Utilizing Depth Information for Emerging 3D Applications

Jun Xie

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2016

Reading Committee:

Ming-Ting Sun, Chair

Linda Shapiro

Eve Riskin

Program Authorized to Offer Degree:
Department of Electrical Engineering

University of Washington

Abstract

Utilizing Depth Information for Emerging 3D Applications

Jun Xie

Chair of the Supervisory Committee:
Professor Ming-Ting Sun
Department of Electrical Engineering

In recent decades, we have witnessed the explosive growth of research in computer vision, graphics, and robotics brought by the evolution of depth cameras. With depth information, computers will have a better understanding of the physical world. However, the quality of the depth images captured by current low-cost depth sensing devices is often poor and noisy. In this dissertation, we describe the efforts that we have made in utilizing the potentially noisy depth information for some emerging 3D applications. Our contributions are mainly in three areas: First, to improve the construction of 3D object models using noisy depth information, we propose a novel algorithm that can achieve a better 3D point cloud registration. Second, since the resolutions of the depth images from current low-cost depth sensing devices are often not sufficient for practical applications, we propose novel algorithms that perform better than existing state-of-the-art algorithms for depth image super resolution. Third, we utilize the depth information to help create a large-scale street-view dataset with semantic and instance level scene labeling using 3D to 2D label transfer for the purpose of scene understanding and autonomous driving research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I am heartily thankful to my advisor, Professor Ming-Ting Sun, for his continuous support and mentoring of my research. His patience, supervision, and immense knowledge enabled me to develop a thorough understanding of the field and become a better person.

I am also fortunate to have received lots of help throughout my Ph.D. career. First of all, I would like to express my sincere gratitude to Dr. Rogerio Feris at IBM Research. I truly appreciate him for being my co-advisor and friend, for providing me with a lot of research insights. I would also like to thank Dr. Holger Winnemöller, Dr. Wilmot Li, Dr. Aaron Hertzman, and Dr. Stephen Schiller at Adobe Research, who were willing to give me an opportunity as a summer intern and to share expertise in their research fields even though I was not familiar with Computer Graphics. I would also like to thank my mentors Gina Venolia and Dr. Cha Zhang at Microsoft Research. Collaboration with them was such a fruitful and enjoyable learning experience.

I extend my thanks to Dr. Andreas Geiger at Perceiving Systems in Max-Planck Institute, Germany, where I spent half a year working closely with him on an exciting project. I would like to thank him for opening a new door to the wonderful world of computer vision for me, teaching me everything wholeheartedly, and showing me what it is like to be an excellent researcher.

I would like to thank all of my colleagues, who inspire me and whose hard work encourages me a lot. I am also indebted to anonymous reviewers for their selflessly sharing great thoughts on my research projects.

Furthermore, special thanks to my committee members, Professor Linda Shapiro, Professor Eve Riskin, and Professor Ali Farhadi for their valuable feedback to make my research work intact.

Finally yet importantly, I want to thank my fiancé Charlie, for always being there, wholeheartedly encouraging and helping me. Thanks to my parents and my family, for their support and for being my strength.

Chapter 1

INTRODUCTION

1.1 Background

Nowadays, depth sensors are becoming more and more popular in various research applications. For example, the time-of-flight cameras SwissRange SR4000 depth camera and the PMD Camcube camera have very high accuracy in capturing object depth. LiDAR laser scanners such as Velodyne rotating scanner HDL or SICK LMS 3D laser scanners are widely used on cars or drones for applications such as 3D scene understanding and reconstruction. RGB-D sensors such as the Microsoft Kinect and the PrimeSense Carmine are nowadays more affordable cameras (less than $400 USD) that are capable of capturing the high-resolution RGB and associated depth images simultaneously.

These depth sensing technologies represent an opportunity to dramatically increase the object recognition, manipulation, navigation, and interaction capabilities in robots. The success of utilizing depth has been demonstrated in applications including autonomous car driving as well as robots navigating the home environment. For example, researchers in [77, 105] have investigated the detection and tracking of cars using the Velodyne laser array as the sensor. Others have looked at object recognition in mobile robots navigating an urban environment with a very high accuracy [140].

However, the depth information is sometimes sparse and noisy. It is quite often a challenge when the quality of depth images is not satisfactory, as this limits the related 3D applications. For instance, the depth images from RGB-D cameras are noisy with some information missing around depth discontinuities. In this sense, the depth information can be misleading for detecting objects. Moreover, the relatively low-resolution depth images also limit the applications that require accurate depth information, such as 3D object reconstruction or 3D scene understanding. Furthermore,

this issue raises an interesting question: under the sparsity constraint of the depth data, can we fully utilize it to aid the 2D applications that require getting the pixel-wise information such as pixel-level image/video segmentation?

1.2 Thesis Outline

In this dissertation, we propose improved techniques for depth processing and applications that utilize depth information. The dissertation demonstrates the following thesis:

1. *Depth-related applications such as point cloud registration can be significantly improved with the combination of RGB and depth data.*

2. *Single-view depth image quality can be enhanced through increasing its resolution and noise removal.*

3. *Sparse depth information can be fully utilized by efficiently creating a large scale semantic and instance level segmentation.*

In Chapter 2, we describe our work on 3D point cloud registration. We propose a new approach for fine registration of 3D point clouds by combining the RGB and depth information in a modified iterative closest point (ICP) framework. We also develop a robust outlier rejection method based on the color information under challenging scenes such as when the object lacks structural features or when there is a dramatic view change. We show in the experiments that our proposed method can achieve superior results compared to other related methods, in terms of both registration accuracy and efficiency.

In Chapter 3, we discuss our two proposed approaches for single-view depth image super-resolution: a Coupled Dictionary Learning-based approach with Local Constraint, and an Edge-Guided approach. In the Coupled Dictionary Learning approach, we jointly increase the resolution and remove the noise of the depth image. In the Edge Guided approach, we propose a new framework which converts the depth super-resolution problem into an edge upsampling and depth interpolation problem. We also investigate the possibilities of learning the high-resolution edges solely

based on self-similarities. Experiments demonstrate that our approach outperforms other methods. As a comparison of the two proposed approaches, the dictionary learning method better handles the noise present in the depth image, while the Edge Guided method performs better in terms of percent of error in the depth map, as well as the visual result. It is worth mentioning that for the Edge Guided method, we are able to achieve a 29% drop in error compared with state-of-the-art methods (including color assisted depth super-resolution approaches).

In Chapter 4, we propose utilizing depth information to help create a large scale scene semantic and instance segmentation benchmark for 2D/3D scene labeling research and applications. We contribute a novel suburban dataset with over 400k frames and laser scans in total and annotate all objects using 3D bounding primitives. We also propose a novel Markov Random Field model capable of transferring the 3D label information to every pixel in every image by reasoning jointly about pixels, sparse 3D points from LiDAR and stereo videos, as well as geometric cues in 3D.

In Chapter 5, we conclude the dissertation and discuss some possible future directions.

Chapter 2

FINE REGISTRATION OF 3D POINT CLOUDS FUSING STRUCTURAL AND PHOTOMETRIC INFORMATION USING AN RGB-D CAMERA

2.1 Introduction and Related Work

3D object modeling is an active research topic, and has many practical applications such as animation, human computer interaction, virtual reality, and object manipulation by industrial robots [2, 3, 29, 66, 70]. With the birth of low-cost RGB-D cameras (such as Kinect), synchronized RGB and depth images can be captured at the same time, making 3D modeling of an object more robust and accessible.

In a typical 3D modeling process using an RGB-D camera [61], first, the 3D partial point clouds of the object from different views are pairwise registered (or aligned) through a coarse registration algorithm such as RANSAC (Random Sample Consensus) [38,57], and then this initial registration is further refined by an iterative fine registration algorithm such as the ICP (Iterative Closest Point) algorithm [11, 21, 127]. After the fine registration, the 3D point cloud model can be transformed to other 3D representations for different applications.

The ICP convergence is sensitive to outliers and noise. Many points on the source point cloud do not have ideal correspondences on the target model. To improve the performance of ICP, some variants of ICP have been proposed [12, 19, 22, 106, 109, 160]. The variants cover the pruning, downweighting, and outlier rejection of the 3D points, as well as the minimization of the error metric. The work in [109] and [22] set up criteria to discard points that are too far away from the correspondence or too close to the geometry boundary. In [106], the fraction of inliers is computed in a statistically robust manner. The work in [19, 160] introduces robust functions to reweight the importance of inliers instead of trimming outliers. In [12], people use sparsity-inducing norms to constrain inliers in order to avoid the heuristics of pruning or reweighting correspondences.

However, in some cases, such as an object lacking distinguishing structural features or under significant camera view changes, even if we achieve an almost perfect alignment in the initial alignment stage, these ICP variants may actually move away from the correct alignment and converge to an incorrect alignment result since they only use structural information (i.e., the 3D coordinate information), as will be shown in the next section.

Some works propose to simultaneously consider multiple correspondences per feature point using the Expectation Maximization (EM) algorithms [25, 48] or based on a Gaussian Mixture Model (GMM) [63]. Although these methods improve the convergence of ICP, in the case of a large number of outliers or noise, registration still heavily relies on the distance to the nearest matching feature, which makes these methods less robust.

Several color-based ICP algorithms [32, 64, 101] have been proposed to alleviate this issue, showing that adding the color information decreases the registration error significantly when objects lack structural features. However, directly using color as a feature is not reliable, since the color may not be consistent in different views due to lighting, shadow, or reflection.

In [51, 79], SIFT descriptors [87] are incorporated into the ICP iteration process for improved registration when objects lack structural features. However, in [79] the algorithm operates solely on sparse SIFT feature points, which is a very small subset of the point clouds when extended to the 3D modeling case. The performance of the algorithm is limited since the rich structural information from the 3D point clouds is not fully used. The algorithm could be extended to running on all the 3D points in the point clouds. However, it would require computing the 128-dimensional SIFT descriptor for every point in the object in the search for the closest distance, which is very computationally inefficient. Also, it will have problems if the object lacks salient texture features. Furthermore, both [79] and [51] utilize a fixed coefficient for weighting the closest distance and SIFT matching distance, which may not provide the best performance.

Aside from ICP, there is also existing work in 3D point cloud registration. In [91], an efficient indexing scheme in 3D registration is proposed, which speeds up the algorithm towards finding the global optimal alignment. Researchers have also proposed to use high-level features to improve the surface registration accuracy. In [69], an intrinsic map is generated between two non-isometric,

genus zero surfaces. Based on the blended map, dense feature correspondences can be established. In [99], a general algorithm is proposed to find intrinsic symmetries of shapes with Heat Kernel Signature [122] in order to match partial and incomplete models efficiently. The work in [129] introduces a new shape-matching algorithm for computing correspondences between 3D surfaces that undergo isometric deformations. However, notwithstanding the demonstrated success of these surface registration approaches, in the RGB-D case, 3D surface reconstruction is vulnerable to the noise contained in the point cloud, which makes the surface-based registration less accurate.

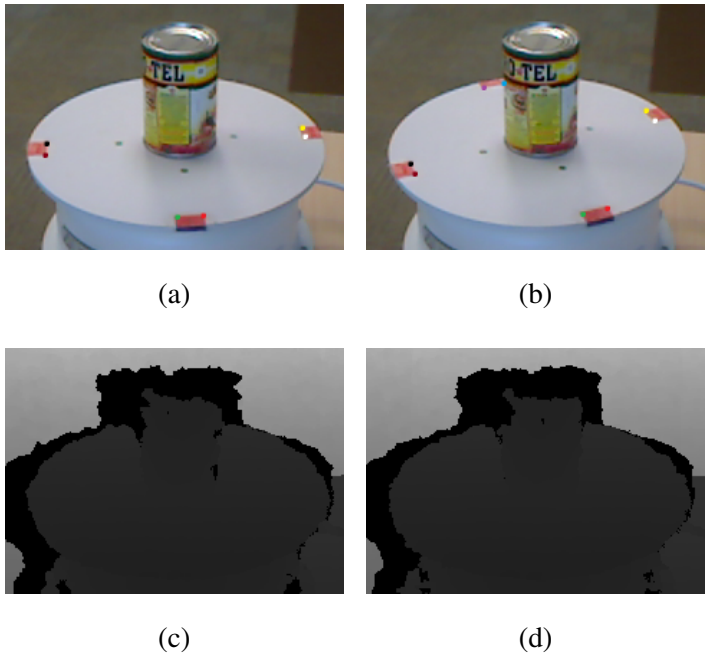
In this chapter, we propose a more robust and efficient point cloud fine registration approach by enhancing ICP with a new cost function that balances the significance of structural and photometric features with dynamically adjusted weights to improve the error minimization process. In addition, we introduce a novel outlier rejection method, which adaptively sets the outlier distance threshold in each ICP iteration, while taking into account both 3D structural features of the object and the spatial distances of the SIFT feature pairs. We show that our contributions can achieve superior results compared to other related methods, both in terms of registration accuracy and efficiency. In particular, we demonstrate our approach in several challenging scenarios, involving objects with symmetrical structures and alignment with large camera view displacements [65, 147].

2.2 The Fine Registration Problems

In this section, we show the setup we used to demonstrate the problems we are addressing, i.e., fine registration under some challenging scenarios involving symmetrical objects and significant camera view changes, where ICP produces inaccurate results.

The RGB-D images we used are from the RGB-D Object Dataset [76]. A round food-can is placed on a turntable to illustrate the case of a symmetrical object. Figure 2.1 shows two RGB images and their corresponding depth images from two views, captured using a Kinect camera. The image resolution is 640×480. In the depth images, the lighter intensity of a pixel means it is farther from the camera. For the black region (e.g., the top part of the can and the background area around the turntable), the depth information is not available.

We use only the SIFT features extracted from the food-can in the RGB images to perform

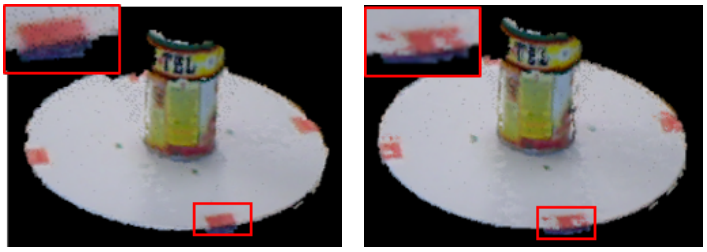


(a) (b)

(c) (d)

Figure 2.1: The RGB and depth images of two different views showing the setup for illustrating a case where the standard ICP fails to perform fine registration due to structural ambiguity.

RANSAC for the initial alignment. After the initial registration, we perform ICP on the food-can for the fine alignment to obtain the final transform. Since the food-can and the turntable are rigid objects and are fixed to each other, ideally, the transform should also apply to the turntable. We use the four red rectangle markers on the turntable as shown in the figure to demonstrate this problem. Since the markers are very sharply defined and have a very distinct color from the turntable, it is easy to precisely extract the corners of the markers (in the simulations we use the Harris corner detector) which are highlighted as color dots in the figure. These corner points serve as the ground truth points for comparing the alignment results in our simulations. With the coordinates of the six ground truth points, which are visible in both 3D point clouds, we can compute the errors of the ground truth points during the fine registration process. Since the markers are at a distance from the food-can, and the ICP is performed only based on the 3D points of the food-can, the markers also serve the purpose of making the errors more visible. It should be noted that most of the black



(a) (b)

Figure 2.2: **3D point clouds registered from two views.** (a) Initial registration result after RANSAC. (b) Fine registration result after ICP.

area around the turntable belongs to the background. Since in the simulations we only deal with the food-can and the corners of the markers, this black part does not affect our results.

In the above case, the food-can is round without any distinct structural features. Figure 2.2a and Figure 2.2b show the partial 3D point clouds after the initial alignment and after the ICP registration from the two views in Figure 2.1, respectively. The pictures look noisy since the depth images from the Kinect camera are noisy. From Figure 2.2a, we can see that after the coarse initial alignment, the red markers are well aligned. However, after ICP is performed for the fine registration as shown in Figure 2.2b, the markers are no longer aligned. In Figure 2.2b, the two sides of the red marker highlighted are mixed with red and white colors, since with the misalignment, the two red markers are not completely overlapped to each other. Due to the noisy depth values, in some locations of the non-overlapped regions, the white color turntable may appear in front, and in other locations, the red color marker may appear in front, which causes the region to have a mixed red and white look.

