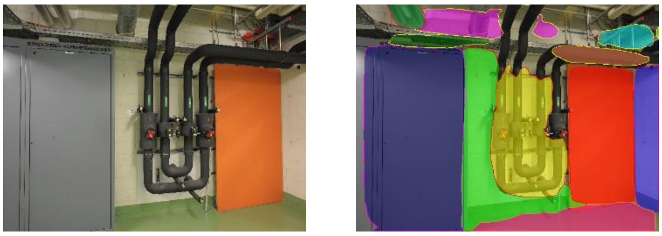
The plane R-CNN was trained in Scannet indoor dataset [23]with the generated plane ground truth. Plane R-CNN outperforms other plane detectors in terms of accuracy and recall in indoor and outdoor scenes. For this reasons; we adopt the Plane RCNN as segmentation network for planar region.

2.2. Plane R-CNN on ETH 3D

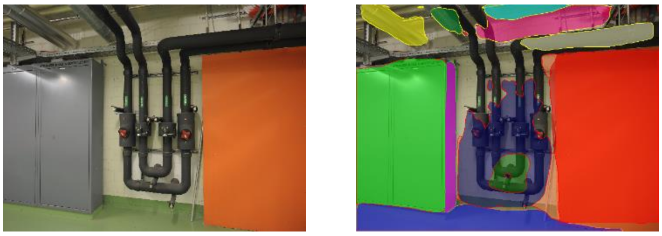
For demonstration purpose; we choose two high resolution indoor and outdoor scenes from ETH 3D dataset to evaluate the performance of Plane R-CNN, we chose multiple neighbouring views per scenes:

- *Indoor views (Pipes Scene):*

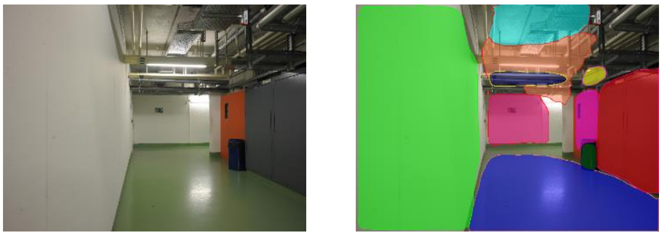
0634



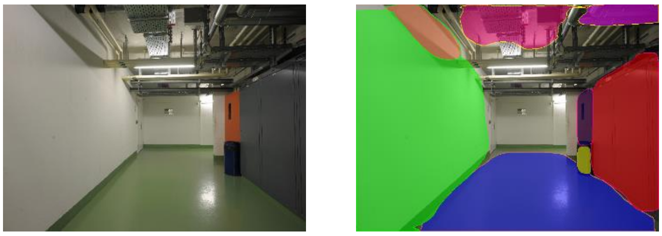
0636



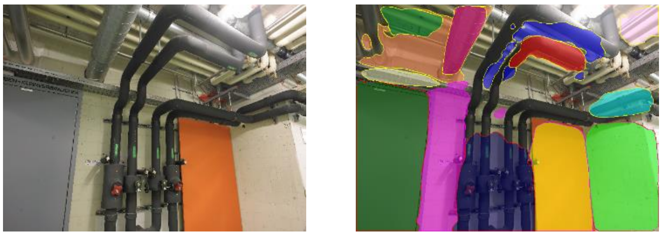
0639



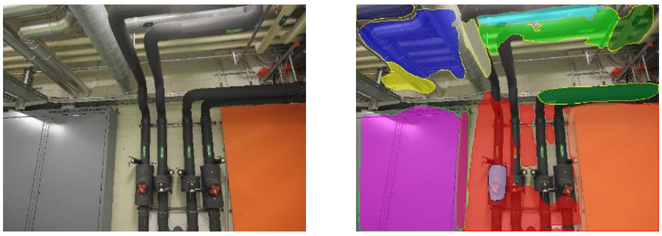
0640



0643



0644



0646

0647

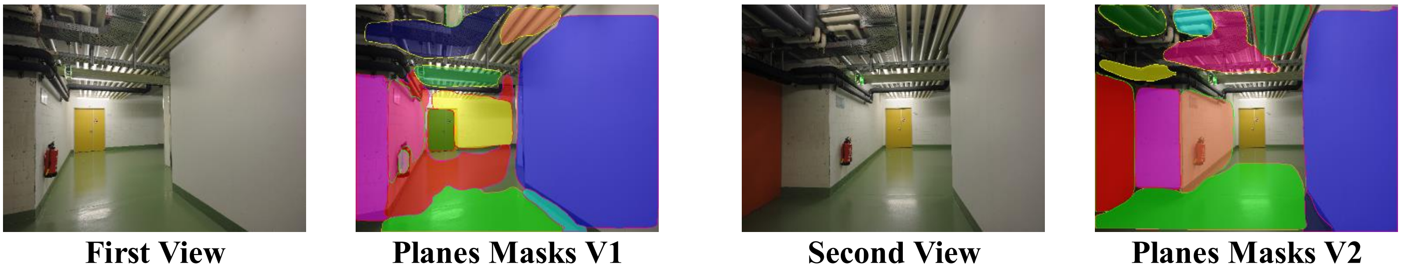


Figure 4.3(a): Plane Mask for neighbouring views in *pipes* scene
left to right: first view, plane masks v1, neighbouring view, plane masks of neighbouring view.