In Figure 2.3, we plot the RMS_d which is the RMS (Root Mean Square) value of the closest distances of the 3D point clouds and the RMS distance of the ground truth points (the 3D coordinates of the red marker corners) in each iteration. From the figure, we can see that although the RMS value of the closest distances of the 3D points of the food-can continues to decrease, the RMS distance error of the ground truth points is increasing, indicating that the ICP is actually converging

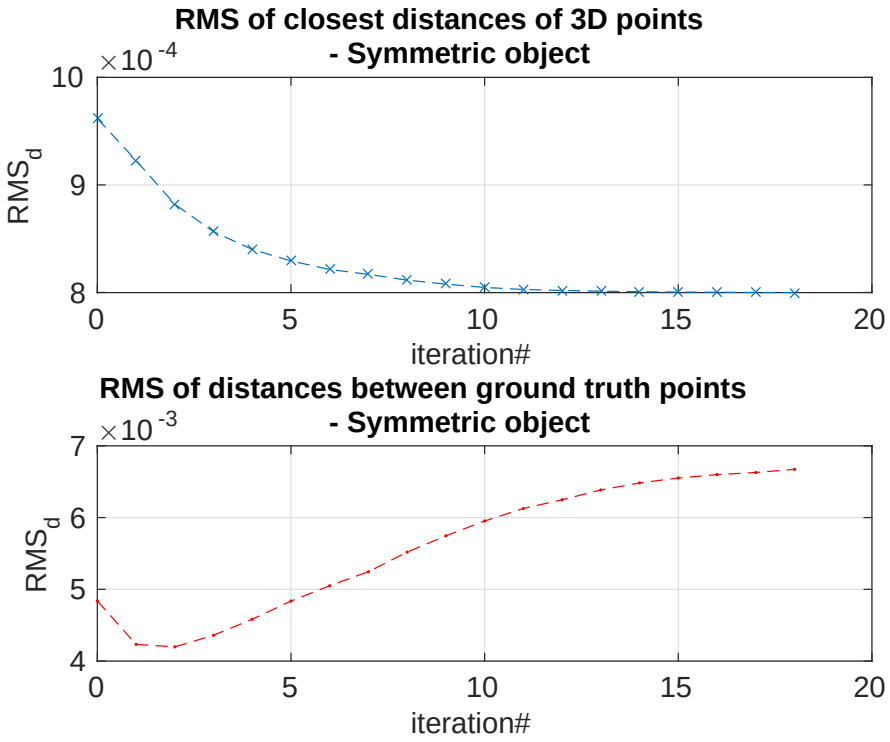
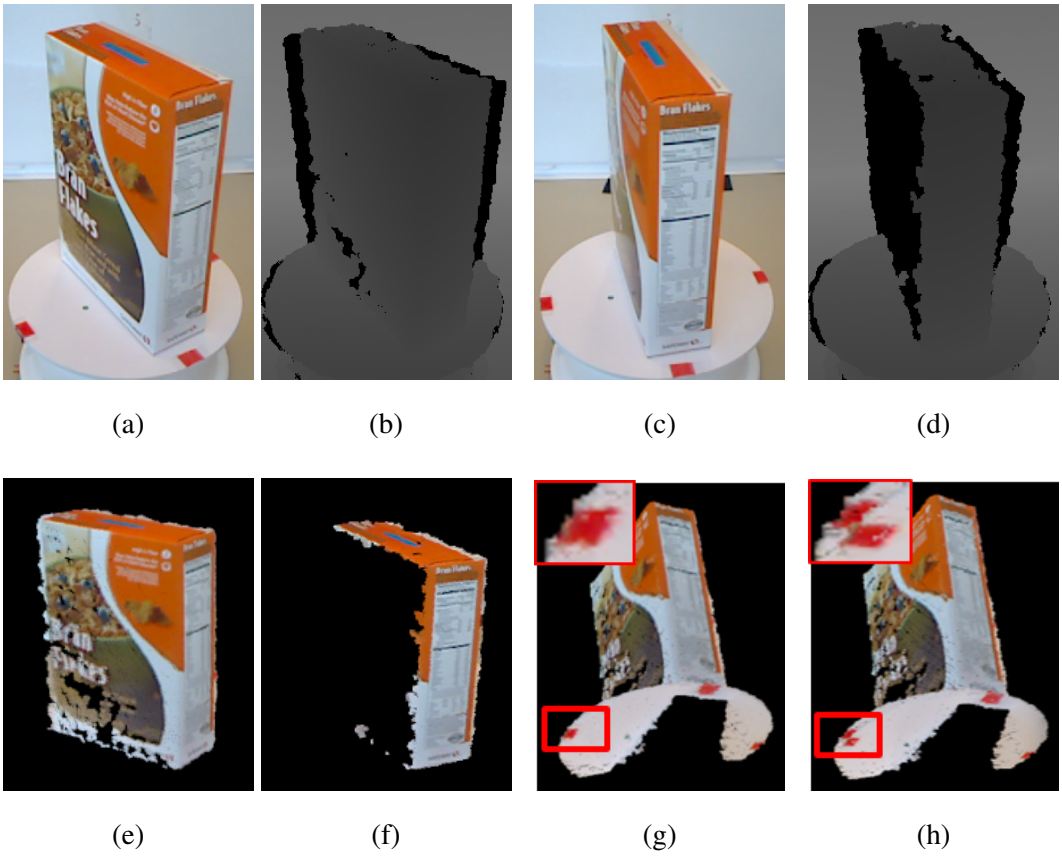


Figure 2.3: Fine-registration error curve for a symmetrical object with ICP. (Unit: m)

to a wrong position due to the lack of structural features.

We also consider the scenario where the overlap region in the two views is relatively small due to significant camera view changes, as shown in the cereal box case in Figure 2.4. Figure 2.4c and Figure 2.4d show the visual alignment result after the initial alignment and after ICP, respectively. 2.5 shows the corresponding error curves. From the figures, we can see that although the RMS value of the closest distances of the 3D points of the cereal box continues to decrease, the RMS error of the ground truth points increases, which also indicates that ICP is converging to a wrong position.



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

(e) (f) (g) (h)

Figure 2.4: **Alignment of ICP for the case with a small overlapping region in the two views.** (a)-(d) The RGB and depth images of two different views. (e)-(f) Partial point clouds from the two views. (g) Visual result after initial alignment. (h) Fine registration visual result after ICP. (Best viewed in color)

2.3 Proposed Method for Fine Registration

Given the RGB and depth images of two views from the RGB-D camera, we can obtain two 3D point clouds $p = \{p_1 ... p_N\}$ and $q = \{q_1 ... q_M\}$, where N and M are the numbers of points in the two 3D point clouds, respectively. The SIFT feature points are extracted from the two RGB images. In the initial alignment provided by RANSAC, we find the set of corresponding SIFT feature 3D points as $cf = (pf_1, qf_1)$, ... , (pf_L, qf_L) where pf_i and qf_i are the corresponding SIFT feature

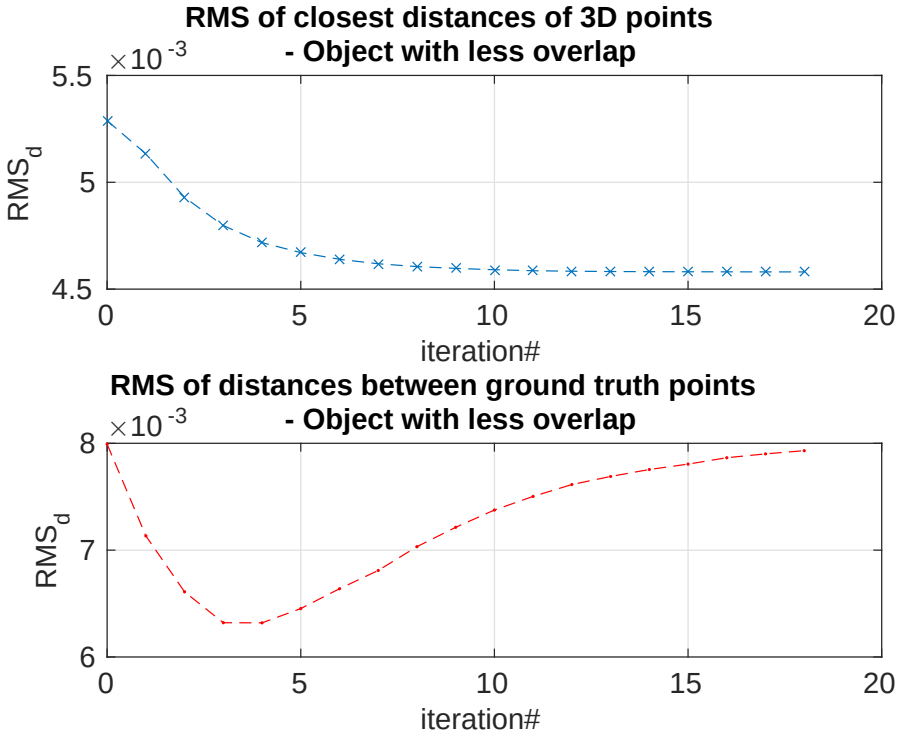


Figure 2.5: Convergence of ICP for the case with a relatively small overlapping region in the two views. (Unit: m)

3-D points in p and q, respectively, and L is the number of the matched SIFT feature point pairs. It should be noted that the matched SIFT feature point pairs are relatively sparse compared to the original 3-D point clouds (i.e., L is much smaller than M and N). After the initial alignment, the standard ICP algorithm performs fine registration of the two point clouds by iteratively associating points through a nearest-neighbor search and estimating the transformation parameters using a mean square cost function (see [11, 21] for details).

2.3.1 Objective Function Fusing Structural and Photometric Information

To overcome the problem associated with the case of objects lacking structural features, a SIFT-based term can be added into the cost function in the error minimization process, in addition to

the closest distances of the nearest neighbors. To fully utilize the rich structural information of the 3D points, unlike [79] where the iterations are only applied to the sparse SIFT feature points, in our proposed approach the iterations are applied to all the 3D points. To prevent the calculation of the 128-dimensional SIFT descriptor for every 3D point which is computationally intensive, we propose to use the spatial distances of the matched SIFT feature pairs, which are readily available in the iterations of the ICP algorithm, instead. In this section, we define $T^{(0)}$ as the transform matrix after RANSAC and before the iteration, and $T^{(k)}$ as the transform matrix after the kth iteration in the process.

The new objective function to be minimized in the kth iteration to find $T^{(k)}$ is

$$E_{(k)} = \sum_{p_i \in p} \alpha_i \| p_i \cdot T - q_i^{*(k)} \|^2 + \sum_{(pf_i, qf_i) \in cf} \beta_i \| pf_i \cdot T - qf_i \|^2 \tag{2.1}$$

where $q_i^{*(k)}$ is the corresponding point in the point cloud q with the closest distance to each point p_i in the point cloud p given two point clouds p and q:

$$q_i^{*(k)} = \underset{q_j \in \{q\}}{\arg\min} (\| p_i \cdot T^{(k-1)} - q_j \|^2). \tag{2.2}$$

pf_i and qf_i are corresponding SIFT features points in q and q in 3D.

α_i and β_i are the weights for the closest distance of the nearest neighbor and the spatial distance of the corresponding SIFT feature pair, respectively, and will be discussed further in the following sections. The first term in the objective function of Eqn. 2.1 is the weighted mean square of the closest distances of the inliers. It should be noted that in Eqn. 2.1 we show the point-to-point distance as the error metric. In the simulations, we also try using the point-to-plane distance [21] as the error metric, and the results are about the same. In the simulation results shown later, the point-to-plane distance is used. The second regularization term is the weighted mean square of 3D spatial distances of the SIFT feature correspondence pairs. It effectively constrains the convergence to the correct direction, which minimizes both the spatial distances of points with structural features and the spatial distances of SIFT correspondence pairs that represent texture features.

2.3.2 Adaptive Outlier Rejection and Dynamic Weighting for the 3D Points

As the alignment being refined, the outliers in the closest distance matching should be rejected in each iteration so that they do not affect the accuracy of the result. To utilize both the structural characteristics related to the statistics of the closest distances of the 3D point clouds as well as the spatial distances of the SIFT correspondence pairs, we propose a new outlier rejection method based on an adaptive threshold that depends on both the matching errors of the closest distances of the 3D points and the spatial distances of the SIFT correspondence pairs. The adaptive threshold for the outlier rejection is defined as:

$$t^{(k)} = c \cdot \sqrt{er^{(k)} \cdot df^{(k)}} \tag{2.3}$$

where c is a constant. $er^{(k)}$ is the root mean square of the closest distances for the statistical inliers $s^{(k)}$ defined as:

$$er^{(k)} = \sqrt{mean_{p_i \in s^{(k)}}(\|p_i \cdot T^{(k-1)} - q_i^{*(k)}\|^2)} \tag{2.4}$$

where

$$s^{(k)} = \{p_i | cd_i^{(k)} + 3 \cdot std(cd_i^{(k)})\} \tag{2.5a}$$

$$cd_i^{(k)} = \|p_i \cdot T^{(k-1)} - q_i^{*(k)}\|. \tag{2.5b}$$

$df^{(k)}$ is the average spatial distance of $m\%$ SIFT feature correspondence pairs with shorter closest distances. We make the adaptive threshold $t^{(k)}$ for outlier rejection depend on the average spatial distance of $m\%$ SIFT feature correspondence pairs with shorter spatial distances instead of the average distance of all the SIFT feature correspondence pairs, because even after the RANSAC initial alignment, some of the SIFT feature points may still not be well matched. So, some of the spatial distances may be relatively large which makes the use of the average distance of all the SIFT feature pairs not appropriate. Using a subset of SIFT feature pairs with shorter distances, we can ensure the accuracy of the chosen feature correspondences. In practice, we choose $m = 30$,

which is trained from experiments. We also find that when using a percentage between 25% and 60%, the final accuracy is not sensitive to the choice of that percentage parameter. The outlier rejection is implemented by setting α_i in Eqn. 2.1 to zero if the closest distance of a 3D point pair is larger than the adaptive threshold calculated from Eqn. 2.3

After the outlier rejection, some inliers are more reliable than others. If two corresponding 3D points based on closest distance have similar surface variations, they are more likely to be the correct pair, and so the weighting can be adjusted with more confidence and vice versa. As mentioned in [104], the surface variation is closely related to the curvature but needs much less computation than the curvature calculation. Since local features can have a better representation of the surface structure, we set α_i based on the 3D local surface variations as:

$$\alpha_i = \begin{cases} \sqrt{\frac{0.01}{|\sigma(p_i)-\sigma(q_i^{*(k)})|}} & cd_i^{(k)} < t^{(k)} \\ 0 & otherwise \end{cases} \tag{2.6}$$

Here $\sigma(*)$ denotes the surface variation defined in [104]:

$$\sigma(p_i) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{2.7}$$

and $\lambda_i(\lambda_0 \leq \lambda_1\lambda_2)$ are eigenvalues of the covariance matrix:

$$C = \frac{1}{r} \cdot \begin{bmatrix} p_{i1} - \bar{p}_i \\ ... \\ p_{ir} - \bar{p}_i \end{bmatrix} \cdot \begin{bmatrix} p_{i1} - \bar{p}_i \\ ... \\ p_{ir} - \bar{p}_i \end{bmatrix}^T . \tag{2.8}$$

$p_{i1}...p_{ir}$ are the closest r points around p_i and $\bar{p}_i$ is the centroid of these local neighbors.

2.3.3 Dynamic Weighting for the SIFT Feature Pairs

From the experiments, we found that the relative weighting for the SIFT feature correspondence distances is also important. If the weighting is too large, the transformation will mainly depend on the relatively small number of the SIFT feature correspondences. This could give problems when the SIFT feature correspondences are not completely reliable. On the other hand, if the weighting

is set too small, the feature based regularization term becomes less significant. To properly balance the significance of the structural and photometric terms, the weight β_i should reflect the relative reliability between the structural and photometric terms. If the 2D SIFT feature matching distance is large, which means that the SIFT feature matching is less accurate, β_i should be smaller. Also, if the RMS of the spatial distances of the SIFT feature correspondences is larger than the RMS distances of the inlier set $s^{(k)}$, β_i should also be smaller. Based on the above argument, β_i is set as:

$$\beta_i = c' \cdot \frac{1}{dist(pf_i, qf_i)} \cdot \frac{er^{(k)}}{\sqrt{mean_{(pf_i,qf_i)\in cf}(\|pf_i \cdot T^{(k-1)} - qf_i\|^2)}} \tag{2.9}$$

where c' is a constant, $dist(pf_i, qf_i)$ is the 2D SIFT matching distance between (pf_i, qf_i), which is available from the SIFT feature matching in the RANSAC initial registration stage, and the second term is the ratio between the RMS distances of the inlier set $s^{(k)}$ and the RMS of the spatial distances of the SIFT feature correspondences. $er^{(k)}$ is calculated from Eqn. 2.4.

2.3.4 Summary of the Overall Algorithm

In the algorithm described above, the adaptive threshold $t^{(k)}$ utilizes both the structural information and the SIFT features of the point clouds. Moreover, the SIFT feature matching constraint is added into the objective function with a dynamic weighting. As a result, the algorithm will converge properly even for challenging scenarios. It should be noted that from the experimental results, the parameters we set are not sensitive to the change of datasets. Unlike the outlier rejection method described in [160], our proposed algorithm utilizes the texture feature information in the outlier rejections. Moreover, unlike the color-based ICP algorithm in [32, 64], our method is more robust to lighting changes. Our proposed 3D registration algorithm is summarized as follows. For the kth iteration $(k = 1, 2, ...)$ in the fine registration process:

(i) For each point p_i in the point cloud p, find its corresponding point $q_i^{*(k)}$ by searching for its closest point in q. In our implementation, we use a k-d tree to efficiently find the nearest neighbor.

$$cd_i^{(k)} = \|p_i \cdot T^{(k-1)} - q_i^{*(k)}\|. \tag{2.10}$$

(ii) Compute the statistical inliers $s^{(k)}$ according to Eqn. 2.5, and $er^{(k)}$ according to Eqn. 2.4. Then we calculate $df^{(k)}$ from the average spatial distance of 30% SIFT feature correspondence pairs with shorter closest distances.