- *outdoor views (electro scene):*

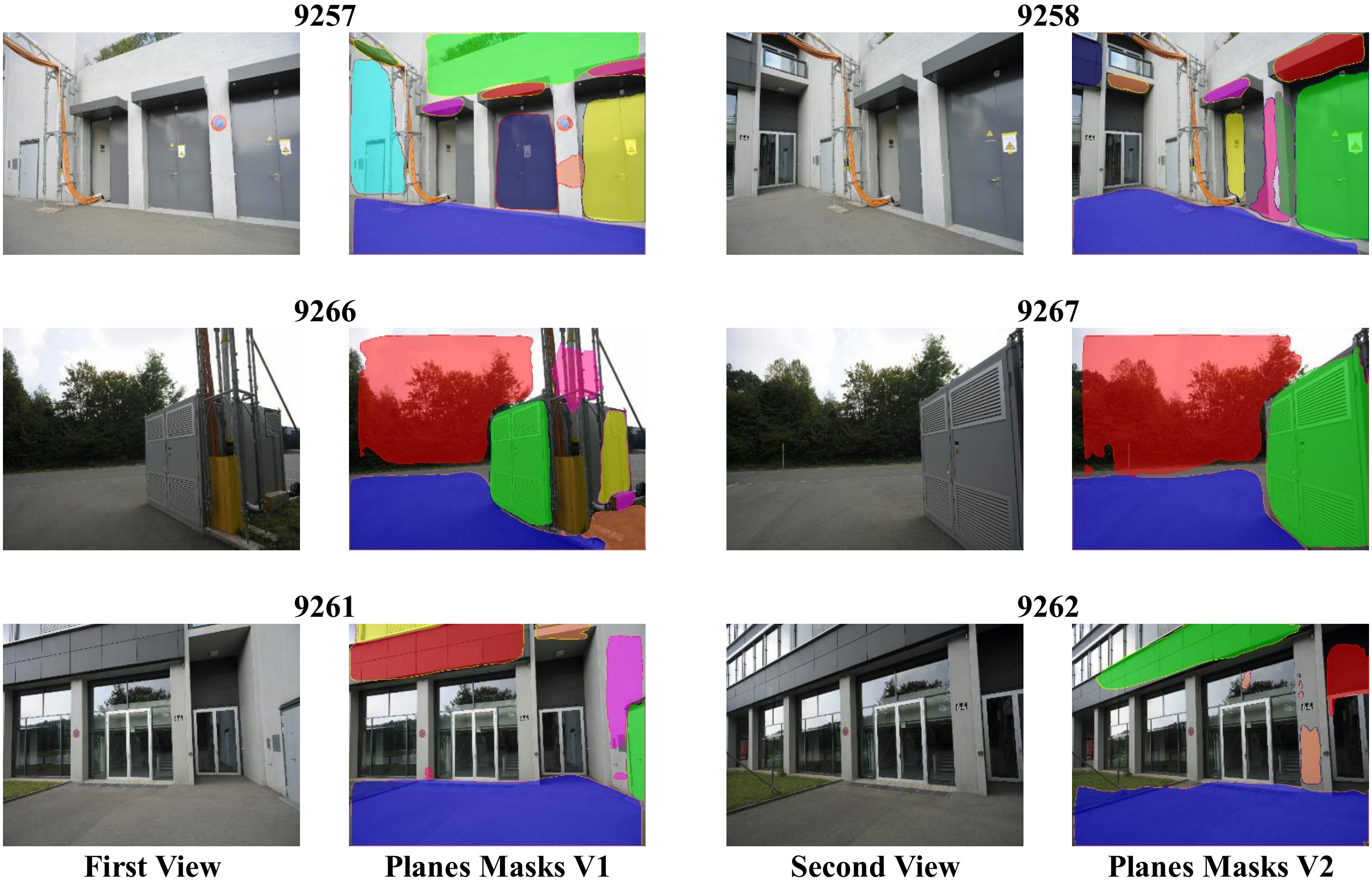


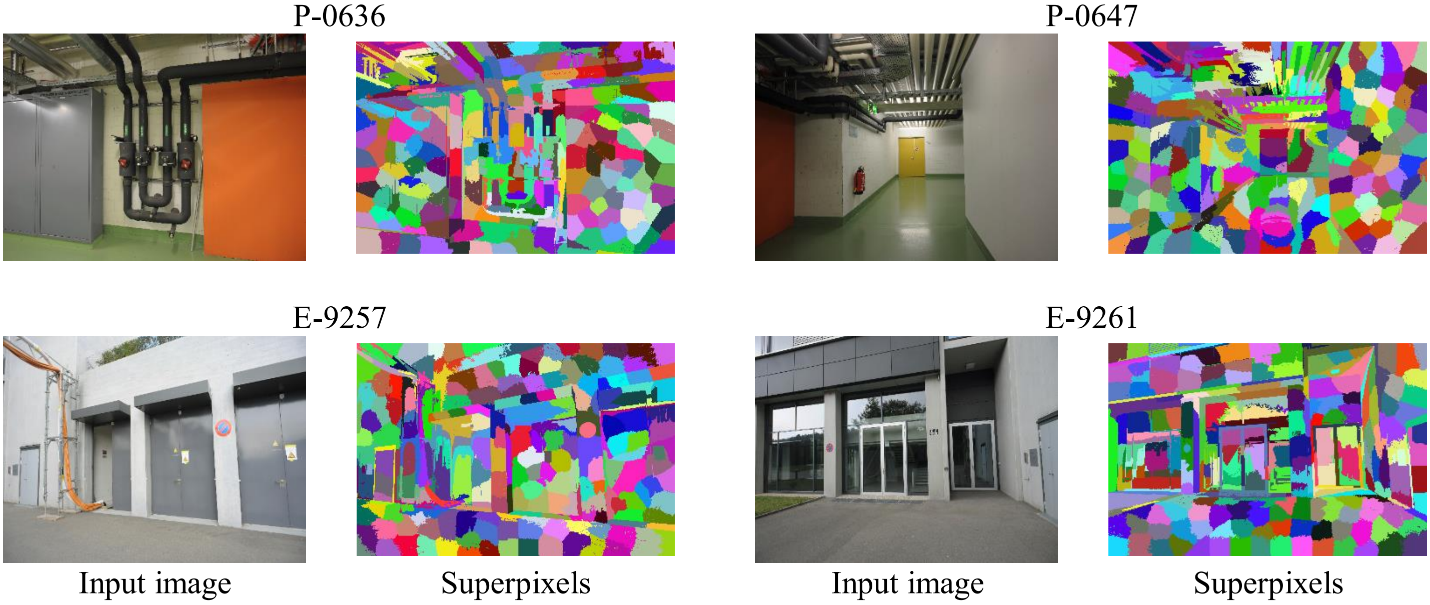
Figure 4.3(b): Plane Mask for neighbouring views in *Electro* scene
left to right: first view, plane masks v1, neighbouring view, plane masks of neighbouring view.

In general; Plane RCNN successfully detects the most dominant planes and textureless surfaces in indoor and outdoor scenes mainly side walls, floor, doors and flat surface as the emergency and electricity boxes in view P-0636 and E-9266 respectively.

The network detects the closest surfaces better than the farther ones and sometimes it fails to detect it as in P-0646 and P-0647; where the exit door at the end of the corridor was not detected. However the network totally fails to detect all windows appearing in E-9261 and E-9261 and sometimes wrong segmentation appears between two neighbouring textureless surfaces as in P-0640 and P-0646; where a part of side wall is included with the floor mask (poor recall to borders).

2.3. Low-Level feature segmentation

We use superpixel segmentation algorithm to describe a group of pixels which are perceptually similar forming clusters. Superpixel are the basics shapes that forms smooth and large surfaces, commonly superpixel are built around an energy function that assigns each pixel to superpixel centres on their similarity appearance or other pixel wise information (depth, normal). Among the large list of developed superpixel algorithm, we adopt the SEED superpixel [24] for two main factors (1) number of generated superpixel are controllable, (2) compactness and better boundary recall which is important in plane segmentation, refer to superpixel evaluation [25] for more details.


P-0636

P-0647

E-9257

E-9261

Input image

Superpixels

Input image

Superpixels

Figure 4.4: SEED superpixels segmentation in *Pipes & Electro* scenes

Although, the superpixel based on similarity appearance over segments the scene; however, it deals perfectly with different edges and shapes boundaries unlike the plane mask. Therefore combination between the low level feature segmentation (superpixel) and high level plane (primitive) segmentation will improve the plane detection part and parameter estimation.

2.4. Merging Algorithm

In superpixel merging; we combine superpixels shapes to construct planes and reduce the number of cluster; non planar region should remain unmerged. We propose two merging stages based on (1) similarity appearance; as in [5]; and (2) on plane masks based on Plane R-CNN detections:

a. Plane Primitives Merging

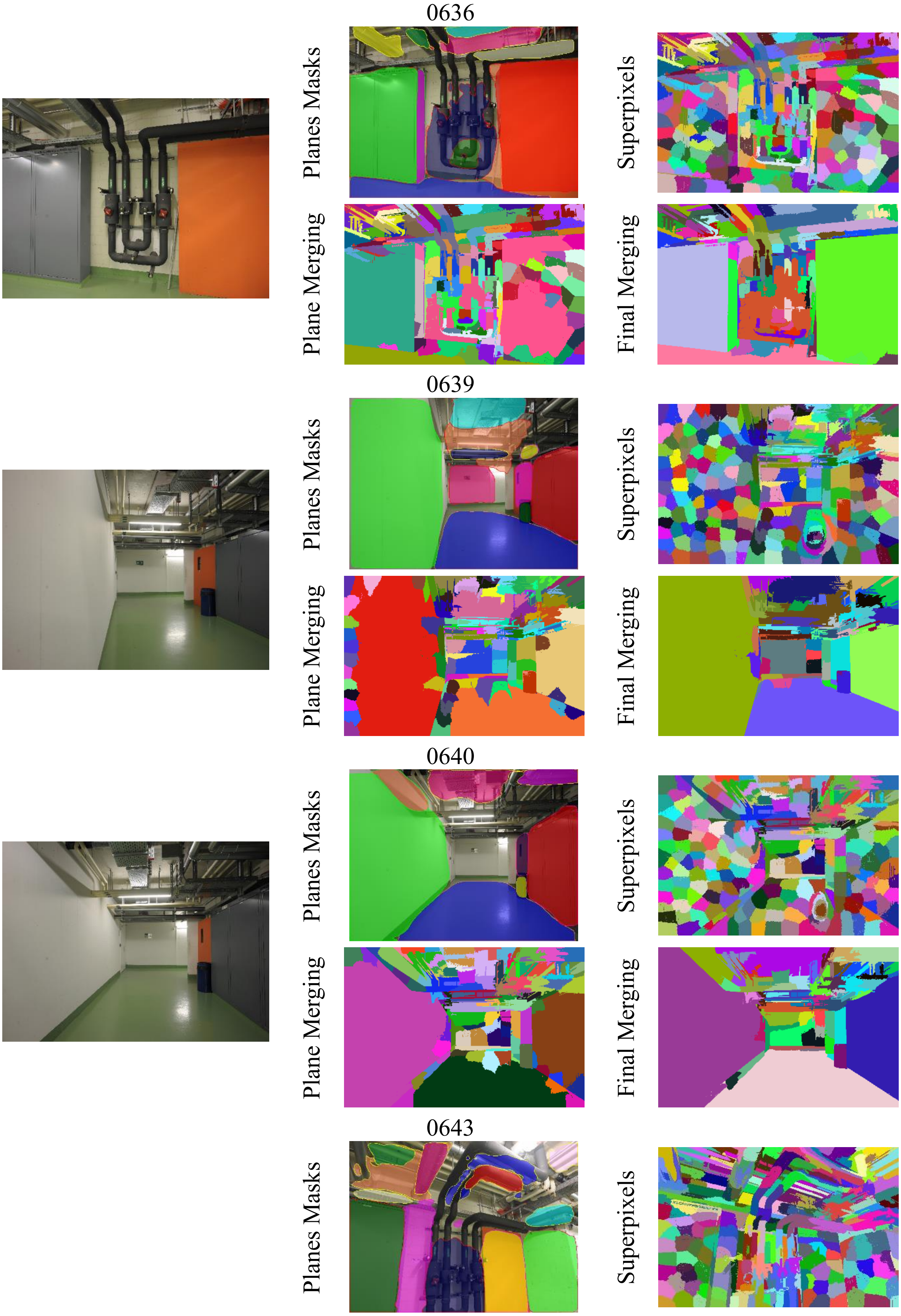
On top of superpixel; we use Plane R-CNN detected masks to group superpixels that overlaps with planes as first merging, we perform this merging operation carefully under the following conditions:

- A superpixel is merged to a plane only if most of its pixels overlaps with plane, otherwise the superpixel remains without merging, therefore we define overlapping coefficient $\delta = 75\%$.
- Shared superpixel between two planes masks or more is assigned to the most dominant plane, otherwise it will not be merger with any.

b. Appearance Merging

We use the fact that superpixel on textureless surfaces are smooth and have same colour distribution; as further merging process, we pick the most similar superpixel in neighbouring to initial plane-superpixel cluster then we merge them forming a new plane-superpixel.