(iii) Calculate the adaptive threshold for outlier rejection defined in Eqn. 2.3:

$$t^{(k)} = c \cdot \sqrt{er^{(k)} \cdot df^{(k)}}.$$

(iv) Compute the dynamic weights for outlier rejection and balancing the two terms α_i and β_i according to Eqn. 2.6 and Eqn. 2.9. Then we find the transformation $T^{(k)}$ by minimizing the objective function:

$$E^{(k)}(T) = \sum_{p_i \in p} \alpha_i \|p_i \cdot T - q_i^{*(k)}\|^2 + \sum_{(pf_i, qf_i) \in cf} \beta_i \|pf_i \cdot T - qf_i\|^2.$$

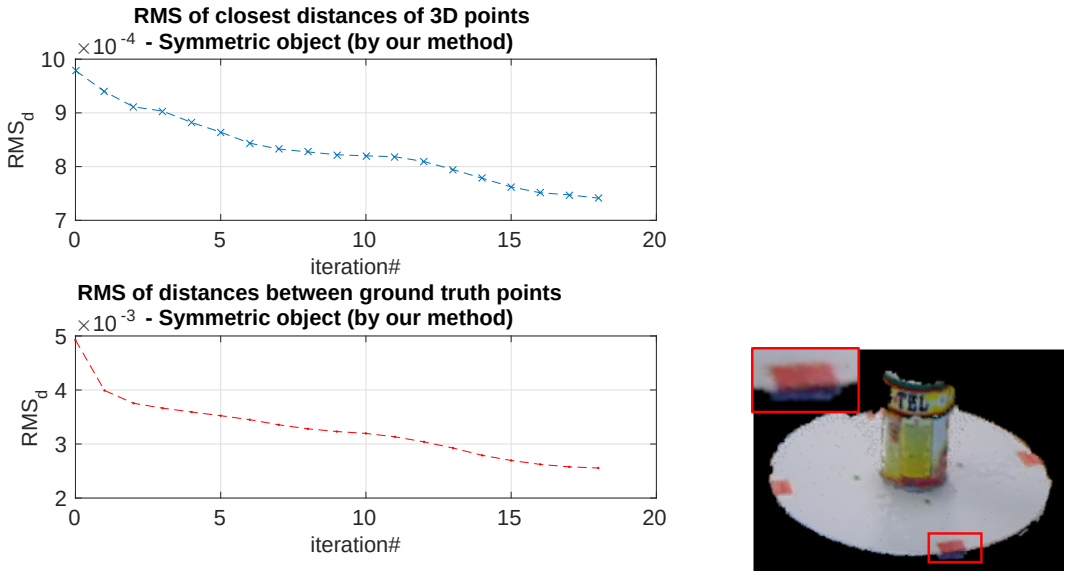
$T^{(k)} = \arg\min_T(E^{(k)}(T))$. Also we delete points p_i from p with $cd_i^{(k)} > 10 \cdot t^{(k)}$ so that in the next iteration, we just need to calculate the closest distance for those remaining points.

(v) The iteration terminates after the RMS of the closest distances of the inliers is smaller than a set threshold, or until a fixed number of iterations is reached (we set the iteration number to 18 in our experiments).

We show that this approach is also effective to improve the performance of ICP in the situation of significant camera view changes. In this situation, the overlapping region is relatively small. If the threshold of the outlier rejection only depends on the statistic information of the closest distances of the 3D points, the threshold will be relatively large due to the large number of outliers, meaning fewer points will be rejected. This causes convergence problems and makes the registration result inaccurate. Our outlier rejection method makes the threshold tighter under this situation, which improves the performance of the registration.

2.4 Experimental Results

In Figure 2.6, we show the alignment result for the case of an object with a symmetrical structure (the food-can case in Figure 2.2) using our proposed algorithm. In this case, the result converges



(a) (b)

Figure 2.6: **Fine-registration errors for the symmetric object by our proposed method.** (a) Error curves (Unit: m). (b) The associated visual result of registering the two point clouds from Figure 2.1. (Compare to Figure 2.2b.)

correctly towards the ground truth points and the errors continue to decrease. Also, the errors are much smaller. In Figure 2.6, we show the associated visual result. Compared to the original ICP in Figure 2.2b, with our proposed approach in Figure 2.6b, the markers are aligned very well, which shows the effectiveness of our approach.

In the food-can case, the approximate percentage of overlaps is about 60%, so we also draw the curves in Figure 2.7a with a fixed 40% (which gives better performance compared to other fixed percentages) outlier rejection method [22], the outlier rejection method in [160], and the SIFT based registration approach in [79] for comparison. We also compare our results with state-of-the-art ICP variant methods such as [12] and [19]. As can be seen from the figure, these methods have different degrees of convergence problems, and have larger mean square errors as the iteration runs. We also draw the curves of results from the color-based ICP approach in [64] with different shading

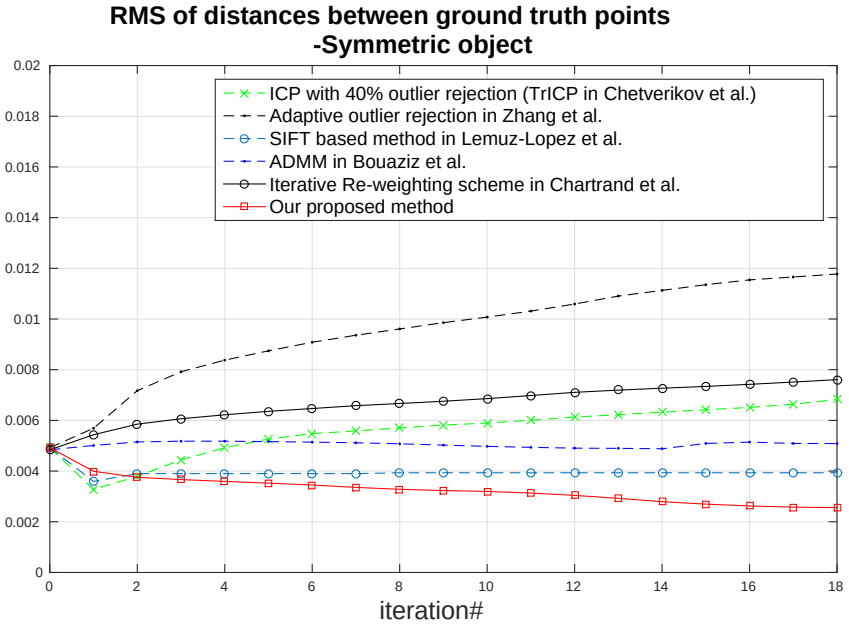
conditions in Figure 2.7b. In the color-based ICP results, the errors are much larger compared to the result of our approach, and its accuracy varies significantly in different shading conditions. In our approach, since the SIFT descriptor is more robust to illumination changes, the inlier SIFT features do not change in this case, so the varied shading does not affect the fine registration results. It should be noted that the scale is different in the two figures due to different ranges of errors in the comparisons.

In Figure 2.8, we show the case for an object with relatively small overlapping regions (the cereal box case in Figure 2.4) using our proposed algorithm. From the error curve, we can see that our result converges correctly. Also, the errors are much smaller. Figure 2.8 shows the associated visual result of this case, in which the markers are aligned well.

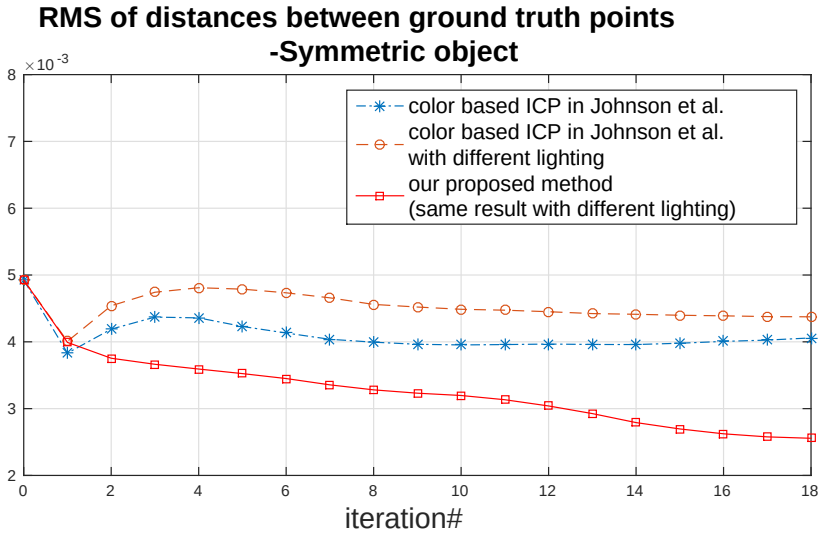
Figure 2.9 shows the RMS error of the ground truth points for the small overlap case with different methods. In this case, the percentage of overlap is about 40%, so in Figure 2.9a, we also draw the curves with a fixed 60% outlier rejection approach [22], (which gives better performance compared to other fixed percentages), the adaptive outlier rejection method in [160], the SIFT based registration approach in [79], and the state-of-the-art ICP variant methods such as [12] and [19]. From Figure 2.9a, we can see that for previously reported approaches, the ICP registration result becomes less accurate as the iteration runs. However, from the result of our approach, the error becomes much smaller as the iteration runs. We also show the result of the color-based ICP in [64] and our proposed method in different shading conditions in Figure 2.9b. From Figure 2.9b, we can see that in different shading conditions, the color based ICP in [64] varies in convergence. For our proposed method, which is independent of the color and illumination, even with different initial alignments, the errors of the ground truth points of both cases converge to much smaller values.

We have also conducted quantitative experiments compared with more 3D registration baseline methods.

Dataset. In addition to the RGB-D dataset in [76], we also pick RGB-D data from the TUM RGB-D dataset [121]. Figure 2.10 shows some of the examples of the RGB-D data that are used in our testing scenario. We construct the point clouds from the RGB-D images also with different point density values. The content of the point clouds ranges from a small object placed on a

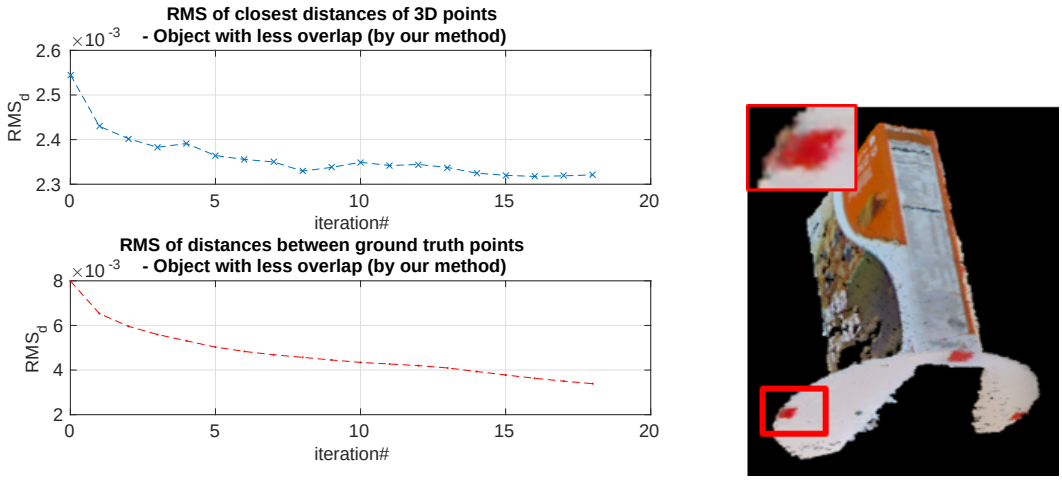


(a)



(b)

Figure 2.7: **Comparison of the RMS of the distances between the ground truth points.** (a) With different methods for the symmetric food-can case. (b) With different lighting. (Unit: m)

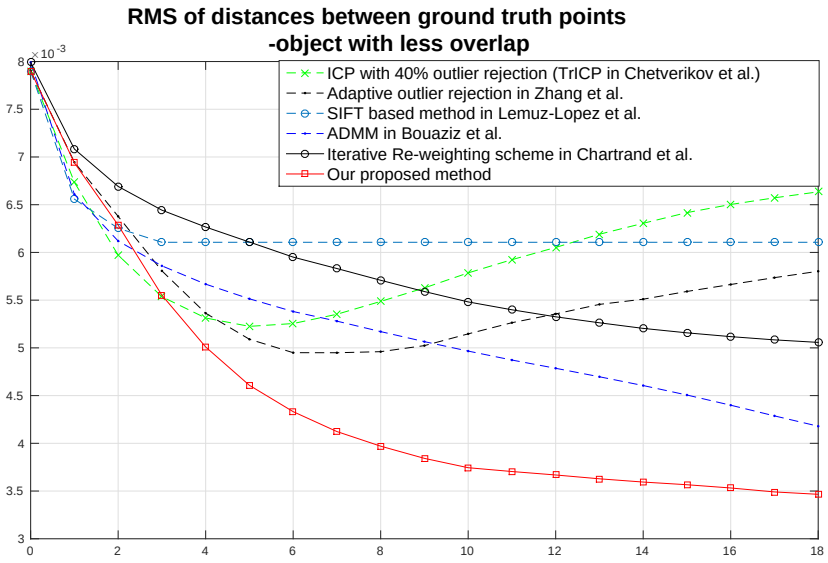


(a) (b)

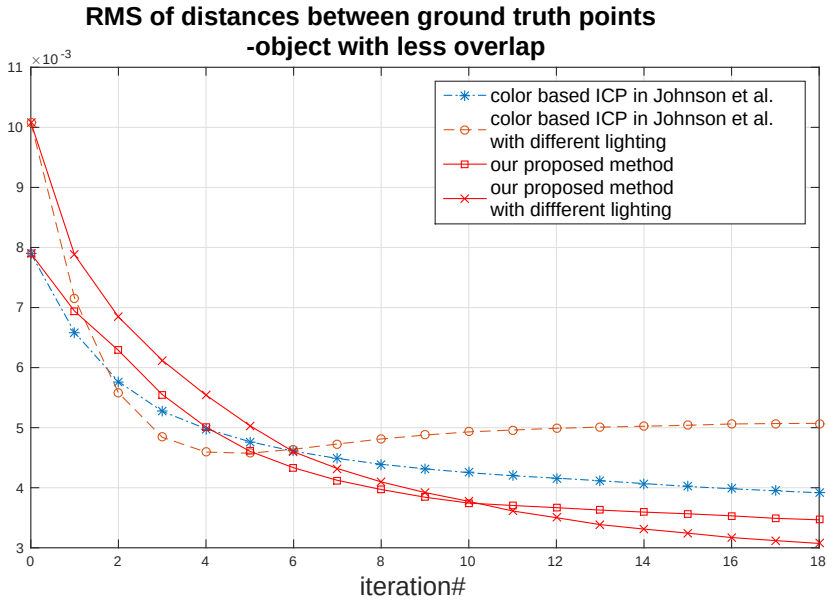
Figure 2.8: **Fine-registration errors for an object with a relatively small overlap in the two views by our proposed method.** (a) Error curves (Unit: m). (b) The associated visual result of registering the two point clouds in Figure 2.4 (Compare to Figure 2.4c).

turntable to a daily office scene. It should be noted that we are mainly focused on the fine registration problem on objects with fewer structural features or different views with small overlaps. So, actually the simple cases such as a round cup or a small box with fewer overlapping regions could be more appropriate in showing the problems of the current ICP algorithms and our superior performance. Nevertheless, for purpose of generalization, we also show our algorithm's effectiveness in more general cases.

The pairwise point clouds are chosen with respect to different categories: 1) Symmetric objects, 2) Two views with smaller overlapping regions, 3) General cases with distinctive geometrical structures. Across the dataset, we arbitrarily choose two views of the object and use the same way to add markers serving as ground truth points. It should be noted that the marker points that we choose will not be covered by the point clouds that we want to register with and are generally far from the point cloud. Therefore, the measured distance between ground truth markers is larger than the



(a)



(b)

Figure 2.9: **Comparison of the RMS of the distances between the ground truth points with different methods.** (a) For the object in two views with relatively small overlap regions. (b) With different lighting. (Unit: m)

actual registration error distance. (This metric magnifies the errors.) For each category, we use six testing data (Test 1 - 6).