With these two merging mechanisms; we avoid wrong plane segmentation in including non-planar regions to a real plane that will lead to wrong depth estimation. We notice also that wrong plane detections disappear after the merging process. To provide depth and normal information to the missing pixel, we perform robust plane estimation in each cluster.



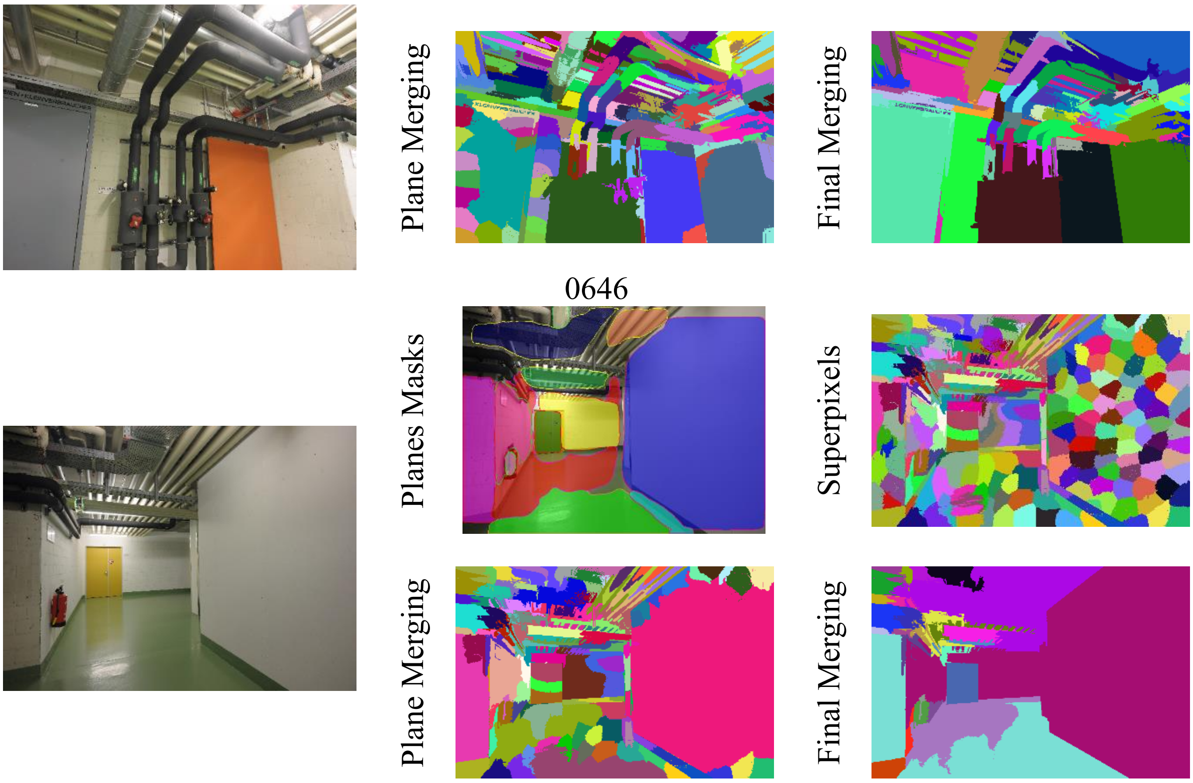
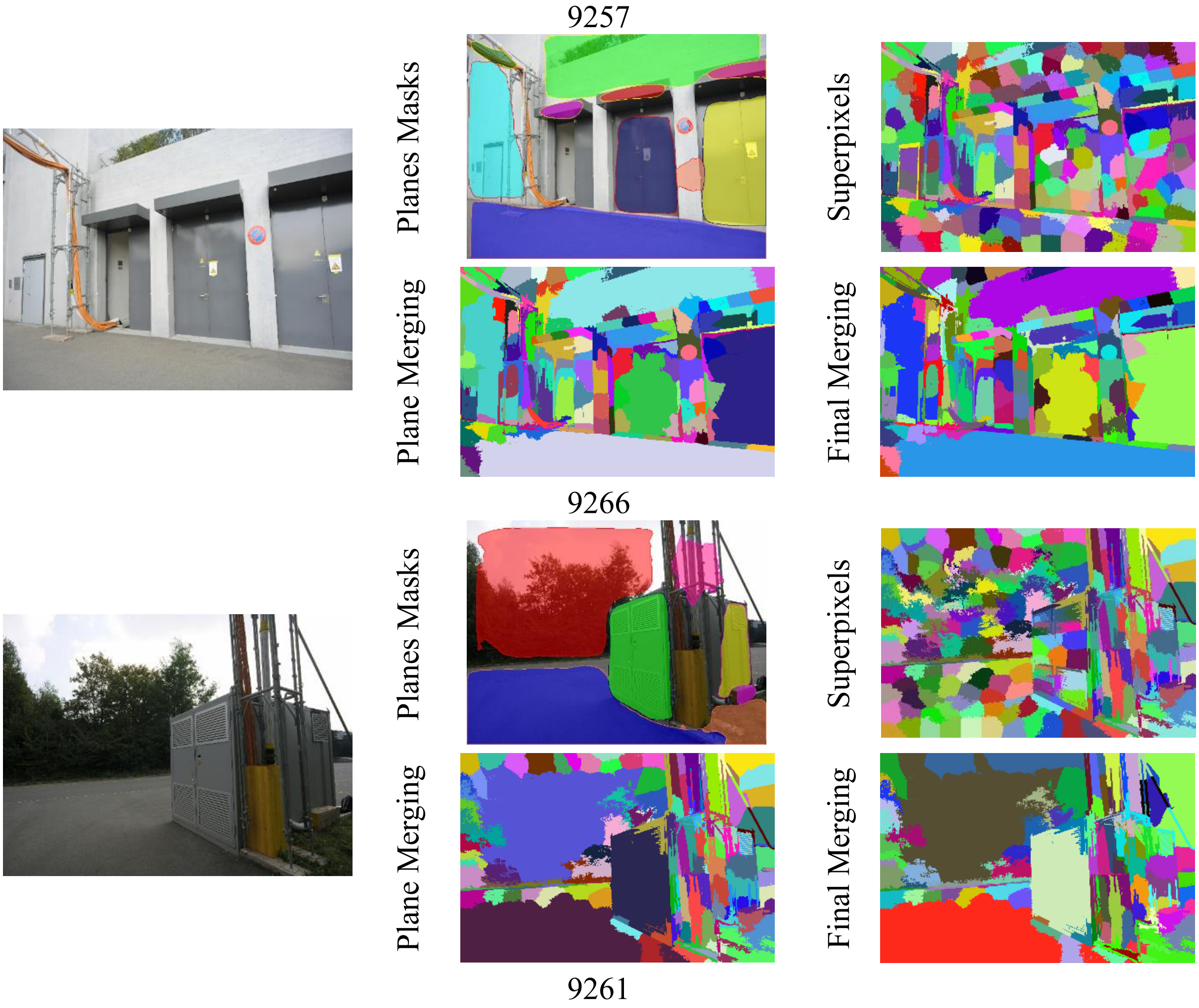


Figure 4.5(a): Plane-Superpixel merging in *Pipes* scene.



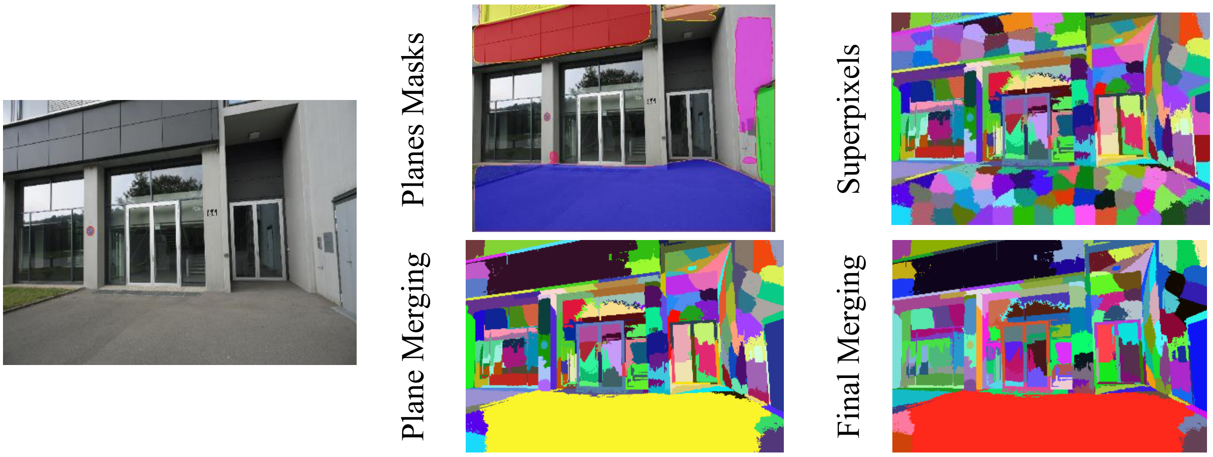


Figure 4.5(b): Plane-Superpixel merging in *Electro* scene.

2.5. Plane Parameter Estimation

In 3d dimension; a plane can be defined geometrically using the following plane equation:

$$n_x\, x + n_y y + n_z\, z + d = 0 \tag{4.2}$$

Where $n = (n_x,\ n_y,\ n_z)^T$ is the plane normal vector and d is the offset or the distance from the origin. As there are only three 3 degrees of freedom for a plane, the length of normal vector must be normalized to unity $\|n\| = 1$ as constraint. A set of 3D point clouds will be used to estimate the plane parameters, then the distance of any point $p = (\,p_x,\ p_y,\ p_z\,)$ from the plane is given by $dist = n^T.p - d$ provided that $n^T.n = 1$.

We use RANSAC to search for best plane estimate in all superpixel clusters using local 3D point clouds. As described in section 2.3 – Chapter two, we select randomly three 3 points and we calculates the parameters of the corresponding plane; then we calculate the number of inliers according to the current plane estimate model *confidence score*”. The best plane fitting is obtained after at most N iterations and with high confidence.

Algorithm 1, details the pseudocode of the RANSAC for plane estimation. Three 3 points are randomly chooses from 3D point cloud $3DPoints$ to calculate the norm and offset $odel$; we set the threshold distance to $d_t = 10\,cm$. We look for all superpixel point clouds that in line with plane $Model$ with $abs(dist)$ less than the defined threshold d_t. The $bestSupport$ aims to remove the 3D point outliers and maximize the set of inliers s, in addition; σ search for the best 3D points that minimizes the total standard deviation within the set s. The process is repeated N Times and the best with highest confidence score.

Table 1: Robust plane fitting using RANSAC.

Algorithm 1: *RANSAC for Plane estimation*

```
Inputs: {3D point cloud, plane-superpixel segmentation}
Outputs: {Plane parameters estimates}
    $bestSupport = 0$    $bestStd = \infty$    $bestModel = \begin{Bmatrix} n = (0,0,0)^T \\ d = 0 \\ q = 0 \end{Bmatrix}$
    $N = round\ (k)\ \ with\quad \begin{matrix} p = 0.99\ (confidence) \\ s = 3\ (numPoints) \\ inlierRatio = numInliers/numPoints \end{matrix}$
    while $i \le N$ do
        $vect3D = rand\ (3DPoints)$ sample 3 point randomly
        $Model = pt2plane(vect3D)$
        $dist = dist2plane(Model\,, 3DPoinrs)$
        $s = find\ (abs(dist) < d_t)$
        $\sigma = std(s)$
        if $(\ length(s) > bestSupport\ or\ \sigma < bestStd\ )$ then
            $bestSupport = length(s)$   $bestModel = planeParam$   $bestStd = \sigma$
        end if
        $i = i + 1$
    end while
```

2.6. Completion

In our method, depth completion is achieved using the fitted planes in previous step; isolated superpixels that have no 3D point clouds has not been merged or included in plane estimation. We distinguish three 3 pixel wise depth information in the input depth map provided by dense reconstruction stage:

- *Missing pixels*: appearing with zero depth values; are either filtered in geometric consistency check or unmatched pixel in the correspondence.
- *Outliers*: or spikes; appearing as high depth values managed to survive in the consistency check and unconsidered in plane fitting.
- *Valid pixels*: are geometrically consistent between N views and provide a good pixel depth estimate used for plane fitting.

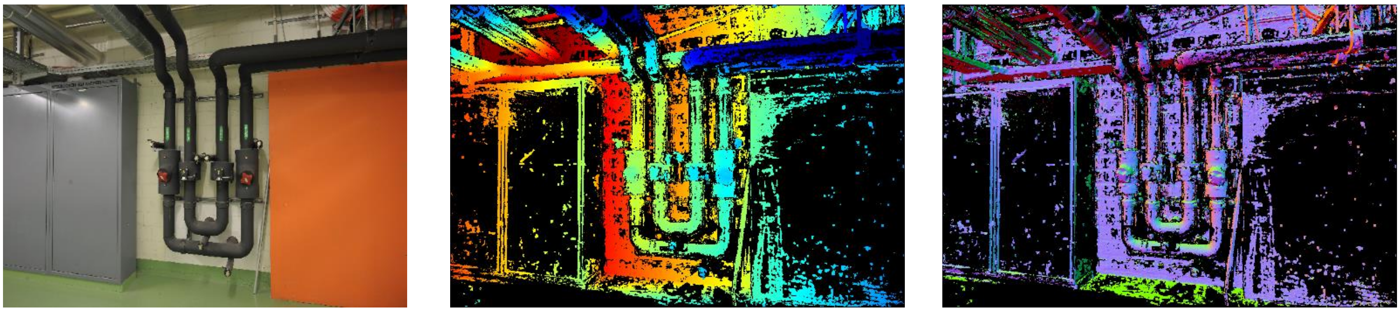
In our depth Completion process; we change outliers' depth values and we fill the missing pixels with their estimate, using equation (4.1), the pixel wise depth is calculated as:

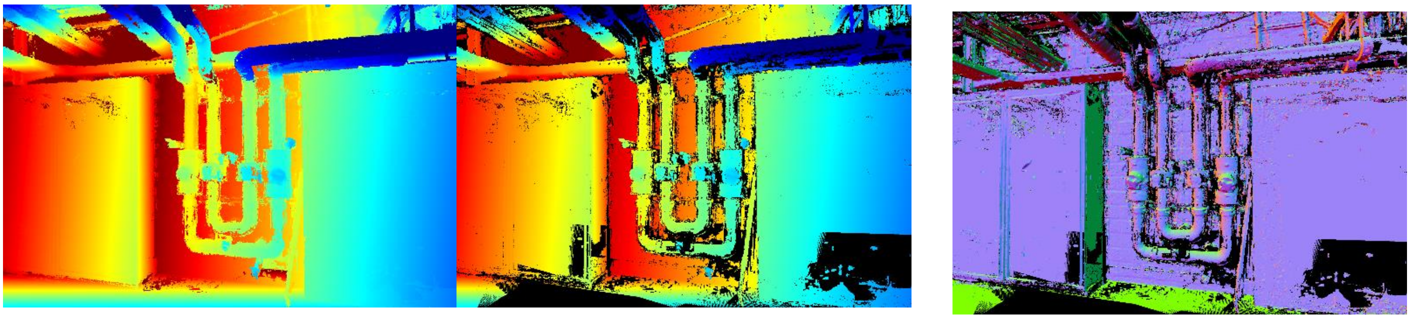
$$z_i = \frac{d}{n^T.X} = \frac{d}{n^T.(K^{-1}x_i)} \tag{4.3}$$

The incomplete pixels are set to high numerical values $z_i = \infty$; accordingly, we complete the normal map for the correspondingly pixels $n_i = (n_x, n_y\,, n_z)^T$. We integrate mutli-view geometric consistency check as forward-backward reprojection error between a source and reference image; similar to [2], we give reprojection error $e_s^r = \|\, x_r - H_r^s\, H_r\, x_r \|$; where H_r^s is the projective backward transformation from source to reference view.