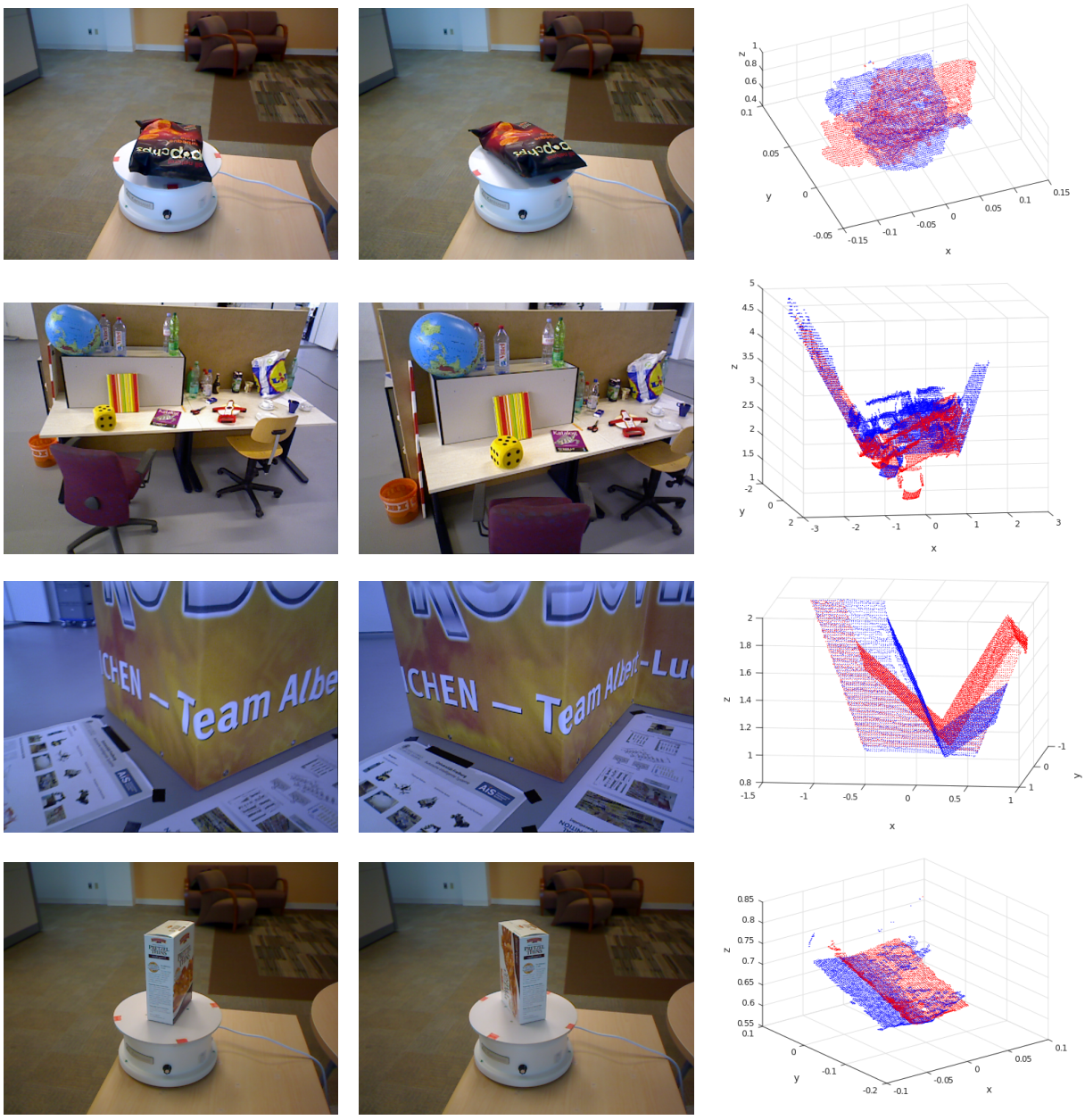


Figure 2.10: Examples of the RGB images from different views (left, middle) and corresponding point cloud (right) for our experiments.

Table 2.1: RMS Error from the Ground Truth with Symmetrical Objects. (Unit: mm)

Data	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Adaptive [160]	3.004	5.087	5.595	11.489	8.801	12.284
TriICP [22]	3.672	9.184	3.570	13.401	7.830	12.030
Reweight ICP [19]	4.193	**4.979**	4.811	17.898	7.616	12.811
Color Based [64]	6.569	7.952	9.579	12.754	8.018	11.841
SIFT Based [79]	7.438	6.115	6.233	11.489	8.801	12.284
ADMM [12]	27.990	22.853	3.135	65.656	13.934	19.135
Super4PC [91]	3.955	27.868	5.116	11.489	39.571	12.357
SURF3D [71]	3.683	20.614	4.179	34.827	70.743	23.280
PFH [112]	20.312	20.614	24.420	85.309	55.983	70.555
Ours	**2.843**	5.578	**3.103**	**11.401**	**7.363**	**11.396**

Baseline Methods. Aside from the ICP related methods such as [160], [64], [22], [79] and [19], we also compare our method with more general 3D registration methods like Super4PC [91], and 3D feature-based registration such as [112] and [71]. For the 3D feature-based registration methods, as mentioned in [112], we first perform k-nearest neighbour to find the correspondences between the descriptors of extracted 3D key points, and then we use RANSAC to reject outliers and estimate the optimal transformation between the two point clouds.

We compare the RMS distances from the ground truth points using the same techniques as stated before. The results are shown in Table 2.1 to 2.3.

From the above results, we can see that our method generally performs better than previous methods. It is surprising to see that methods based on 3D structural correspondences such as SURF3D [71] and PFH [112] perform much worse than the other fine registration methods. The reason behind it might be that 3D structure-based features are not as reliable as photometric features in the point cloud present with noise. Also in situations when objects lack salient structural features or when there is a large view change between two point clouds, the 3D structure-based features

Table 2.2: RMS Error from the Ground Truth for Two Views with Less Overlap Regions. (Unit: mm)

Data	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Adaptive [160]	4.339	12.201	5.036	6.691	4.309	8.258
TriICP [22]	32.571	15.296	4.076	60.758	26.348	28.432
Reweight ICP [19]	3.703	12.592	4.599	5.002	4.919	7.902
Color Based [64]	25.456	13.033	**2.537**	44.688	19.390	20.004
SIFT Based [79]	5.832	12.486	5.507	10.291	4.081	10.608
ADMM [12]	3.728	24.275	3.047	58.319	17.329	7.818
Super4PC [91]	6.200	29.141	3.540	34.360	34.672	10.836
SURF3D [71]	30.197	29.836	4.489	34.456	34.073	34.179
PFH [112]	3.611	21.433	3.627	22.821	34.212	26.686
Ours	**2.840**	**11.586**	2.679	**3.780**	**3.855**	**2.953**

Table 2.3: RMS Error from the Ground Truth for General Cases with Distinctive Geometric Structures. (Unit: mm)

Data	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Adaptive [160]	4.249	2.724	3.034	3.932	**12.417**	459.867
TriICP [22]	4.118	1.768	2.836	3.302	53.676	7.799
Reweight ICP [19]	4.489	1.563	2.882	4.908	20.098	8.820
Color Based [64]	4.231	1.209	2.854	3.243	53.227	7.288
SIFT Based [79]	4.051	2.754	3.054	6.025	12.431	**6.264**
ADMM [12]	4.011	1.362	2.927	5.300	32.088	6.618
Super4PC [91]	16.577	5.360	6.210	4.050	29.884	6.248
SURF3D [71]	3.398	4.955	5.789	18.172	48.407	56.910
PFH [112]	8.475	4.729	4.649	10.953	37.842	9.213
Ours	**3.280**	**1.194**	**2.773**	**2.934**	12.805	6.345

often fail to extract salient points or find matches. The state-of-the-art methods such as ADMM and Super4PC generate large registration errors in the above situations as well since solely relying on the 3D structure is not enough.

We also conduct experiments on the computation comparison. Table 2.4 and 2.5 list the registration time comparisons for the food-can case (Figure 2.2) and the cereal box case (Figure 2.4), respectively. We implement our algorithms in MATLAB 2010b. The simulations are carried out using a machine with a dual-core 3.1 GHz Intel i3-2100 CPU, 8.0 GB RAM running 32-bit Machine. From the tables, we can see that the color-based ICP method [64] is more computationally intensive than the original ICP, since the closest distance search is performed for all the points and the distance combines both the 3D coordinates and the color channels. The SIFT-based method in [79] has a shorter processing time because in the ICP process only the key feature points are taken into account. The ADMM method in [12] is the most computationally intensive, as it introduces higher order metrics during optimization. In our method, we can see the time to pre-calculate the surface variation and the outlier rejection threshold is not significant. Also, since we have deleted some outliers in each iteration, the time spent on searching for closest distance decreases a lot, especially in the case of objects with less overlapping regions, which demonstrates the efficiency of our algorithm.

2.5 Conclusion

In this chapter, to improve the accuracy and robustness of the ICP algorithm, we introduced a regularization term incorporating the spatial distances of the SIFT feature pairs with dynamically adjusted weights to balance the errors in the error minimization process. We also proposed a new outlier rejection method which is based on dynamic thresholding and leveraged both the structure and sparse feature pairs from the texture of the RGB images as a constraint to keep the ICP iterations in the right convergent track. Simulation results demonstrate the effectiveness of the proposed approach compared to previous methods, and the robustness under challenging situations such as objects lacking structural features, significant camera view changes, and different lightings.

Table 2.4: Registration Time Comparison for the Food-can Case with Other ICP based Methods

Number of points: N=3214, M= 3200, after 18 iterations. (unit: sec.)			
Compared Method	Pre-process time (RANSAC, SIFT matching, etc.)	ICP process time	Total
TriICP [22]	0.720	0.287	1.007
Adaptive ICP [160]	0.720	0.299	1.019
Color based ICP [64]	0.921	0.855	1.776
SIFT based ICP [79]	0.723	0.129	**0.852**
Reweight ICP [19]	0.720	0.526	1.246
ADMM [12]	0.720	10.668	11.388
Ours	0.780	0.504	1.284

Table 2.5: Registration Time Comparison for the Cereal-box Case with Other ICP based Methods

Number of points: N=6390, M= 3413, after 18 iterations. (unit: sec.)			
Compared Method	Pre-process time (RANSAC, SIFT matching, etc.)	ICP process time	Total
TriICP [22]	1.042	1.396	2.438
Adaptive ICP [160]	1.042	1.401	2.443
Color based ICP [64]	1.183	2.086	3.269
SIFT based ICP [79]	1.114	1.110	2.224
Reweight ICP [19]	1.042	2.701	3.743
ADMM [12]	1.042	17.615	18.657
Ours	1.140	1.081	**2.221**

Chapter 3

SUPER-RESOLUTION FOR A SINGLE DEPTH IMAGE

3.1 Introduction

During recent years, we have witnessed a rapid progress in the field of 3D imaging. The birth of low-cost 3D scanning devices such as Microsoft Kinect and Time-of-Flight (TOF) cameras have opened the door for new applications in different research disciplines, including computer vision, graphics, human computer interaction, and virtual reality. However, the limited resolution and low quality of the depth map generated by these cameras still pose serious issues for various 3D applications. For example, the resolution of the SwissRange SR4000 depth camera and the PMD Camcube camera are only about 200×200. Even for Kinect, the resolution of the depth image is only 320×240 for Kinect v1 and 512×424 for Kinect v2, which is much lower than that of its corresponding color image (1280×960 at 12 fps for Kinect v1 and 1920×1080 at 30 fps for Kinect v2). In this chapter, we aim to enhance the resolution of depth images for 3D applications relying solely on a single depth image as input. Image super-resolution (SR) aims to reconstruct a high-resolution image from its low-resolution counterpart. It is a challenging task in the computer vision field. In its essence, image super-resolution requires the prediction of a large amount of unknown pixels based on the input pixels. To date, super-resolution is also intensively related to a variety of other problems such as image denoising, deblurring, or inpainting.

In this chapter, we address the problem of depth image super-resolution and denoising, which offers unique challenges that are different from color image super-resolution. The depth map captured by existing consumer cameras is usually degraded by noise due to inaccurate scanning hardware or difficulties in calculating the disparity. Although depth maps contain less texture compared to color images, human eyes are usually more sensitive to noise in 3D (when noise is along surface normal directions, it is more likely to be noticed compared to viewing in 2D), so the

artifacts produced from depth super-resolution will become less tolerable for 3D applications.

We propose two approaches from different aspects to achieve our goals: a Coupled Dictionary Learning–based approach with Local Constraint (CDLLC) and an Edge Guided approach (EG). In CDLLC, we propose a novel dictionary learning-based algorithm for the single depth image super-resolution problem. We add local constraints into the coupled dictionary learning and reconstruction process to remove the prediction uncertainty and to prevent the dictionary from overfitting. We also tackle the jagged noise problems in depth image super-resolution by incorporating an adaptively regularized Shock filter. We show that since the Shock filter can simultaneously clean up the jagged noise and sharpen the edges, it is particularly suitable for de-noising the depth map, which contains less texture information. Furthermore, we propose to jointly reconstruct and smooth the high-resolution image using an L0 gradient smooth constraint. We perform the reconstruction and L0 smoothing in an iterative scheme so that the reconstructed high-resolution depth image is more robust to noise.

In EG, we propose a novel framework for single depth image super-resolution guided by a reconstructed high-resolution edge map. We convert the super-resolution problem from high-resolution texture prediction to high-resolution edge prediction, which is motivated by the essence that edges are of particular importance in the textureless depth image. We also explore the self-similarity of patches during the edge construction stage, when limited training data is available. Guided by the predicted high-resolution edge map, a modified joint bilateral filter is applied to reconstruct the high-resolution depth textures.

From our experimental results, the CDLLC approach better handles the noise present in the depth map and has slightly better performance regarding the RMSE measurement, while the EG method performs better in terms of percent of error in the depth map as well as the visual result. However, the EG method requires significantly more computation compared to CDLLC.

3.2 Related Work

Single Image Super-Resolution. Image super-resolution is one of the most active research topics in the field of image processing. Among them, single image super-resolution has been widely

studied. For instance, the example-based approach is one of the most popular methods for single image super-resolution [40, 128, 149]. In [18], locally linear embedding is applied, in which a training dataset including low-resolution and high-resolution image pairs is collected. For each patch in the input low-resolution image, similar patches from the training dataset are searched and the corresponding high-resolution patches are then used to reconstruct the high-resolution output. In [40], image super-resolution is formulated as a Markov Random Field labeling problem, with each hidden node representing the label of a high-resolution patch. However, since each image patch is directly obtained externally, the reconstruction is highly reliant on the similarity between the input and the patches in the dataset, and is highly biased towards examples in the dataset. This method could also result in blur or discontinuity artifacts.

Recently, several dictionary learning-based methods have been shown to outperform the classical example-based approach [17, 42, 68, 139, 152, 153, 158]. In [17], high-resolution and low-resolution patches are reconstructed as sparse linear combinations of learned coupled dictionary atoms based on the assumption that low-resolution and high-resolution patches should share the same reconstruction coefficients. In [158], the work of [153] is extended by using a K-SVD dictionary training procedure, which achieves significantly better results. The work of [139] and [68] relax the fully-coupled constraint and learn a mapping through low-resolution and high-resolution image pairs. However, blurry and ringing effects near the edges exist in their SR results. In [152], it is proposed to split the feature space into subspaces and learn a low-resolution and high-resolution mapping in each subspace. The work of [154] presents a novel self-learning super-resolution approach, in which support vector regression is explored with sparse representation. Without the collection of low-resolution and high-resolution training images nor the prior information of self-similarity, a SVR model is learned by minimizing an error function to produce the high-resolution reconstruction. In [31], a deep learning method based on the convolutional network for single image super-resolution is proposed (SRCNN), in which the mapping between the low-resolution and high-resolution images is learned. However, sometimes it is difficult to learn this mapping, which is many to one, yielding reconstruction problems such as blurry or ringing artifacts.

Self-similarity has also been explored for image super-resolution. These methods are based on

the fractal nature of images [164], which suggests that patches of a natural image recur within and across scales of the same image. The work of [46] proposes to search for similar image patches across multiple downscaled versions of the image with no need to collect training image data beforehand. More recently, in [55], it has been proposed to expand the internal patch search space by allowing geometric variations (SRF).

In addition to patch synthesis based approaches, edge or texture-based methods have been proposed. In [123], a gradient profile prior is proposed to improve color image super-resolution. In [125], a multi-scale edge representation is introduced to direct the color image super-resolution, combined with a tensor voting strategy. In [50], to overcome the problems of the direct patch synthesis, texture recognition is used to aid image upsampling. However, these works still generate blurry artifacts at the edges and are not suitable for the depth image super-resolution scenario.

Depth Super-Resolution with Multiple Images. Traditional depth super-resolution methods are focused on fusing multiple low-resolution depth maps to get a high-resolution depth image [28, 58, 107, 113, 116]. The multiple frames can provide complementary spatial-temporal information and help fill in missing or noisy parts of each single image [78, 120], making the upsampling result more robust and reliable. For example, the Lidarboost approach [116] combines multiple range images in an optimization framework with data fidelity and geometry prior to produce a high-resolution depth map. The Kinect Fusion algorithm [58] generates superior 3D reconstruction results by registering a sequence of overlapped depth maps captured by a Kinect camera. In [28], based on a probabilistic alignment algorithm, high quality 3D shapes could be achieved by fusing multiple low-resolution and noisy scans. However, despite the good results obtained by these methods, they heavily rely on the assumption that multiple static range images are available with small camera movement, which may not be true for many practical applications. Moreover, the quality of the super-resolution result is also sensitive to the camera pose estimation errors.