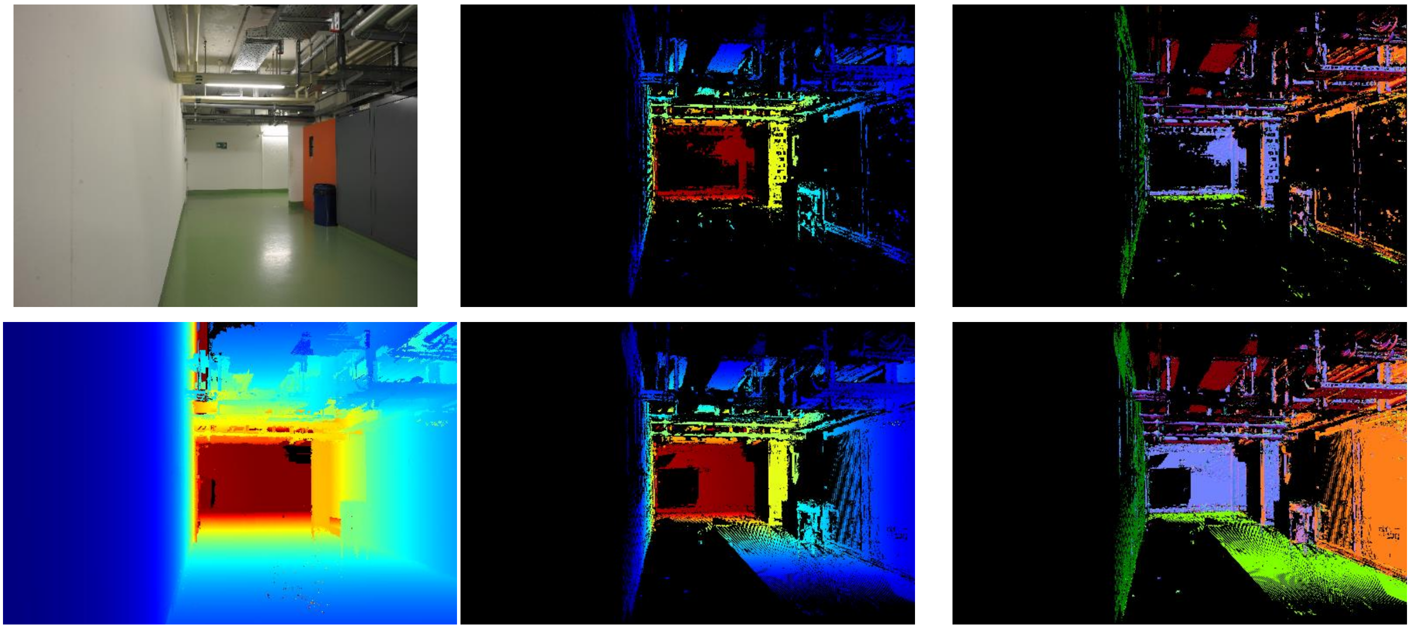
The depth value in reference frame x_r is wrapped to the source frame at $x_r^s = H_r\, x_r$ where $H_r = K_s\,[R \quad T]\, K_r^{-1}$ as describe in section 2.2.b chapter two. The estimated depths and normal are consistent if the reprojection error is small $e_s^r \leq 1\, px$.

0636

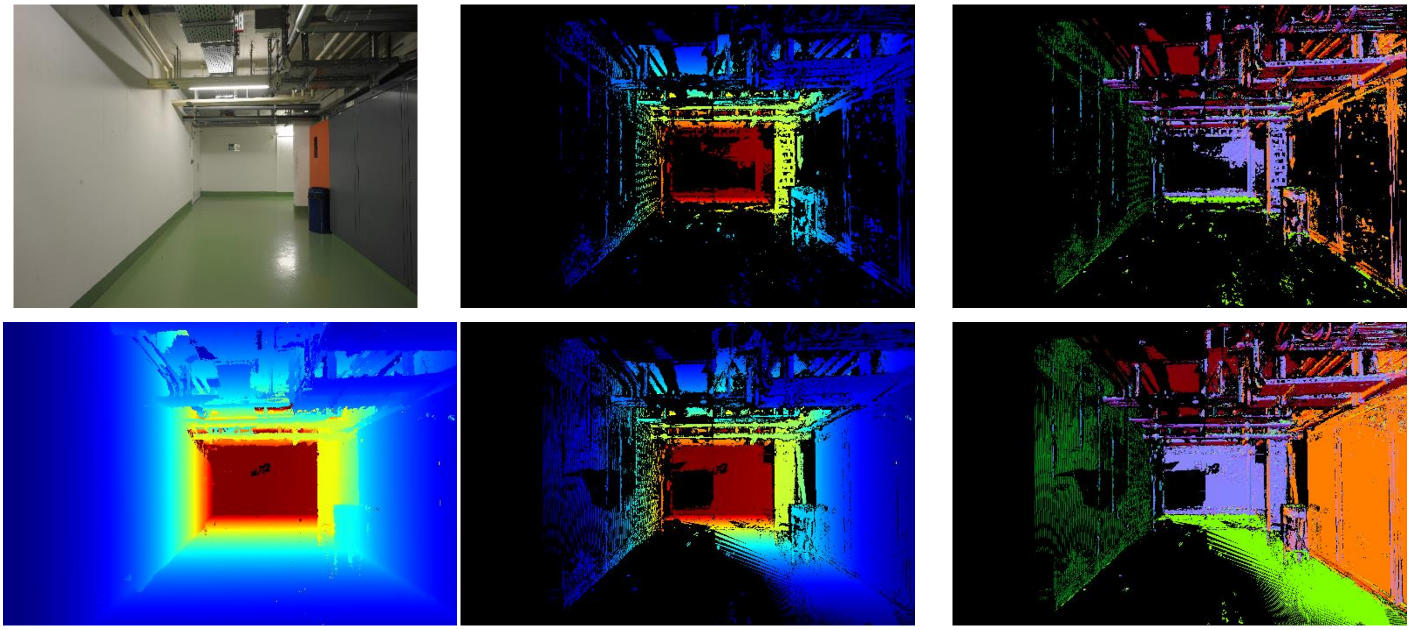




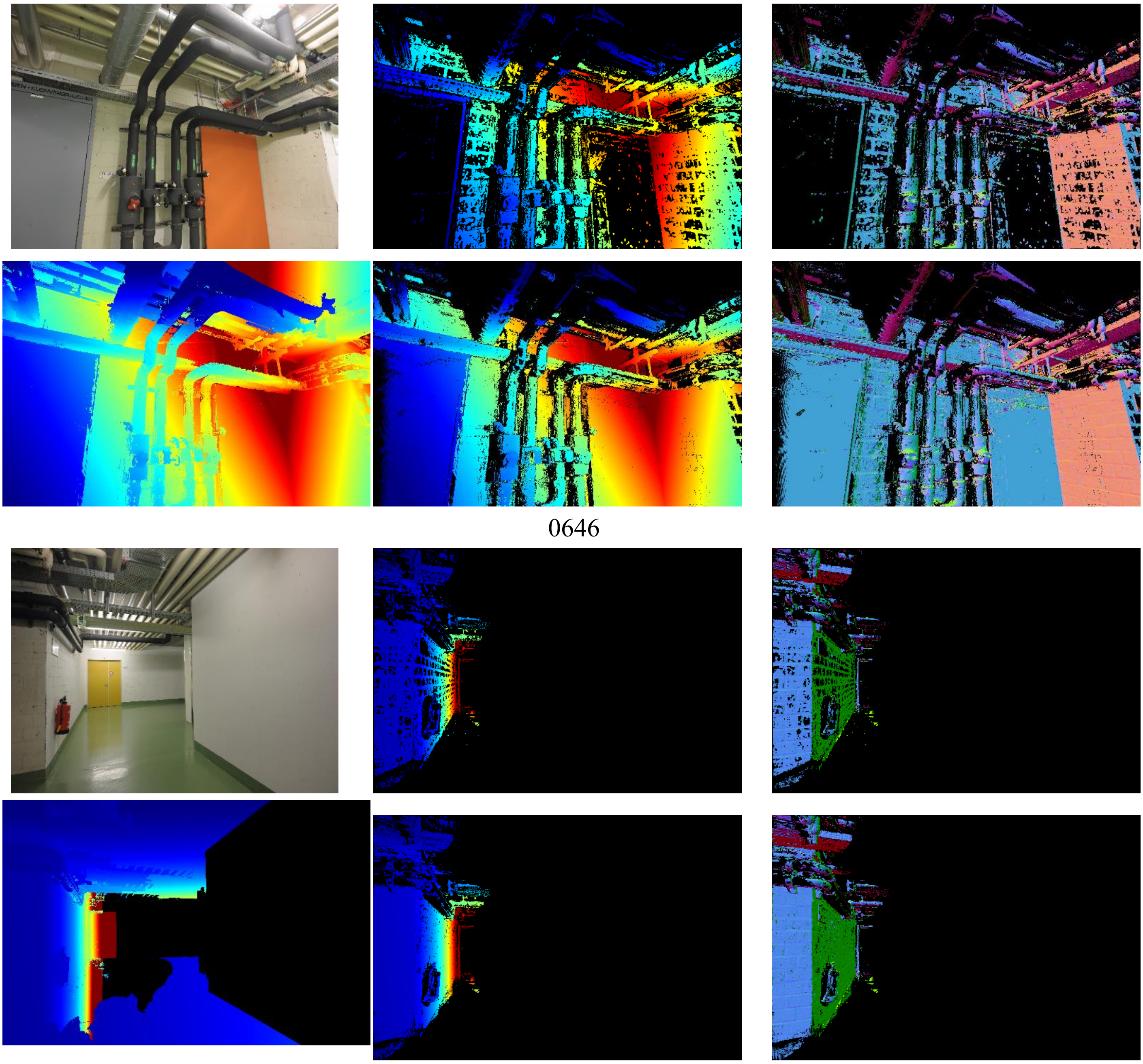
0639



0640



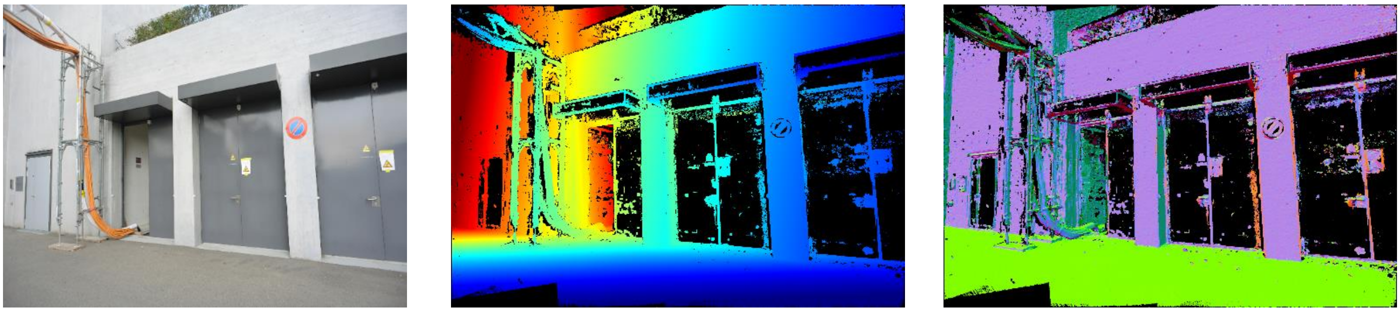
0643



0646

Figure 4.6(a): Completed Depth and Norma maps in *Pipes* scene.
Top-left: RGB Image, Top-middle: Input depth map, Top-right: input normal map
Bottom-left: Completed depth, Bottom -middle: Filtered depth , Bottom –right: Filtered normal

9257



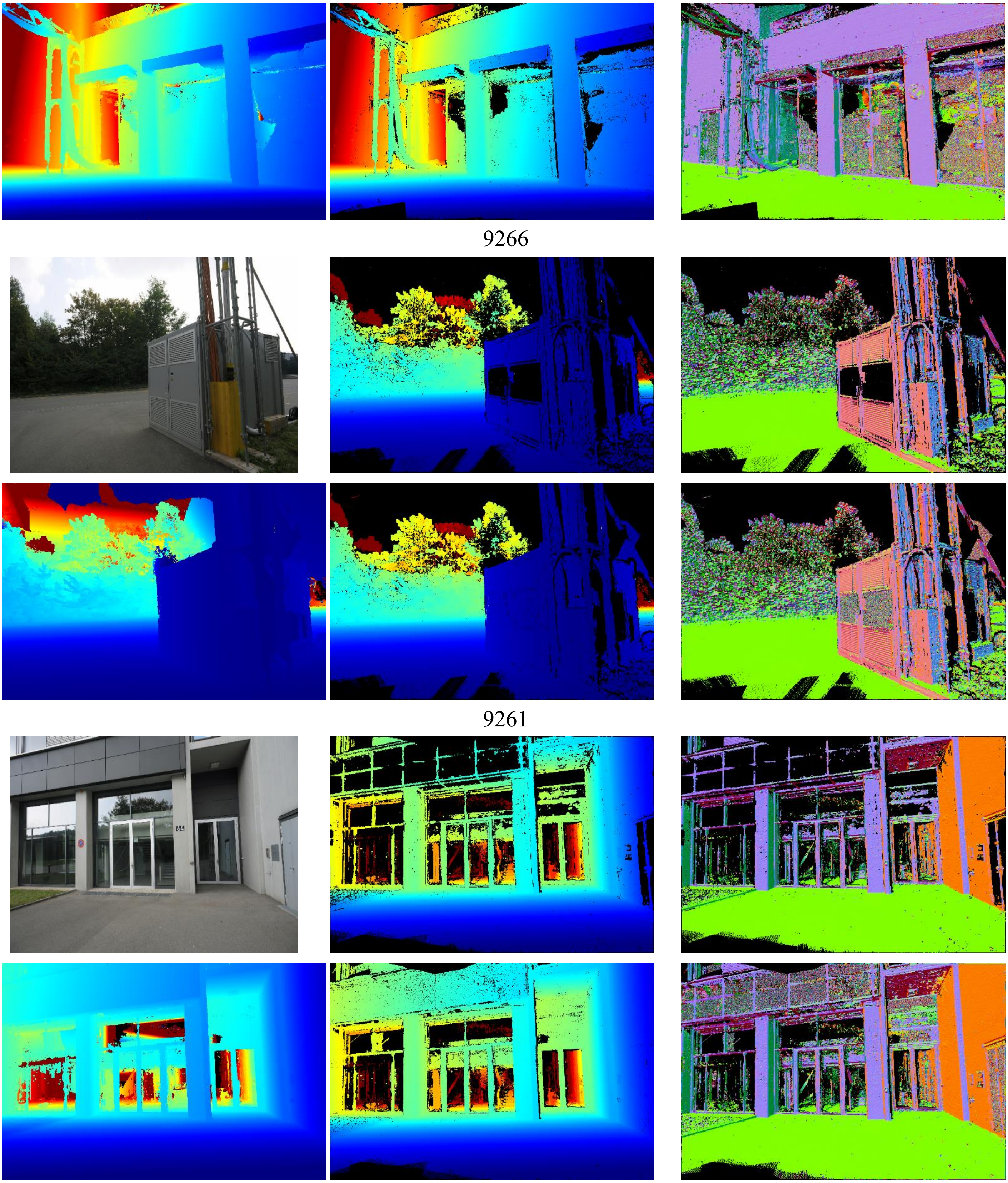


Figure 4.6(b): Completed Depth and Norma maps in *Electro* scene.
Top-left: RGB Image, Top-middle: Input depth map, Top-right: input normal map
Bottom-left: Completed depth, Bottom -middle: Filtered depth , Bottom –right: Filtered normal

In order to evaluate our work; we fuse the depth normal map of each scene as point clouds, the 3D model of pipes and electro scenes are show in figure 4.7. The generated 3D point clouds of each scene are evaluated against its ground truth provided in ETH 3D Training dataset.

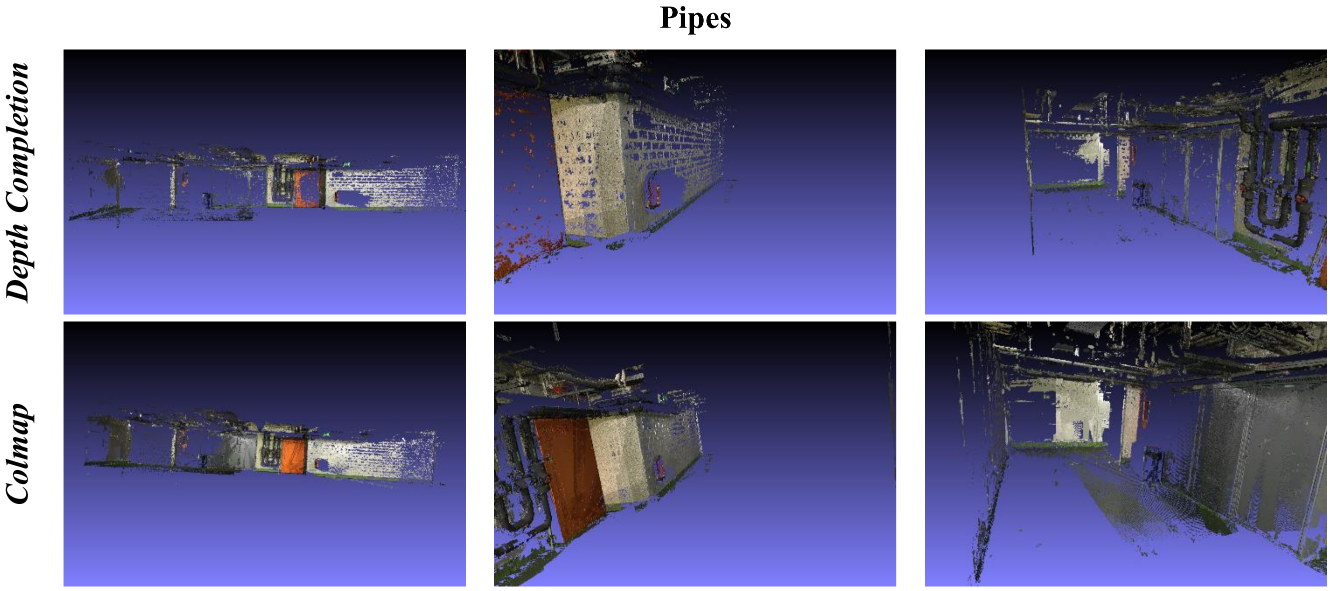


Figure 4.7(a): *Pipes* Fused 3D model (Top) Plane Depth Completion (Bottom) Colmap.