Color Assisted Depth Super-Resolution. It also has been proposed to use a pre-aligned high-resolution color image to help upscale the depth map since the high frequency components in color images such as edges can be utilized to assist the depth pixel prediction. For instance, in [37, 73, 84, 85, 155], joint color and depth upsampling approaches are proposed based on the edge

information from the high-resolution color image. In [103], a nonlocal means filter (NLM) is used to regularize the depth image in order to maintain the detailed structure. In [24], a region segmentation-based method is proposed to tackle the texture-transfer and depth-bleeding artifacts. In [37], an anisotropic diffusion tensor, calculated from a high-resolution intensity image, is used to guide the depth image upsampling (TGVL2). However, notwithstanding the appealing results that such approaches could generate, in many cases, the high-resolution color image fully registered with the depth image may not be always available, which makes the color assisted approaches less general.

Single Depth Image Super-Resolution. Single depth image super-resolution offers unique challenges compared to color image super-resolution (e.g., edge-preserving denoising should also be properly tackled). The work of [88] and [30] extends [40] to the depth domain by using a patch-based MRF model (PB). The method proposed in [54] searches for high-resolution patches by identifying self-similar 3D patch correspondences via a rigid body transformation. In [81], the patchwork assembly framework in [88] extends by adding geometric constraints from self-similar structures. Although they get promising super-resolution performances, there is no guarantee that patch redundancy always exists within or across image scales. This is especially the case for the depth image, which has a starting low resolution and contains less unique texture patterns. Thus, it reduces the generality of the depth super-resolution framework.

Techniques based on dictionary and sparse representation of depth images have been applied for depth restoration such as de-noising and inpainting [89, 130, 131]. In [130], sparse priors are learned from the data corrupted with spatially varying noise. The reconstructed clean depth map can then be inferred by the learned sparse priors. In [89], the framework of learning the sparse representation is applied to fill missing data in the 3D surface. The work of [131] extends this method for joint intensity and depth estimation based on the sparse dictionary learning algorithm.

However, notwithstanding the demonstrated success of dictionary learning based approaches, existing methods still have some problems especially in the case of depth super-resolution. Since in depth images, which are relatively textureless, the mapping between high-resolution and low-resolution patches tends to be many to one. Therefore, basis learning methods may suffer from

over-fitting, which causes similar low-resolution patches to produce very different high-resolution counterparts. Also, it is not feasible to represent this one-to-many relationship via a simple mapping function since it will cause the fundamental ambiguity that the learned bases with least errors to represent the low-resolution patch may not always produce the best reconstruction result for the high-resolution patch in the testing phase. In 3D applications, the resultant artifacts visually become more severe since human eyes are sensitive to 3D noise.

Depth Image De-noising. Another existing problem in depth super-resolution is depth de-noising. According to previous works [39, 80], the noise contained in depth images—such as fluctuating pixel values at the depth discontinuity and random noise with the variance depending on the intensity and distance—can be characterized as boundary noise. Therefore, directly upsampling depth images will magnify the noise, producing artifacts along edges. There are many previous works on depth image de-noising such as [41, 56]. For the purpose of super-resolution, in order to preserve depth edges, a bilateral filter is utilized in the pre-processing step for noise reduction in [88]. However, from our observations, not only the noise, but also the jagged artifacts around depth discontinuities caused by inaccurate sampling and heavy quantization of the disparity in the original low-resolution image are magnified. In our work, we will deal with all the noise or artifacts that are mentioned above during the depth super-resolution process.

Edge Aware Image Smoothing. In recent years, methods have been proposed for edge-aware image smoothing, which can effectively sharpen major edges by increasing the steepness of transitions while eliminating a manageable degree of low-amplitude structures [4, 27, 34, 35, 102, 151]. In [151], an L0 gradient constraint is used to approximate the image structure. In [27, 34], geodesic and diffusion distances are introduced to describe the color difference in the smoothing process. In [4, 102], a local Laplacian filter is utilized to address the halo artifacts in manipulating image multi-scale details. Although these methods are mainly used for image editing and abstraction, we have found that they are particularly suitable for edge-preserving depth image de-noising. Although performing edge-preserving smoothing could heavily smooth regions with texture, the depth image usually contains less texture information. Thus, it will remove noise such as ringing or blurry artifacts around edges in the depth image, but not degrade its quality much.

As an introduction, in our first method, we follow the dictionary learning-based approach but add a local constraint into the coupled dictionary learning process which better preserves the manifold assumption and prevents the dictionary from over-fitting. Besides, we propose to jointly reconstruct and smooth the high-resolution image using an L0 gradient smoothing constraint. We also apply an adaptively regularized Shock filter to tackle the depth image jagged noise while simultaneously reducing noise and sharpening edges.

In the second approach, we utilize the joint bilateral filtering idea, but instead of using a pre-aligned high-resolution color image, we estimate a high-resolution edge map and use the reconstructed edge map to guide the upsampling of the depth image.

We show that our approaches lead to better results compared to previously reported methods. [143–146].

3.3 Single Depth Image Super-Resolution and De-noising via Coupled Dictionary Learning with Local Constraints and Adaptive Shock Filtering (CDLLC)

In the following discussion, we denote upsampling the image by $g \times g$ as upscaling by a factor of g. We denote $S = [S^l, S^h]$ as the input low-resolution image and its synthesized high-resolution counterpart. We denote $\mathbf{x}_i = [\mathbf{x}_i^l, \mathbf{x}_i^h]$ as the paired feature vector of the ith patch in S^l and S^h. We will further discuss the extracted features in Section 3.3.6. $\mathbf{D} = [\mathbf{D}^l, \mathbf{D}^h]$ is the coupled dictionary that contains N atoms. $\mathbf{D}_k$ is the kth atom in the learned coupled dictionary ($k = 1, 2, \ldots, N$). $\mathbf{c}_i$ is the coefficient vector with length N, containing the weights of each dictionary atom for synthesizing $\mathbf{x}_i$.

3.3.1 Coupled Dictionary Learning Based on Locality Coordinate Coding (LCC)

In this part, we propose a coupled dictionary learning approach with a locality constraint for depth image super-resolution. Our algorithm is patch-based in which we treat training and testing images as overlapping patches with the same size ($m \times n$ after upscaling). We synthesize the patches as sparse linear combinations of the learned dictionary bases. The most important issue is to find the effective dictionary bases for the patch prediction. Generally, in the depth maps, which are

simpler than natural images in terms of texture, it is easier to reconstruct patches as combinations of several representative geometry priors. However, it also leads to the proneness of dictionary over-fitting: similar low-resolution patches represented by the learned dictionary bases with the least error may produce significantly different high–resolution patches during the testing phase. Therefore, inspired by the sparsity and the nature of Local Coordinate Coding (LCC) [156], we could benefit the dictionary learning with a locality constraint to alleviate this problem. Although it was proposed in [133] to approximate the locality constraint in image super-resolution by using a fast implementation called Locality Linear Coding, it has been proved [161] that this online incremental codebook learning algorithm has a performance close to K-Means, and the sparsity cannot be fully utilized especially in the depth image super-resolution scenario.

3.3.2 Coupled Dictionary Learning for Image Super-resolution

To learn the relationship between paired high-resolution and low-resolution data, in the sparse coding scheme [153], it is proposed to learn a coupled dictionary and reconstruct the high-resolution patches by minimizing the objective function with a sparse regularizer:

$$\min_{\mathbf{D},\mathbf{c}} \sum_{i} (\|\mathbf{x}_i - \mathbf{D} \cdot \mathbf{c}_i\|^2 + \lambda \sum_{j} \|\mathbf{c}_{i,j}\|_1) \tag{3.1}$$

where λ is a weighting constant, and $\mathbf{c}_{i,j}$ is the jth component of coefficient vector $\mathbf{c}_i$. In the above equation, the first term is to minimize the error of dictionary approximation. The second term is a sparsity regularizer.

In the reconstruction stage, given the input low-resolution patch feature $\mathbf{x}$, its coefficients of the dictionary basis $\mathbf{c}_i$ are computed by

$$\min_{\mathbf{c}_i} \|\mathbf{x}_i^l - \mathbf{D}^l \cdot \mathbf{c}_i\|^2 + \lambda \|\mathbf{c}_i\|_1. \tag{3.2}$$

Under the assumption of common sparse representation between the low-resolution and high-resolution patches, we can reconstruct $\mathbf{x}_i^h$ as

$$\mathbf{x}_i^h = \mathbf{D}^h \cdot \mathbf{c}_i. \tag{3.3}$$

3.3.3 Proposed Coupled Dictionary Learning with LCC (DLLCC)

The introduction of a coupled dictionary is to learn the relationship between paired high-resolution and low-resolution data under the assumption that pairing patches should have common sparse representation. The motivation of LCC is to find the dictionary bases without violating the locality constraint. Our coupled dictionary learning with LCC can be written as:

$$\min_{\mathbf{D},\mathbf{c}} \sum_i (\|\mathbf{x}_i - \mathbf{D} \cdot \mathbf{c}_i\|^2 + \lambda \sum_j \|\mathbf{D}_j - \mathbf{x}_i\|^2 \|\mathbf{c}_{i,j}\|_1). \tag{3.4}$$

In Eqn. 3.4, the second term constrains the dictionary atoms with non-zero coefficients to be similar with the input patch. Since the linear combination of dictionary atoms is based on the assumption that small image patches form manifolds with similar local geometries in the feature space [156], the second term better preserves the manifold assumption and keeps the locality constraint. More importantly, with the locality constraint, for each low-resolution patch, only the dictionary bases which are more similar to it are selected, effectively preventing the dictionary from over-fitting. The detail of the non-convex optimization is shown in Section 3.3.6.

3.3.4 Joint Reconstruction based on L0 Constraint (JRL0)

In the reconstruction stage, we jointly reconstruct and smooth the high-resolution patches without losing the edge structural information by optimizing the following energy function as

$$\min_{\mathbf{c},\mathbf{S}^h} \sum_i (\|\mathbf{x}_i^l - \mathbf{D}^l \cdot \mathbf{c}_i\|^2 + \lambda \sum_j \|\mathbf{D}_j^l - \mathbf{x}_i^l\|^2 \|\mathbf{c}_{i,j}\|_1) + w\|\mathbf{S}^h - R_i \bigoplus B(\mathbf{D}_i^h \cdot \mathbf{c}_i)\|^2 + \varphi\|\bigtriangledown \mathbf{S}^h\|_0 \tag{3.5}$$

where B is an operator that transfers the patch from the feature space to the intensity space as defined in the next section. R_i is the operator that puts together the high-resolution intensity patches $B(\mathbf{D}_i^h \cdot \mathbf{c}_i)$ into a full-resolution image (by averaging the pixel values in the overlapped regions

between patches). $R_i \bigoplus B(*)$ means that we first convert the high-resolution patches in the feature space to the intensity space and then put those patches together to get a full-resolution depth image. λ, w and φ are constant weights.

We add the L0 norm gradient term to jointly smooth and reconstruct the depth image with sharp edges. The L0 gradient constraint helps assemble the smoothed latent high-resolution image and get the optimal coefficients during the reconstruction. As a result, it alleviates the reconstruction error caused by the one-to-many mapping and also makes the reconstruction more robust to noise. The joint reconstruction and L0 smoothing result is shown in Figure 3.1a to 3.1d. From the result, we can see that the ringing artifacts along the edge are reduced after joint smoothing and reconstruction. It should be noted that since L0 smoothing trades off detail-flattening with sharp edge preservation, directly applying L0 smoothing on images might produce flattening artifacts in regions with gradually changing intensity. However, in our approach, instead, the L0 smoothing is used as a constraint to get the optimal coefficients, which makes the reconstruction less vulnerable to noise and not affected by the flattening effects. We will discuss the details about optimizing the non-convex function of Eqn. 3.5 in Section 3.3.6. In our approach, we choose L0 gradient smoothing instead of other edge-aware smoothing methods because our purpose is to demonstrate that joint edge-preserving smoothing and high-resolution image reconstruction improve the super-resolution and de-noising results for depth images. Our joint framework can also flexibly incorporate other state-of-the-art edge-aware smoothing methods such as geodesic based smoothing and local Laplacian filters [27] [102].

3.3.5 Edge Denoising based on an Adaptive Shock Filter (ASF)

As stated before, human eyes are more sensitive to noise in 3D. Moreover, the depth maps captured by TOF cameras or even expensive laser scanners may still produce jagged edges along depth discontinuities due to quantization or measurement errors (Figure 3.2b). If we directly apply the algorithm that we discussed before to a raw depth image, those jagged artifacts will be magnified (Figure 3.2c) and are often unacceptable in 3D.

In our algorithm, we incorporate a smoothing Shock filter as pre-processing to remove the

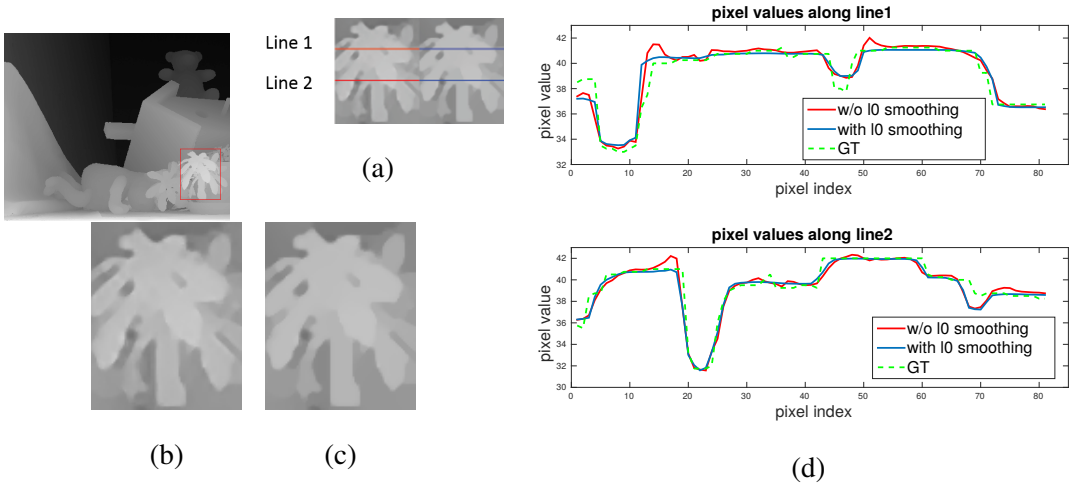


Figure 3.1: **Results of using L0 gradient constraint.** (a) Depth image sample. (b) Visual result without using L0 gradient constraint for reconstruction. (c) Visual result using L0 gradient constraint for reconstruction. (d) Top: line1 samples in (a); Bottom: line2 samples in (a). (Best viewed in color.)

artifacts around edges. A Shock filter is a morphological method to iteratively enhance image edges based on partial differential equations [98,126,135]. It is widely used in image denoising and deblurring [23, 126, 135]. Compared to other edge preserving methods such as the bilateral filter, the Shock filter also has the advantage of keeping edges non-oscillatory, which can effectively remove the jagged noise in our scenario. It should be noted that the Shock filter sometimes may degrade the contrast of image texture. However, in the application of denoising the depth image, which does not have much texture information, this distortion is not serious. In our approach, we use the regularized Shock filter similar to [45], which not only enhances edges anisotropically, but smooths out noise as well:

$$S_t = -\frac{2}{\pi}\arctan(a \cdot \mathrm{Im}(\frac{S}{\theta}))\,|\nabla S| + \alpha S_{\eta\eta} + \boldsymbol{\beta} \odot S_{\xi\xi} \tag{3.6}$$

where a, θ and α are tuning constants and $\boldsymbol{\beta}$ is adaptively adjusted for each pixel, which will be

explained in Eqn. 3.7. $\odot$ is the element-wise multiplication operator. S is the original image going through the complex diffusion process. Im stands for the imaginary operator. $S_{\eta\eta}$ and $S_{\xi\xi}$ stand for diffusion in the normal and tangent direction. S_t is the evolution of image S at iteration t. The first part is the basic Shock term for edge enhancement and the latter two terms are regularizers for noise removal, in which α and $\boldsymbol{\beta}$ control the smoothed diffusion in the gradient and tangent direction respectively.

In our approach, we modify the above equation by adaptively changing the weights of the diffusion terms. Since we want to preserve the edge as much as possible, we therefore minimize the impact of the diffusion in the gradient direction by setting its weight (α) to be very small (α = 0.01 in our algorithm). Smoothing in the tangent direction can effectively alleviate jagged noise but as we do not want to over-smooth other regions, the value of $\boldsymbol{\beta}$ is adaptively adjusted as follows:

$$\boldsymbol{\beta}_{i,j} = \begin{cases} 3K, & \text{if } S(i,j) \in \text{jagged region without corner region;} \\ K, & \text{otherwise.} \end{cases} \tag{3.7}$$

The jagged region is defined as follows: we first adaptively subdivide the depth image using a quadtree structure. Each non-leaf node of the quadtree has four children that subdivide the space into four quadrants based on pixel value differences. The larger pixel value difference, the more sub-nodes will be produced. The division is terminated until the size of the leaf node is less than s by s. As a result, the maximum subdivision occurs along the edges. We take the region containing nodes generated in the last two subdivisions with the smallest and second smallest size ($[s, s]$ and $[2s, 2s]$) as the potential jagged region, as shown in red in Figure 3.3a. Since corners should not be over-smoothed, we exclude the corner regions from the extracted jagged regions using Harris corner detection (result of excluding corners is shown in blue in 3.3a). It should be noted that we can also use an edge detection algorithm to extract the jagged region. However, from experiments we found that the filtering result after the quadtree subdivision is more consistent than that of edge detection algorithms because the quadtree subdivision considers homogeneous regions instead of single pixels. After Shock filtering, the smooth parts are de-noised while the edges are enhanced (Figure 3.2e). It should be noted that in the experiments, we set α and $\boldsymbol{\beta}$ small so that

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

Figure 3.2: **Jagged artifacts and Shock filter result.** (a) Original low-resolution image. (b) A patch from (a). (c) The same patch from the high-resolution image directly reconstructed by DLLCC. (d) Using bilateral filter in the pre-processing and (e) Using Shock filter in the pre-processing.

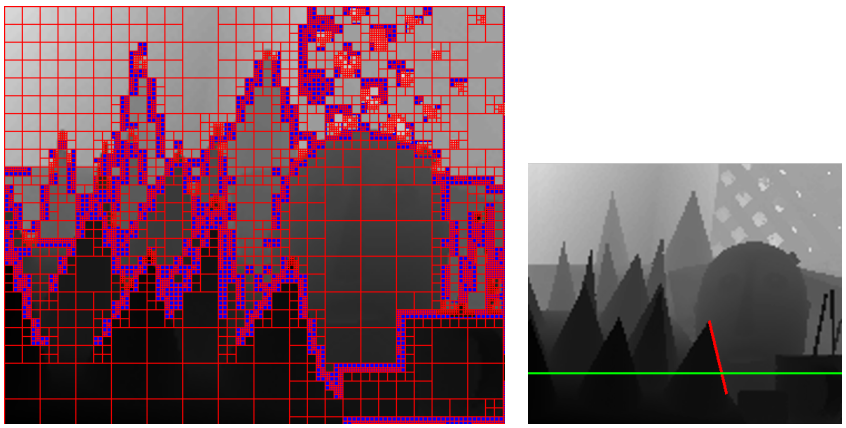
the smooth parts will not be over smoothed. In Figure 3.3b, we extract two lines (red and green) in two directions and plot the corresponding pixel values variations in Figure 3.3c. From it, we can see that the Shock filter can effectively smooth the jagged edges while well preserve the image structure. Simply using the bilateral filter, however, cannot remove the jagged edge artifacts as shown in Figure 3.2d.

3.3.6 Implementation Details of CDLLC

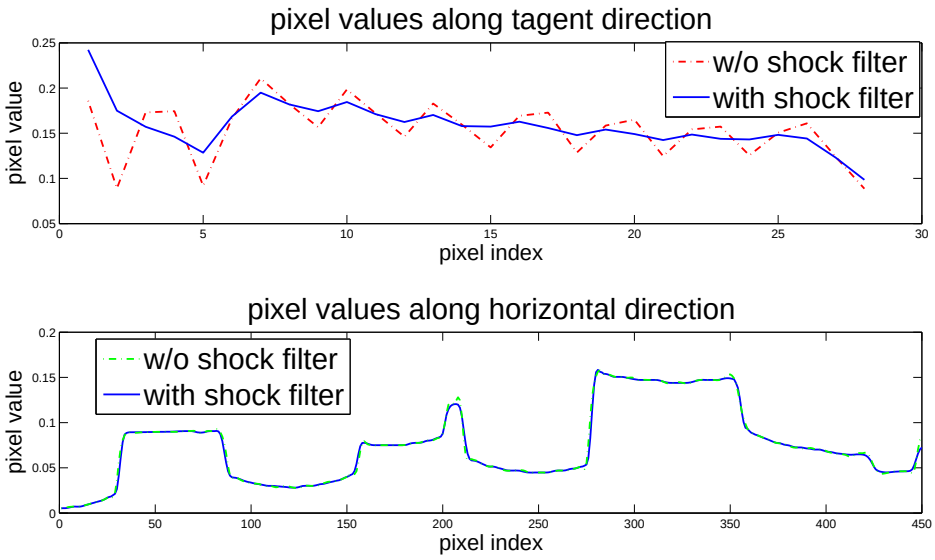
The complete algorithm we propose for depth image super-resolution is summarized as follows: Given an input low-resolution depth image, we want to upscale it by g.

A. Pre-Processing

We first apply an iterative bilateral filter (with window size 21×21, and $\sigma_d = 3, \sigma_{intensity} = 15$), to fill up holes in the low-resolution depth image. Then, we perform the regularized Shock filter to remove the jagged edge artifacts and noise in the depth image. We further upscale the filtered result using bilinear interpolation to produce a pre-processed image $\mathbf{S}^m$ to the destination size, as the low-frequency component of $\mathbf{S}^h$.



(a) (b)



(c)

Figure 3.3: **Results of using Shock filter.** (a) Quadtree subdivision. (b) Pixel values along two lines are sampled. (c) Top: red line samples in (b). Bottom: green line samples in (b). (Best viewed in color.)

B. Coupled Dictionary Learning Optimization

The feature extraction process is employed on the full image. Similar to [153], we compute the 1st and 2nd-order derivatives of $\mathbf{S}^m$ as the features of the low-resolution image $\mathbf{S}^l$ (four filter responses considering the derivatives in both the horizontal and vertical directions). The features of high-resolution image $\mathbf{S}^h$ are extracted by removing its low frequency component $\mathbf{S}^m$. Feature patches $\mathbf{x}$ are then extracted. Reversely, operator B (an operator that transfers the patch from the feature space to the intensity range) in Eqn. 3.5 is defined as:

$$\mathbf{y}^h = B(\mathbf{x}^h) = \mathbf{x}^h + \mathbf{y}^m \tag{3.8}$$

where $\mathbf{y} = [\mathbf{y}^l, \mathbf{y}^h]$ is denoted as the paired high-resolution and low-resolution image patches and $\mathbf{y}^m$ as the low-frequency patches.

Therefore, the dimensions of patch $\mathbf{x}_i^l$ and $\mathbf{x}_i^h$ are $4mn$ and mn, respectively. The dimension of dictionary $\mathbf{D}^l$ and $\mathbf{D}^h$ is $N \times 4mn$ and $N \times mn$, respectively. N is the number of atoms in dictionary $\mathbf{D}$, and $[m, n]$ is the patch dimension.

We then learn a coupled dictionary with the locality constraint by minimizing Eqn. 3.4. Since Eqn. 3.4 is non-convex, we iteratively update $\mathbf{D}$ and $\mathbf{c}$ until it converges. The process is summarized in Algorithm 1.

C. Super-Resolution Reconstruction

Once the coupled dictionary is learned, in testing, we extract the features of the low-resolution patches in the same way (by upscaling it to the same size with the high-resolution image using bilinear interpolation) as discussed above.

To simultaneously find the optimal coefficients $\mathbf{c}_i$ and $\mathbf{S}^h$, we modify the cost function of Eqn. 3.5 by introducing an auxiliary variable $\mathbf{h}_i$ as:

$$\min_{\mathbf{c},\mathbf{S}^h} \sum_i (\|\mathbf{x}_i^l - \mathbf{D}^l \cdot \mathbf{c}_i\|^2 + \lambda \sum_j \|\mathbf{D}_j^l - \mathbf{x}_i^l\|^2 \|\mathbf{c}_{i,j}\|_1) + w \cdot \|\mathbf{h}_i - B(\mathbf{D}^h \cdot \mathbf{c}_i)\|^2) + w \cdot \|\mathbf{S}^h - R_i(\mathbf{h}_i)\|^2 + \varphi \|\bigtriangledown \mathbf{S}^h\|_0. \tag{3.9}$$

By splitting Eqn. 3.9 into two new defined functions as follows, we update S^h, $\mathbf{h}$ and $\mathbf{c}$ in Eqn. 3.9 in an alternating scheme (Algorithm 2):

$$f(\mathbf{h}_i, \mathbf{c}) = \sum_i \|\mathbf{x}_i^l - \mathbf{D}^l \cdot \mathbf{c}_i\|^2 + \lambda \sum_j \|\mathbf{D}_j^l - \mathbf{x}_i^l\|^2 \|\mathbf{c}_{i,j}\|_1) + w \cdot \|\mathbf{h}_i - B(\mathbf{D}^h \cdot \mathbf{c}_i)\|^2, \quad (3.10)$$

$$g(\mathbf{h}_i, \mathbf{S}^h) = w \cdot \|\mathbf{S}^h - R_i(\mathbf{h}_i)\|^2 + \varphi \|\bigtriangledown \mathbf{S}^h\|_0. \quad (3.11)$$

Algorithm 1: Learning a coupled dictionary with LCC

Input: Coupled low resolution and high resolution patch $\mathbf{x}$ (in the feature space)

Output: Learned coupled dictionary D with size of $(4mn + mn)N$

Initialization:

Randomly choose N patches and assign to $\mathbf{D}_1$ to $\mathbf{D}_N$.

for $t \leftarrow 1$ **to** T **do**

Update c **:**

Fix $\mathbf{D}$, minimize Eqn. 3.4 by solving a Lasso problem for each $\mathbf{c}_i$:

$$\mathbf{c}_i^* = \mathbf{\Omega}_i^{-1} \cdot \arg\min_{\boldsymbol{\alpha}_i} (\|\mathbf{x}_i - \mathbf{D}(\mathbf{\Omega}_i^{-1}\boldsymbol{\alpha}_i)\|^2 + \lambda \|\boldsymbol{\alpha}_i\|^1) \quad (3.12)$$

where $\mathbf{c}_i^*$ is the optimal value of c_i, $\mathbf{\Omega}_i = \mathit{diag}(\|\mathbf{D} - \mathbf{x}_i\|^2)$ and $\boldsymbol{\alpha}_i = \mathbf{\Omega}_i \cdot \mathbf{c}_i$

Update $\mathbf{D}$ **:**

Fix $\mathbf{c}$, minimize Eqn. 3.4 using gradient descent to get the optimal $\mathbf{D}$.

end

Return $\mathbf{D}$

Optimization. We use a linear regression with L0 norm regularization to approximate Eqn. 3.14 and use the Orthogonal Matching Pursuit (OMP) algorithm [132] to compute the optimal coefficients. It has been shown that L0 minimization based on OMP on the entire image performs generally faster than L1 minimization [132] for each patch. We have also tried the L1 optimization for Eqn. 3.14. However, despite the fact that OMP is greedy and suboptimal, we notice the

Algorithm 2: Computing the optimal HR image $\mathbf{S}^h$

Input: Learned coupled dictionary $\mathbf{D}$ with size of $(4mn + mn)N$

Output: High resolution image $\mathbf{S}^h$

Initialization:

Minimize Eqn. 3.2 to get the optimal coefficients $\mathbf{c}_i^0$ and compute the initial reconstructed $\mathbf{h}_i^0$ from Eqn. 3.3 as $\mathbf{h}_i^0 = B(\mathbf{D}^h \cdot \mathbf{c}_i^0)$.

for $t \leftarrow 1$ **to** T **do**

Update $\mathbf{S}_i^h$:

$$\mathbf{S}^h = \min_{\mathbf{S}^h} g(\mathbf{S}^h, R_i(\mathbf{h}_i^{t-1})); \text{ (}\textit{Smoothing}\text{)} \tag{3.13}$$

$$\mathbf{h}^t \leftarrow \mathbf{S}^h;$$

Update $\mathbf{c}^t$:

$$\mathbf{c}^t = \min_{\mathbf{c}} f(\mathbf{c}, \mathbf{h}_i^{t-1}); \text{ (}\textit{Reconstruction}\text{)} \tag{3.14}$$

$$\mathbf{h}_i^t \leftarrow B(\mathbf{D}_i^h \cdot \mathbf{c}_i^t);$$

end

Return $\mathbf{S}_h$

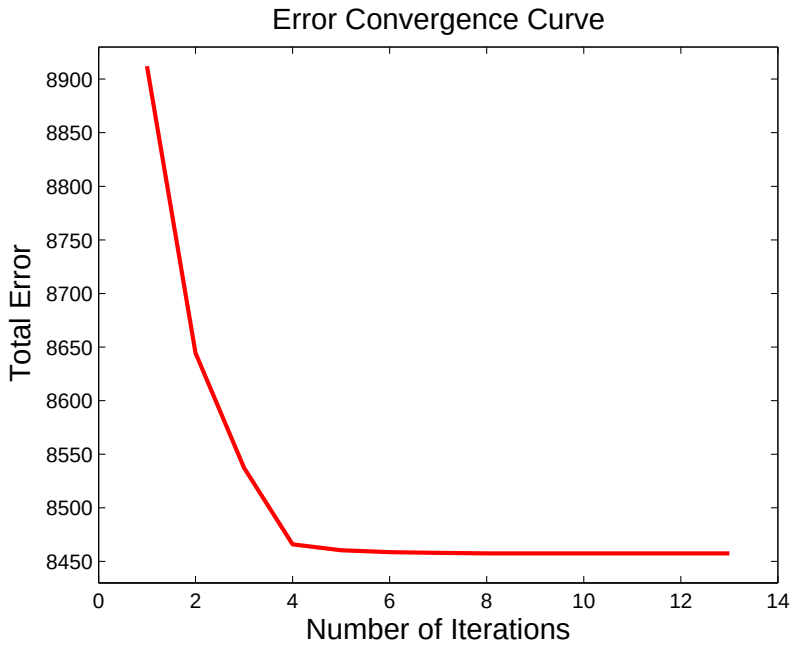


Figure 3.4: Error convergence of our joint smoothing and reconstruction method.

difference in the results is very small (in the order of about 1% to 2% better for L1 optimization in terms of Root Mean Square Error (RMSE) comparison).

For Eqn. 3.13, we use the L0 optimizer mentioned in [151] to solve the L0 gradient constrained optimization. Figure 3.4 shows the convergence reaching to a local minimum of the alternating joint reconstruction and smoothing algorithm. From it, we can see that this algorithm quickly converges in about 6 to 7 iterations.

3.4 Edge-Guided Single Depth Image Super-Resolution (EG)

In this section, we propose a novel framework for single depth image super-resolution guided by a reconstructed high-resolution edge map. The algorithm is motivated by the color–assisted, joint up-sampling approaches, in which the high-resolution color image provides edge guidance so that pixels in a local region with different depth values can be weighted differently in the upsampling process. Compared to the exemplar or learning-based super-resolution methods, the guidance of

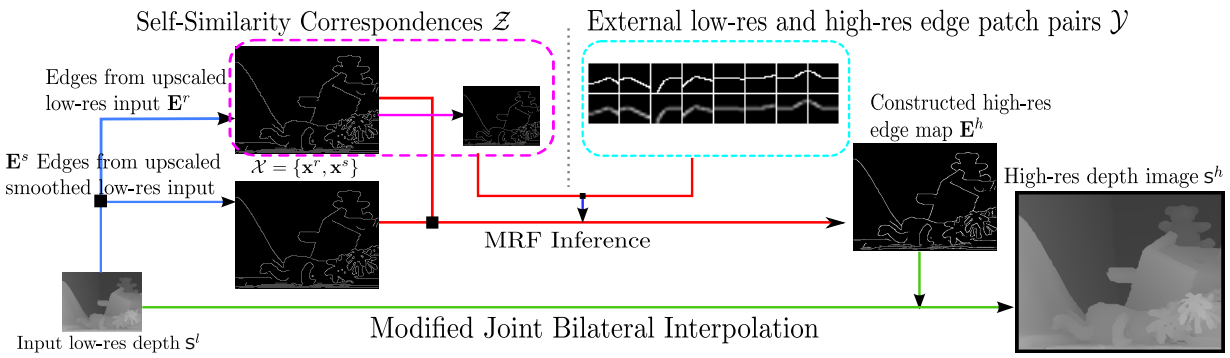


Figure 3.5: **Overview of the proposed method.** Given an input low resolution depth image, we first extract edges from it (blue). Along with an external (cyan) or internal (from self similarity, (megenta)) dataset including high resolution and low resolution edges patch pairs, we can infer a high resolution edge map via an MRF framework (red). Guided by the constructed high resolution edge map, depth values are interpolated via a modified joint bilateral filter to obtain a high resolution depth image (green).

high-resolution edges can alleviate artifacts such as blurring or ringing around edges, which are generated by direct depth value prediction. Furthermore, the constructed high-resolution edge map is smoothed without jagged artifacts, which could be magnified by learning-based methods. Thus, guided by the edge map, the high-resolution depth image will contain sharp and smooth edges (along the boundary). An overview of our proposed method is shown in Figure 3.5. In our proposed framework, we first extract edges from the low-resolution depth image. We then infer a high-resolution edge map based on MRF using either an external dataset or an internal (from self-similarity) patch collection composed by high-resolution and low-resolution edge patch pairs. Finally, guided by the constructed high-resolution edge map, high-resolution depth values are interpolated via a modified joint bilateral filter.