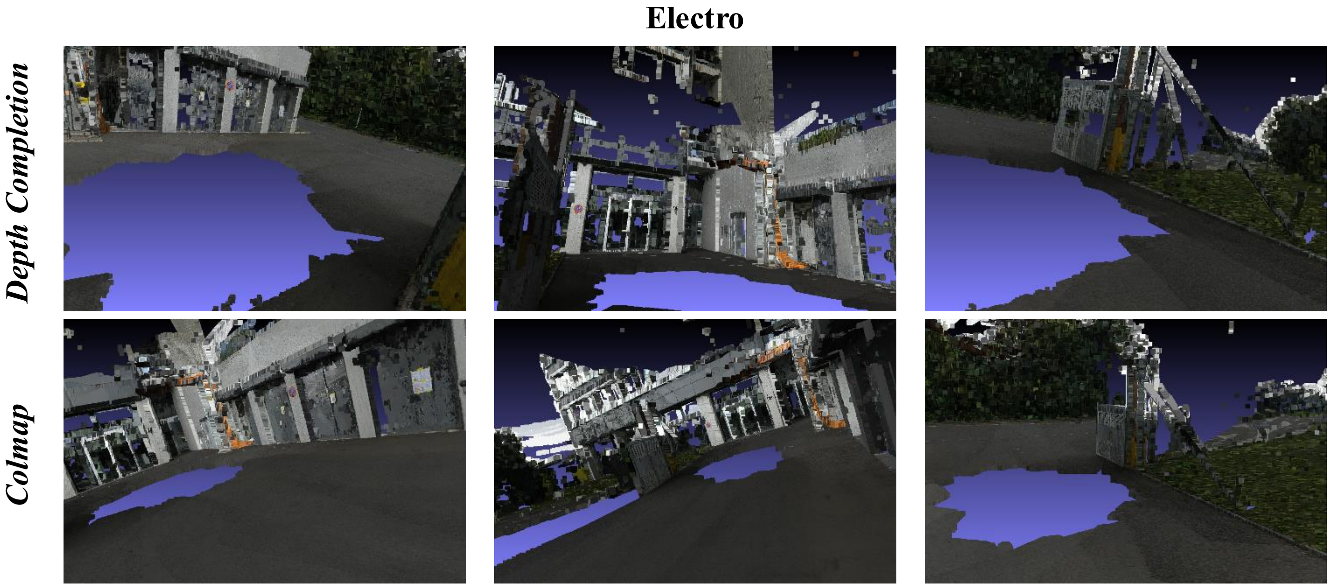


Figure 4.7(b): *Electro* Fused 3D model (Top) Plane Depth Completion (Bottom) Colmap.

Chapter 5

V. Evaluation

We perform evaluation on the ETH 3D high resolution training dataset in terms of accuracy and completeness as described [26]. The dataset contains 13 indoor/outdoor scenes recorded using DSLR camera and the ground truth geometry has been obtained using high-precision laser scanner. We generate the sparse depth maps using COLMAP [27] in machine of 32 core 2.4 GHz with 16 GPUs Nvidia Tesla K80 of 11 GB memory.

For better control, we use SEED to generate 200 superpixels in each view and we require 75% as minimum plane overlapping in the first stage of merging. For each cluster, we estimate the plane parameters with 30% inlier ratio and 0.99 confidence. We consider at least two 2 views that has to be geometrically consistent with 1 pixels in projection and reprojection error.

In the 3d fusion stage, we require a maximum reprojection error of 2 pixels and two 2 points at least for good scene representation quality with maximum depth error of 0.1 and 25 maximum normal error.

Table 5.1.a and 5.2.b shows the evaluation results of our Plane-Superpixel depth Completion process in Pipes and Electro scenes. For each scene; we compare the MVS reconstruction as point clouds against the laser scan ground truth; we evaluate Colmap and our Plane-Superpixel depth Completion in terms of accuracy and completeness in range of distances thresholds r from 1 cm to 50 cm.

We define the *accuracy (a)* as a fraction of all points within a distance radius r of the ground truth point and *completeness (c)* as the amount of the ground truth points with the radius. The F1 score is a global reconstruction score defined as harmonic mean $F_1 = 2 \times (a.c)/(a + c)$.

Table 5.1(a): Plane Depth Completion reconstruction evaluation in terms of completeness and accuracy against Colmap in *Pipes* scene.

Pipes							
Tolerances (m)		**0.01**	**0.02**	**0.05**	**0.1**	**0.2**	**0.5**
Completeness	*Colmap*	0.252746	0.33798	0.465243	0.558864	0.650544	0.750019
	Depth Completion	**0.409780**	**0.490407**	**0.577670**	**0.628898**	**0.679316**	**0.750330**
Accuracies	*Colmap*	**0.934714**	**0.969448**	**0.983854**	**0.988988**	**0.993955**	0.99881
	Depth Completion	0.915554	0.958032	0.98090	0.987324	0.993399	**0.999079**
F1-Scores	*Colmap*	0.3979	0.501219	0.631747	0.714164	0.786394	0.856718
	Depth Completion	**0.56616**	**0.648734**	**0.727124**	**0.768368**	**0.80687**	**0.85702**

Table 5.1(b): Plane Depth Completion reconstruction evaluation in terms of completeness and accuracy against Colmap in *Electro* scene.

Electro							
Tolerances (m)		**0.01**	**0.02**	**0.05**	**0.1**	**0.2**	**0.5**
Completeness	*Colmap*	0.570057	0.695369	0.804491	0.872414	0.923713	0.968071
	Depth Completion	**0.664289**	**0.791398**	**0.880407**	**0.923921**	**0.947172**	**0.971636**

Accuracies	*Colmap*	0.813764	**0.910184**	**0.973879**	**0.988934**	**0.993558**	**0.996368**
	Depth Completion	**0.815456**	0.906377	0.968492	0.985663	0.991788	0.995604
F1-Scores	*Colmap*	0.670451	0.788406	0.881118	0.927027	0.957363	0.982016
	Depth Completion	**0.732151**	**0.844995**	**0.922351**	**0.953793**	**0.968967**	**0.983474**

We have achieved a better a better 3D reconstruction in terms of completion and accuracy in indoor/outdoor high resolution images in comparison to Colmap, we provide also in the Table 5.2, evaluation for all 13 Scenes in terms of F1 Score only.

In general; depth completion based on Plane-superpixel segmentation provides a high scene completion with a good accuracy; we notice that our algorithm provides better scene completion in indoor scene as in Pipes, Office and Kicker, and satisfactory performance in outdoor scenes for the following reasons:

- Indoor Scenes; our depth completion algorithm based on Plane-Superpixel segmentation successfully provides depth information for missing pixels in large textureless areas which are dominant and numerous.

- The number of views per scene also play a role in performance of both algorithm; for large number of views a slight difference in performance between colmap and our depth completion algorithm (redundancy) as in courtyard (38 views) and Delivery area (44 views).

- The large presence of sky and windows lead to unsatisfactory depth completion due the fact that the sky is also detected as plane by the network, where the reflection in windows also prevent the network to extract its plane mask. The network has been trained on Scannet dataset where only few scenes includes large windows and sky.

- Since our plane estimation is based on 3D point clouds, plane segments with no sufficient points; as the left side wall in Pipes 0646; are not included in the depth completion.

Table 5.2: 3D reconstruction evaluation in terms F1-score for all ETH 3D high resolution dataset comparing depth completion against colmap.