3.4.1 Learning a High-Resolution Edge Map from Low-Resolution Edges

Given the low-resolution depth image $\mathbf{S}^l$, we first apply bicubic interpolation to upscale $\mathbf{S}^l$ to the same resolution of $\mathbf{S}^h$ and then extract edges using the Canny edge detector to obtain an edge map $\mathbf{E}^r$. We find that the Canny edge detector is robust in our application. To ensure that most of the depth discontinuities can be found, we intentionally set the threshold of the detector low, so that more edges could be extracted. We also tried higher level contour detection algorithms such as gPb [100], but we obtained very similar results. This is due to the fact that the depth image has relatively less texture compared with color images; thus, edges extracted from different algorithms do not differ much. From Figure 3.6b, we can see that the edges extracted from the bicubic interpolated depth is not smooth and contains jagged edges, which will yield significant artifacts when used to guide the depth image interpolation. To have a higher quality high-resolution edge map, we apply a Shock filter [45] as a post-processing step on the bicubic interpolated depth map before edge detection. And as a result, edges are regularized to be straight, which is defined as E_s. However, the regularized edge map still contains wavy pattern artifacts around edges (Figure 3.6c). Therefore, together with $\mathbf{E}^r$ and prior knowledge from an external training dataset, we will refine $\mathbf{E}^s$ into a smooth high-resolution edge map $\mathbf{E}^h$ described in the following.

Given a jagged edge map $\mathbf{E}^r$ and the smoothed edge map $\mathbf{E}^s$, we construct $\mathbf{E}^h$ via a Markov Random Field (MRF) framework in a patch-based manner: for each edge pixel p_i in $\mathbf{E}^r$, we extract patches with the size of w by w pixels in $\mathbf{E}^r$ and $\mathbf{E}^s$, which are centered at p_i, and we denote the patches as $\mathbf{x}_i^r$ and $\mathbf{x}_i^s$, respectively. To reduce the computation time (i.e., reduce the graph size), we only extract patches with a small overlap area with each other (i.e., less than $1/5$ of the patch area). We denote $\mathcal{X} = \{\mathbf{x}^r, \mathbf{x}^s\}$ as a collection of stacked patches $\mathbf{x}^r$ and $\mathbf{x}^s$. In the external dataset we obtain $w \times w$ (jagged and not smooth) edge patches $\mathbf{y}_i^r$ from the given low-resolution images and the smooth edge patches $\mathbf{y}_i^s$ from the given high-resolution images. In addition, we also add the rotated patches of $\mathbf{y}_i^r$ and $\mathbf{y}_i^s$ into the dataset in order to make the patch data more complete. We denote $\mathcal{Y} = \{\mathbf{y^r}, \mathbf{y^s}\}$, containing a collection of stacked patches $\mathbf{y^r}$ and $\mathbf{y^s}$.

The basic idea of MRF is to obtain high-resolution edge patches from the external dataset under

some likelihood and coherence constraints. Instead of directly obtaining depth intensity patches, our intuition is based on the fact that binary edge patterns are much simpler than intensity patterns especially in depth images, leading to a smaller searching dimension. Thus, it could give better matches for the edge patch, making the reconstruction less biased to the dataset. In our Markov grid model, each $\mathcal{X}_i$ forms the node, and the hidden label corresponds to an edge patch $\mathcal{Y}_i$ from the dataset.

More formally, for each patch $i \in \mathcal{X}$, we specify a variable l_i denoting the indices $\{1, \ldots, N\}$ of the N candidate patches in the dataset $\mathcal{Y}$. Let $\boldsymbol{l} = \{l_i | i \in \mathcal{X}\}$. We specify our model in terms of an energy function:

$$f(\boldsymbol{l}) = \sum_{i \in \mathcal{X}} \varphi_i^r(l_i) + \sum_{i \in \mathcal{X}} \varphi_i^s(l_i) + \sum_{\mathbf{x_i^r} \cap \mathbf{x_j^r} \neq \emptyset} \psi(l_i, l_j), \tag{3.15}$$

with unary potentials $\varphi(\cdot)$ and pairwise potentials $\psi(\cdot)$, which are applied on overlapped patches.

Unary Potentials: The first unary potential $\varphi_i^r(l_i)$ encodes the likelihood of the similarity between the edge patch in $\mathbf{x}^r$ and the candidate edge patches in $\mathbf{y}^r$ in terms of the Euclidean difference of their corresponding distance transforms:

$$\varphi_i^r(l_i) = w^r \| d(\mathbf{x}_i^r) - d(\mathbf{y}_{l_i}^r)) \|^2, \tag{3.16}$$

where $d(\cdot)$ stands for the distance transform [36] of the edge patch. Distance transform is used to compare binary feature maps, which are not fully aligned. Thus, the introduction of distance transform results in a better similarity measurement of the binary patterns.

The second unary potential $\varphi_i^s(l_i)$ measures the similarity between the smoothed low resolution edge patch and high resolution edge patch in the dataset. In addition to $\varphi_i^r(l_i)$, the purpose of this term is to ensure that the high resolution edge patch candidates should have consistent similarity measurement both in terms of the corresponded original and the smoothed edge pattern:

$$\varphi_i^s(l_i) = w^s \| \mathbf{x}_i^s - \mathbf{y}_{l_i}^s \|^2. \tag{3.17}$$

Pairwise Potentials: The smoothness term $\psi(l_i, l_j)$ enforces coherence in the overlapping regions between the neighboring edge patch candidates, where $O_{ij}(\cdot)$ is an overlap operator that extracts the region of overlap between the distance transform of patch $\mathbf{y}^s_{l_i}$ and $\mathbf{y}^s_{l_j}$:

$$\psi(l_i, l_j) = w^p \| O_{ij}(d(\mathbf{y}^s_{l_i})) - O_{ij}(d(\mathbf{y}^s_{l_j})) \|^2. \tag{3.18}$$

i In equations defined above, w^r, w^s and w^p are nonnegative parameters chosen by experiments. (For more details, please refer to "Parameter Setting" in Section 3.5)

Optimization: For each low resolution patch $\mathcal{X}_i = \{\mathbf{x}^r_i, \mathbf{x}^s_i\}$, we first find its closest N candidate patches in $\mathcal{Y} = \{\mathbf{y}^r_i, \mathbf{y}^s_i\}$ using k nearest neighbor searching method, namely, with k-d tree to reduce searching complexity. We use the Euclidean distance defined in Eqn. 3.19 searching for the nearest neighbors:

$$dist(i, j) = \| d(\mathbf{x}^r_i) - d(\mathbf{y}^r_j) \|. \tag{3.19}$$

We obtain a minimizer of the corresponding Gibbs energy in Eqn. 3.15 to get the optimal patch in the N candidate patches using TRW-S [72]. As a result, for each edge patch $\mathcal{X}_i$, its discrete label l_i which corresponds to a high resolution edge patch in $\mathcal{Y}$ can be inferred. Finally, $\mathbf{y}^s_{l_i}$ are put together by averaging pixel values in the overlapped region to form a "likelihood" edge map (normalized to the range of $[0, 1]$). To obtain the binary edge map $\mathbf{E}^h$, we first perform non-maximum suppression on the likelihood edge map and then threshold the result to a binary edge map as show in Figure 3.6d. From the figure, we can see that in our result, the edges are smoothed (straight in the zoomed-in region) without jaggy or wavy artifacts.

3.4.2 Exploring Patch Self-similarity

Our framework can be extended to the case where an external training dataset is not available, by finding similar patches across different scales.

Given the input low-resolution depth image $\mathbf{S}^l$, we first further downsample $\mathbf{S}^l$ with a scaling factor of $1/g$ to obtain $\mathbf{S}^t$. After applying bicubic interpolation to upsample $\mathbf{S}^l$ and $\mathbf{S}^t$ to the target

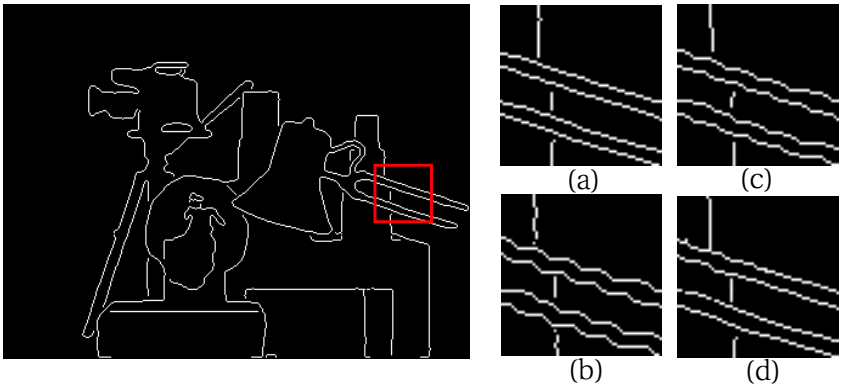


Figure 3.6: **Constructed edge map with an upscale factor of 4.** Zoomed in results of (a) The ground truth edges. (b) Edges of the bicubic upsampled depth. (c) Edges of the bicubic upsampled depth after using Shock filter. (d) Edges of our result.

resolution, we extract edges from the upsampled depth maps to obtain edge maps, denoted as $\mathbf{E}^r$ and $\mathbf{E}^t$, respectively. We directly obtain the w by w (jagged and not smooth) edge patches $\mathbf{z}_i^r$ from $\mathbf{E}^t$ and the the smooth edge patches $\mathbf{z}_i^s$ from $\mathbf{E}^r$. We also add the rotation variations to $\mathbf{z}_i^r$ and $\mathbf{z}_i^s$ by sampling every 45° out of 180° (since most of the edge patches are symmetric). The self-similarity patch pairs are denoted as $Z = \{\mathbf{z}_i^r, \mathbf{z}_i^s\}$, which contains a collection of stacked patches extracted from the same image across different scales. With $\mathcal{X}$ and $\mathcal{Z}$, we follow the same MRF framework to infer $\mathbf{E}^h$ by replacing $\mathcal{Y}$ with $\mathcal{Z}$ from Eqn. 3.15 to Eqn. 3.18. It should be noted that instead of constructing an internal patch collection $\mathcal{Z}$, one can also use PatchMatch [7] to alternatively get the N candidates for inference, as the self-similarity of the patches can be fully explored according to the image structure. The patch match cost is defined the same as in Eqn. 3.19. An illustration of the self-similarity patch match framework is shown in Figure 3.7.

Unlike previous super-resolution methods such as [54, 123], which are also based on self-similarity, we find the self-similar edge patches instead of texture patches. Our motivation of searching for self-similarity for edges is twofold: 1) Edges have higher recurrence in the image across different scales. 2) Compared with color images, depth images contain fewer unique texture

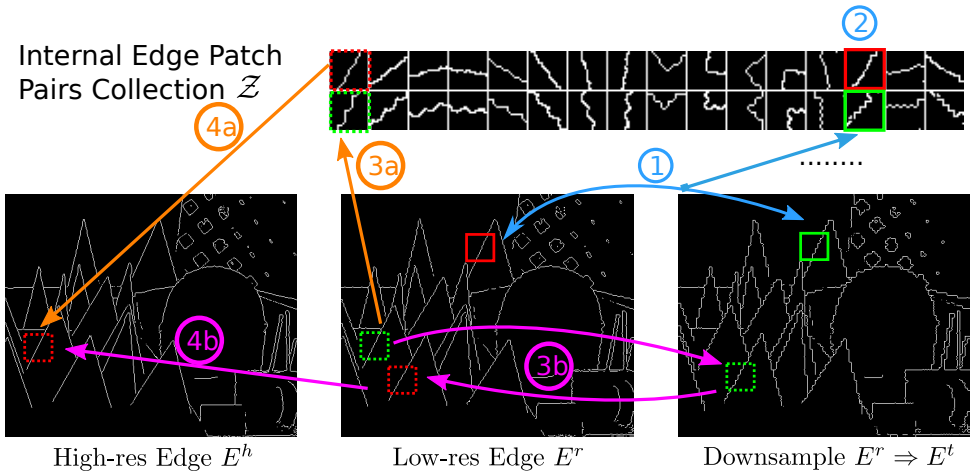


Figure 3.7: **Overview of the Self-similarity Patch Match Framework.** (1) We first extract low-res and high-res edge patch pairs from E^r and E^t, a downsampled edge map from E^r. (2) After traversing through the the entire image, we can obtain an internal edge patch pair collection $\mathcal{Z}$. (3) 3a: For each edge patch in E^r, we search for the matching edge patch pair candidates from the internal collection. 3b: Alternatively, the corresponding edge patch pair candidates can be also obtained directly from PatchMatch. Then the inference of the best match edge patch pairs is carried out via the MRF framework. (4) The high-res edge patch is obtained from the associated high-res patch (4a: from the internal patch collection, 4b: from the corresponding patch in the current low-res image E^r).

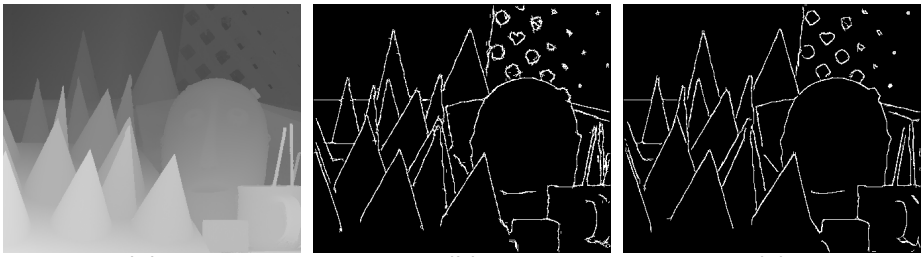


Figure 3.8: **Comparison of predicted edges with self-similarity and with external dataset by a scaling factor of 3.** (a) High resolution depth map. (b) Edges constructed based on self-similarity. (c) Edges constructed based on an external dataset.

patterns, meaning texture pattern redundancy is much less than that in natural images. A comparison of the constructed high-resolution edge map between using self-similarity and using an external dataset is shown in Figure 3.8. From it, we can see that the edges predicted from self-similarity are a bit noiser than those from an external training dataset. Although exploring self-similarity has the merit of learning local features from the current image, in our scenario, however, we are learning the depth edges, which are different from image texture patterns. More data in the external training set give larger potential to find edge patch correspondences. It also explains why the approach with external datasets has better performance than that with self-similarity.

3.4.3 Edge Assisted Depth Interpolation

Once the high-resolution edge map $\mathbf{E}^h$ is constructed either externally or internally, the high-resolution depth image $\mathbf{S}^h$ can be interpolated via a modified joint bilateral filter. For each pixel p in the target high-resolution depth image, we have:

$$\mathbf{S}^h(p) = \frac{1}{k_p} \sum_{p \subseteq N(p)} \mathbf{S}^l(q_\downarrow) \cdot f_s(\|p_\downarrow - q_\downarrow\|) \cdot f_r(\mathbf{E}^h, p, q), \tag{3.20}$$

where $N(p)$ is an s by s supporting window centered at pixel p. $p_\downarrow$ and $q_\downarrow$ denote the corresponding pixel location in the low resolution depth image. Note that $p_\downarrow$ and $q_\downarrow$ take only integer coordinates

in the low resolution image. $f_s(\cdot)$ is a zero mean spatial Gaussian kernel with a standard deviation σ_d. k_p is a normalizing factor.

The range kernel $f_r(\cdot)$ is a binary indicator defined as

$$f_r(\mathbf{E}, u, v) = \begin{cases} 1, & \text{if } u \text{ and } v \text{ are at the same side of } \mathbf{E}, \\ 0, & \text{otherwise.} \end{cases} . \tag{3.21}$$

During the interpolation process, the Gaussian kernel also smooths the depth values. Meanwhile, with the guidance provided by the high-resolution edge map, only pixels at the same side of the edge will be considered during averaging so that edges can be well preserved.

3.4.4 Graph-Based Edge Separation

To determine whether two pixels p and q are at the same side of the edge, we first dilate the patch to its 4-connected neighbors. The dilation result is denoted as $\mathbf{T}$. Therefore, if pixel i is on or next to the edge, $\mathbf{T}(p_i) = M$, where M is a constant, otherwise, $\mathbf{T}(p_i) = 0$.