		Tolerances (m)					
		0.01	**0.02**	**0.05**	**0.1**	**0.2**	**0.5**
Courtyard	*Colmap*	**0.629295**	0.819978	0.90647	**0.942959**	**0.957633**	**0.97004**
	Depth Completion	0.615622	**0.820336**	**0.906509**	0.940758	0.953561	0.968998
Delivery_area	*Colmap*	0.667562	0.802857	0.900584	0.939063	0.965004	**0.986292**
	Depth Completion	**0.692918**	**0.831981**	**0.925545**	**0.953765**	**0.970637**	0.985557
Electro	*Colmap*	0.670451	0.788406	0.881118	0.927027	0.957363	0.982016
	Depth Completion	**0.732151**	**0.844995**	**0.922351**	**0.953793**	**0.968967**	**0.983474**
Facade	*Colmap*	**0.468573**	**0.661183**	**0.835533**	**0.887234**	0.914967	**0.940241**
	Depth Completion	0.456521	0.645729	0.828742	0.884052	**0.914696**	0.934206
Kicker	*Colmap*	0.539274	0.636944	0.761775	0.859078	0.944242	0.990958
	Depth Completion	**0.72988**	**0.816431**	**0.893462**	**0.93801**	**0.970412**	**0.992623**
Meadow	*Colmap*	0.374174	0.530808	0.674868	0.754573	**0.827516**	**0.904809**
	Depth Completion	**0.464044**	**0.589917**	**0.708272**	**0.76926**	0.825841	0.894436
Office	*Colmap*	0.427731	0.528466	0.66375	0.761791	0.850255	0.951334
	Depth Completion	**0.600556**	**0.689161**	**0.782626**	**0.84154**	**0.894904**	**0.960659**
Pipes	*Colmap*	0.3979	0.501219	0.631747	0.714164	0.786394	0.856718
	Depth Completion	**0.56616**	**0.648734**	**0.727124**	**0.768368**	**0.80687**	**0.85702**
Play_Ground	*Colmap*	0.521271	0.682413	0.845518	0.91095	0.945754	0.973094
	Depth Completion	**0.53022**	**0.692895**	**0.856412**	**0.921031**	**0.95105**	**0.974347**
Relief	*Colmap*	0.704322	0.804553	0.880108	0.921156	0.953362	0.979029
	Depth Completion	**0.723707**	**0.825141**	**0.897729**	**0.932395**	**0.957934**	**0.975867**
Relief_2	*Colmap*	0.676665	0.788244	0.874211	0.91684	**0.949031**	**0.974282**
	Depth Completion	**0.688115**	**0.803549**	**0.88559**	**0.921755**	0.948532	0.970945
Terrace	*Colmap*	0.760262	0.861278	0.937537	0.970564	0.988105	**0.996787**
	Depth Completion	**0.779455**	**0.876898**	**0.94359**	**0.973467**	**0.988118**	0.996195
Terrains	*Colmap*	0.695597	0.787254	0.880995	0.935029	0.973722	**0.994064**
	Depth Completion	**0.805346**	**0.886269**	**0.943516**	**0.968479**	**0.983312**	0.993836

VI. Conclusion

We have presented a plane segmentation and depth completion for high resolution images, our algorithm; includes low level feature segmentation and plane primitives mask to segment planes present in the scene. We perform robust estimation to estimate the plane parameters using the 3D point cloud in each plane segment, then; we complete the missing pixels in the input depth maps with their estimates.

We have shown a significant improvement in evaluating our depth completion reconstruction approach in terms of completeness on the ETH 3D dataset benchmark while keeping high level of accuracy. Further improvements can be done in different level in the pipeline:

- Further training to Plane R-CNN network for indoor and outdoor scenes; will enhance the plane detection for specific planes as windows.

- Multi-view Plane-Superpixel segmentation and refinement (merging) is needed to propagate planes between views (more planes per single view) to obtain same plane segmentation per K views having the same 3D point clouds, such scene global plane segmentation leads to a better plane estimation, and K views depth map consistency.

- Better plane segmentation; allow us to create a new plane dataset for high resolution images; building a CNN model that jointly detects scene planes by grouping superpixels based on their similarity appearance and geometrical constraints; in an objective to overlap the planes with their ground truths, and complete the input sparse depth maps by minimizing the number of the invalid pixels.

References

[1] E. Zheng, E. Dunn, V. Jojic and J. M. Frahm, “PatchMatch Based Joint View Selection and Depthmap Estimation,” *IEEE Conference on Computer Vision and Pattern Recognition,* 2014 .

[2] J. L. Schonberger, E. Zheng, M. Pollefeys and J.-M. Frahm, “Pixelwise view selection for unstructured multi-view stereo,” *European Conference on Computer Vision,* 2016.

[3] P. H. Huang, K. Matzen, J. Kopf, N. Ahuja and J. B. Huang, “DeepMVS: Learning Multi-view Stereopsis,” *IEEE Computer Vision and Pattern Recognition,* 2018.

[4] A. Romanoni and M. Matteucci, “TAPA-MVS: Textureless-Aware PAtchMatch Multi-View Stereo,” *IEEE Computer Vision and Pattern Recognition,* 2019.

[5] A. Kuhn, S. Lin and O. Erdler, “Plane Completion and Filtering for Multi-View Stereo Reconstruction,” *41th German Conference on Pattern Recognition,* 2019.

[6] R. Tsai, “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses,” *IEEE Journal on Robotics and Automation*, 1987.

[7] R. Hartley and A. Zisserman, Multiple view geometry in computer vision, Cambridge Univesity Press, 2003.

[8] C. D. Herreramm, J. Kannala, L. Ladicky and J. Heikkila, “Depth map inpainting under a second-order smoothness prior,” *7944,* pp. 555-566, June 2013.

[9] X. Gong, J. Liu, W. Zhou and J. Liu, “Guided depth enhancement via a fast marching method,” *Image and Vision Computing,* vol. 31, no. 10, pp. 695-703, 2013.

[10] D. Doria and R. J. Radke, “Filling large holes in LiDAR data by inpainting depth gradients,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,* 2012.

[11] K. Matsuo and A. Y, “Depth image enhancement using local tangent plane approximations,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2015.

[12] J. Lu, D. Min, S. Pahwa and M. N. Do, “A revisit to MRF-based depth map super-resolution and enhancement,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 2011.

[13] J. Lu and D. Forsyth, “Sparse depth super resolution,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2015.

[14] Z. Jia, Y. Chang, T. H. Lin and T. Chen, “Dense interpolation of 3D points based on surface and color,” *18th IEEE International Conference on Image Processing,* 2011.

[15] A. Sampath and J. Shan, “Segmentation and Reconstruction of Polyhedral Building Roofs From Aerial Lidar Point Clouds,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 48, no. 3, pp. 1554 - 1567, 2010.

[16] A. G. Schwing, T. Hazan, M. Pollefeys and R. Urtasun, “Efficient structured prediction for 3D indoor scene understanding,” *IEEE Conference on Computer Vision and Pattern Recognition,* 2012.

[17] C. Liu, J. Yang, D. Ceylan, E. Yumer and Y. Furukawa, “PlaneNet: Piece-wise Planar Reconstruction from a Single RGB Image,” *Conference on Computer Vision and Pattern Recognition,* 2018.

[18] Z. F and Z. Yang, “Recovering 3D Planes from a Single Image via Convolutional Neural Networks,” *European Conference on Computer Vision,* 2018.

[19] C. Liu, K. Kim, J. Gu, Y. Furukawa and J. Kautz, “PlaneRCNN: 3D Plane Detection and Reconstruction from a Single Image,” *IEEE Conference on Computer Vision and Pattern Recognition,* 2019.

[20] K. He, G. Gkioxari, P. Dollár and R. B. Girshick, “Mask R-CNN for Object Detection and Segmentation,” *IEEE Conference on Computer Vision and Pattern Recognition,* 2017.

[21] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan and S. Belongie, “Feature Pyramid Networks for Object Detection,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 2117-2125, 2017.

[22] O. Ronneberger, P. Fischer and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, LNCS,* vol. 9351, pp. 234--241, 2015.

[23] D. Angela, C. X. Angel, S. Manolis, H. Maciej, F. Thomas and N. Matthias, “ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes,” *IEEE Computer Vision and Pattern Recognition (CVPR) Proceeding.,* 2017.

[24] M. v. d. Bergh, X. Boix, G. Roig, B. d. Capitani and L. v. Gool., “SEEDS: Superpixels extracted via energy-driven sampling,” *Proceedings of the European Conference on Computer Vision,* p. 13–26, 2012.

[25] D. Stutz and L. B, “Superpixel Segmentation: An Evaluation,” *German Conference on Pattern Recognition,* pp. 555-562, 2015.

[26] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys and A. Geiger, “A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2017.

[27] J. L. Schönberger and J. Frahm, “Structure-from-Motion Revisited,” *IEEE Conference on Computer Vision and Pattern Recognition,* 2016.