Then we construct a graph $\mathcal{G}$ so that each node $\mathcal{N}_i$ corresponds to a pixel p_i in $\mathbf{T}$ and the edge $\mathcal{E}$ is formed by connecting the 8-connected neighbors of each node $\mathcal{N}_i$. The edge weighting $w(i, j)$ between neighboring pixel p_i and q_j is determined as:

$$w(i, j) = \|p_i - p_j\| \cdot (max(\mathbf{T}(p_i), \mathbf{T}(p_j)) + 1), \forall (i, j) \in \mathcal{E}. \tag{3.22}$$

Based on the assigned edge weights, we then compute a path $\mathcal{S}$ with shortest geodesic distance between p and q in $\mathcal{G}$. Our idea is to add more weights on pixels near the edge when we compute the shortest geodesic path so that the path will not touch the edge as much as possible. We draw $\mathcal{S}$ in the patch by including the 4-connected neighbors around each pixel along the path (except p and q) and we denote this "dilated" path as $\mathcal{S}'$. An example of the graph is shown in Figure 3.9a.

We consider two cases for discussion: 1) p is not an edge pixel, 2) p is an edge pixel.

CASE I: p is *not* an edge pixel. If $\mathcal{S}'$ covers the edge pixels (cyan pixels), p and q can be classified as at two sides of the edge, otherwise, p and q are at the same side. It should be noted

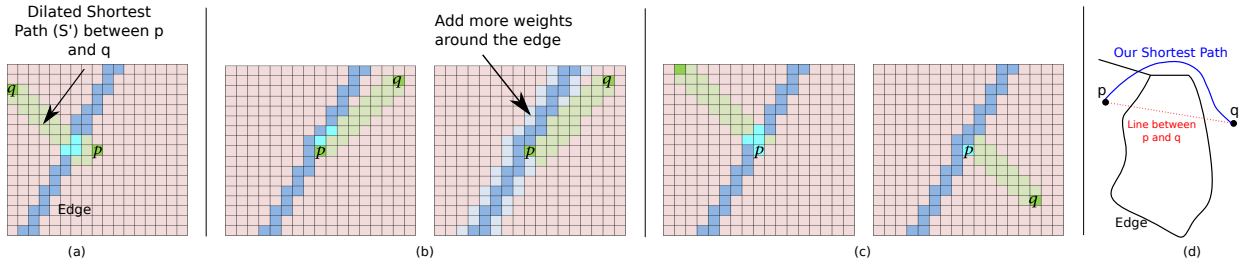


Figure 3.9: (a) Illustration of two pixels at different sides of the edge. (b) Left: A special case of two pixels mistakenly classifies as at the different sides of the edge because the dilated path intersects with the edge. Right: Adding more weights near the edge avoids situation in the left since the shortest geodesic path will choose the path with lower weights. (c) Illustration of the case that p is on the edge. Left: p and q are at different sides. Right: p and q are at the same side. (d) Note that simply connecting p and q and checking the number of intersections between the edge and the line segment from p to q does not work in this case. (Best viewed in color.)

that in some special cases, since we also add 4-connected neighbors along $\mathcal{S}$, p and q could be mistakenly classified as at opposite sides because $\mathcal{S}'$ covers the edge pixel as shown in Figure 3.9b (left). However, because we add more weights on pixels near the edge, the calculated shortest geodesic path will avoid going through pixels near the edge. As a result, $\mathcal{S}'$ will not cover the edge as shown in Figure 3.9b (right).

CASE II: p is on the edge. If p is on the edge, it is ambiguous to decide whether p and q are on the same side or not. In case that the edge is multiple pixel-wise, we first examine the number of edge pixels along $\mathcal{S}'$ in order to reduce the error caused by the thicker edge case. If the number of edge pixels along $\mathcal{S}'$ is larger than 1, p and q are classified as at different sides. Otherwise, p and q are at the same side as shown in Figure 3.9c.

It should be noted that by simply checking if an intersection between the edge and the line segment between p and q exists, we cannot determine whether p and q are on different sides, as in some situations shown in Figure 3.9d, in which p and q are at the same side but mistakenly classified as at different sides.

Once we determine whether two pixels in the support are at the same side of the edge, the high-resolution image can be interpolated using Eqn. 3.20. In case that there is no pixel on the same side of p (i.e., $g(\mathbf{E}^h, p, q) = 0, \forall q \in N(p)$), which rarely happens, we simply use the corresponded pixel value from bicubic interpolation as the upsampled value for p.

3.5 Experimental Results

In this section, we evaluate our proposed method (CDLLC, EG with self-similarity and with external dataset) both quantitatively and qualitatively with respect to other state-of-the-art super-resolution methods. We perform experiments on depth images obtained from multiple sources such as a TOF camera (PMD Camcube Camera), a laser scanner, and the Middlebury Stereo dataset.

We use the synthesized depth data mentioned in [88] as the training dataset (for both our proposed method and any baseline methods which require training data). It should be noted that the proposed Edge Guided method does not constrain to use the depth image as the training dataset. We can generally extract high-resolution and low-resolution edge patches from any natural images. However, since some of the exemplar-based or learning-based baseline methods need the depth images as the training data, for fair comparison, we also use the same data for training.

Parameters Setting: We set the same parameters in all of our experiments for different scales and different images via trial-and-error:

CDLLC: We train a coupled dictionary with size of 1000 for different scales. We select the patch size as $[m, n] = [3g, 3g]$. We set the parameters in all of the experiments as $a = 0.4$, $\theta = \pi/1000$, $\alpha = 0.01$, $w = 1$, $\lambda = 0.5$, $\varphi = 0.001$ and $K = 0.5$.

EG: The size of edge patches are $w = 21$. $w^s = 5$, $w^p = 10$ in Eqn. 3.15. We set the constant $M = 3$ for computing the shortest geodesic path. The size of the bilateral filter supporting window is $s = 7$ and $\sigma_d = 0.5$. The number of candidate patches is $N = 5$.

Baseline Methods: We compare our results with the following three categories of methods:. *1) State-of-the-art single depth image super-resolution methods:* PB [88] and Self-similarity-based approach [54]. *2) State-of-the-art general single color image super-resolution approaches*, including self-similarity-based method SRF [55], CNN-based method SRCNN [31], and sparse repre-

sentation approaches such as K-SVD [158], SCDL [139] and ScSR [153]. *3) Color-assisted depth image super-resolution approaches:* NLM [103] and TGVL2 [37]. We either use the source code provided by the authors or implement those methods by ourselves. We also select the parameters of baseline methods by experiments. More specifically, we use a few sets of parameter settings for each method (at least one of them is from the default parameters provided by the authors and the others are chosen by ourselves) and we choose the one that generates the lowest percentage of errors for all the testing images.

3.5.1 Quantitative Results

In the quantitative comparison section, for the CDLLC method, we also analyze the influence of coupled Dictionary Learning with LCC (DLLCC), Joint Reconstruction and L0 smoothing (JRL0), and Adaptive Shock Filtering (ASF) individually in the algorithm.

We first test our algorithms for the laser scanner data [88] with a scaling factor of 4. Since the method of CDLLC deals with denoising, we also compare our results of CDLLC with the effect of different edge preserving filters in pre-processing (bilateral filter in our experiment), as well as with the effect of L0 smoothing in post-processing (Post L0) in order to individually analyze the influence of Shock filtering and joint reconstruction and smoothing. Comparison of Root Mean Square Error (RMSE) results for different methods are listed in Table 3.1.

From the result, we can see the CDLLC algorithm outperforms other approaches. Also, applying Shock filter provides a better RMSE performance compared to results without Shock filter or with bilateral filter. Moreover, merely using L0 smoothing for post-processing does not help the super-resolution result. This is because smoothing the image as a post-processing step will remove some high frequency components in the resultant high-resolution image, thus yielding unwanted artifacts such as blur or over-smoothing in regions with gradual intensity changes. Using L0 smoothing as a constraint during the reconstruction stage gives better performance since the joint smoothing well constrains the latent image estimation, which alleviates the over-fitting problem and makes the reconstruction more robust to noise. Besides, the edge-guided method achieves comparable results to the CDLLC method.

Table 3.1: RMSE Comparison on the Laser Scanner Data with Different Methods and Edge Preserving Filters by a factor of 4.

	scene 021	scene 030	scene 042
Nearest Neighbor	0.0215	0.016	0.0400
Sparse coding [153]	0.0290	0.0350	0.0540
SCDL [139]	0.0258	0.0245	0.0552
K-SVD based [158]	0.0168	0.0158	0.0380
Mac Aodha et al. [88]	0.0200	0.0170	0.0400
Tsai et al. [133]	0.0165	0.0159	0.0379
Hornáček. et al. [54]	0.0210	0.0180	**0.0300**
DLLCC	0.0156	0.0157	0.0369
DLLCC+Biltaeral	0.0153	0.0157	0.0377
ASF+DLLCC	0.0152	0.0150	0.0350
ASF+DLLCC+Post L0	0.0168	0.0159	0.0361
CDLLC (ASF + DLLCC + JRL0)	**0.0150**	**0.0144**	0.0350
Edge-Guided (EG)	0.0159	0.0153	0.0368

We also evaluate our results on 15 depth images from the Middlebury Stereo dataset [114, 115] with scaling factors of 2 and 4, respectively. Note that all the depth images in the Middlebury dataset are converted to disparity range. To create low-resolution images, we smooth and down-sample the ground truth disparities beforehand. As the evaluation metric, we leverage the Root Mean Square Error (RMSE) and percent error score (PE), which is calculated as percentage of pixels for which the absolute difference in disparity exceeds 1. Table 3.2 and 3.3 show the comparison of our proposed methods in terms of different scaling factors with respect to the baseline methods. The marker means that the method does not need any external training dataset nor any other additional information except the input low-resolution depth image. Note that methods in the first two rows of the tables are color-assisted depth image super-resolution methods, in which an additional registered high-resolution color image is also provided. Numbers in bold indicate the best performance and those with an underline indicate the second best performance.

In Table 3.2, the numerical comparison is carried out with a relatively small scaling factor 2. The data from D1 to D15 are corresponding to Cones, Teddy, Tsukuba, Venus, Adirondack, ArtL, Jadeplant, Motorcycle, Piano, Pipes, Playroom, Playtable, Recycle, Shelves and Vintage in the dataset, respectively. Our proposed Edge Guided methods generate much smaller errors in terms of the Percent of Error score compared with other methods and are ranked top compared with other single depth super-resolution methods as well. Moreover, in the proposed approach EG, the one with an external dataset generates better results compared with the one with self-similarity. In Table 3.3, the results are based on a larger scaling factor 4. Our proposed methods (EG) again are ranked the best in terms of the Percent of Error score. It is also interesting that with a larger scaling factor, our performance difference between the one with an external dataset and the one with self-similarity becomes smaller. We conclude that it is because with a larger scaling factor, it is more difficult to find the correct edge patch pairs externally.

In terms of the comparison between our methods CDLLC and EG, we can see that EG achieves slightly better results for the accuracy evaluation using RMSE when the scaling factor is smaller but worse RMSE results when the scaling factor is larger when compared to CDLLC. This is because CDLLC also deals with denoising, which smooths the depth image. However, regarding

Table 3.2: RMSE and Percent of Error Comparison on the Middlebury Data with Different Methods by a Scaling Factor of 2.

RMSE x 2	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
NLM [103]	1.03	0.78	0.61	0.23	1.68	3.05	7.55	2.70	0.89	3.35	2.43	1.49	1.11	1.05	1.07
TGVL2 [37]	**0.73**	**0.57**	0.53	**0.17**	**1.10**	2.52	6.74	2.32	**0.67**	2.78	2.03	**1.12**	**0.73**	**0.86**	0.98
ScSR [153]	1.15	0.90	0.64	0.29	1.93	3.43	8.42	2.90	1.32	3.48	2.85	1.82	1.34	2.33	2.22
K-SVD [158]	0.91	0.70	0.51	0.23	1.52	2.69	6.62	2.27	1.07	2.83	2.28	1.42	1.08	1.95	1.87
SRCNN [31]	1.12	0.88	0.64	0.28	1.93	3.38	8.50	2.97	1.64	3.75	3.09	1.87	1.45	3.18	2.86
SRF [55]*	1.15	0.90	0.66	0.29	1.95	3.43	8.52	2.96	1.57	3.57	3.12	1.89	1.44	2.93	2.89
PB [88]	1.18	0.89	0.62	0.30	1.89	3.38	8.42	2.89	1.34	3.46	2.82	1.78	1.30	2.40	2.20
CDLLC	0.85	0.67	0.48	0.21	1.42	2.54	6.57	2.18	0.95	2.73	2.03	1.29	0.95	1.82	1.62
Ours (*Self.*)*	0.81	0.64	0.47	0.18	1.31	2.42	6.36	2.07	0.74	2.66	1.86	1.21	0.87	1.27	**0.92**
Ours	0.76	0.63	**0.45**	0.19	1.26	**2.40**	**6.29**	**2.03**	0.74	**2.65**	**1.82**	1.22	0.86	1.31	1.09
PE x 2	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
NLM [103]	3.26	3.12	2.35	0.44	2.17	4.61	12.81	4.99	2.41	6.80	4.45	3.93	1.73	2.31	2.20
TGVL2 [37]	2.54	2.31	1.79	0.41	1.60	4.18	10.51	4.46	1.64	5.59	3.92	2.96	1.15	1.87	2.21
ScSR [153]	4.43	3.76	3.27	0.71	4.25	10.21	23.69	8.06	3.23	12.03	7.76	5.70	2.65	5.75	2.88
K-SVD [158]	3.97	2.97	2.48	0.59	3.51	8.21	18.44	6.44	2.90	10.67	6.56	4.54	2.31	5.02	2.24
SRCNN [31]	4.99	3.98	2.99	0.71	4.65	11.54	24.45	8.99	4.38	14.61	9.52	6.58	2.94	7.80	3.23
SRF [55]*	4.52	3.88	3.53	0.67	4.13	9.82	22.87	8.17	3.61	12.09	7.91	5.89	2.75	6.93	2.94
PB [88]	4.35	4.13	1.57	0.39	2.31	5.41	19.45	7.29	3.05	9.13	8.47	7.26	2.40	3.43	6.20
CDLLC	3.68	2.99	2.41	0.71	3.00	6.55	15.06	5.52	2.59	9.52	5.66	4.27	1.92	4.19	1.98
Ours (*Self.*)*	1.96	1.71	1.40	0.38	1.28	2.18	5.73	2.67	1.33	4.11	2.55	2.31	0.97	1.47	0.86
Ours	**1.72**	**1.61**	**1.27**	**0.37**	**1.11**	**2.01**	**5.64**	**2.49**	**1.17**	**3.76**	**2.30**	**2.08**	**0.89**	**1.45**	**0.81**

Table 3.3: RMSE and Percent of Error Comparison on the Middlebury Data with Different Methods by a Scaling Factor of 4.

RMSE x 4	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
NLM [103]	1.52	1.08	0.77	0.29	2.08	4.02	10.16	3.65	1.21	4.75	3.05	2.14	1.42	1.37	1.51
TGVL2 [37]	1.13	**0.83**	0.71	**0.24**	**1.52**	3.48	9.01	3.07	**0.93**	3.90	2.64	**1.49**	**1.04**	**1.15**	**1.42**
ScSR [153]	1.45	1.18	0.82	0.38	2.38	4.48	11.09	3.74	1.80	4.85	3.67	2.33	1.68	3.34	3.46
K-SVD [158]	1.15	0.92	0.66	0.30	1.88	3.54	8.73	2.92	1.40	3.94	2.86	1.81	1.35	2.70	2.81
SRCNN [31]	1.41	1.10	0.79	0.34	2.30	4.42	11.28	3.75	2.07	5.08	3.86	2.30	1.70	4.12	3.94
SRF [55]*	1.48	1.23	0.87	0.39	2.46	4.93	11.94	3.99	2.42	5.33	4.34	2.61	1.90	4.74	4.76
PB [88]	1.56	1.26	0.86	0.38	2.54	4.88	11.73	3.81	1.74	5.33	3.85	2.41	1.87	3.18	2.99
CDLLC	**1.07**	0.85	**0.61**	0.27	1.76	**3.31**	**8.32**	**2.77**	1.25	**3.79**	**2.60**	1.65	1.25	2.48	2.57
Ours (*Self.*)*	1.24	0.97	0.71	0.29	1.94	3.65	9.60	3.11	1.21	4.30	2.82	1.85	1.35	2.26	2.07
Ours	1.16	0.95	0.67	0.29	1.94	3.67	9.50	3.05	1.15	4.28	2.80	1.82	1.35	2.16	2.29
PE x 4	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
NLM [103]	7.18	6.27	4.44	0.90	5.46	11.37	27.40	10.31	5.29	15.11	9.91	8.71	4.17	5.10	6.95
TGVL2 [37]	4.34	3.72	3.08	0.60	2.29	6.66	16.96	7.23	2.85	9.11	6.60	4.76	2.01	3.52	5.12
ScSR [153]	9.33	7.79	6.15	1.43	9.64	21.45	38.26	15.82	7.61	27.88	17.00	11.95	5.86	13.56	6.28
K-SVD [158]	6.45	5.17	4.30	1.22	5.55	13.65	28.56	10.33	4.87	18.05	10.57	7.63	3.67	8.49	3.68
SRCNN [31]	8.64	6.92	5.52	1.30	7.92	18.37	33.83	13.85	7.66	23.59	15.40	10.94	4.96	14.27	5.40
SRF [55]*	8.44	7.37	6.20	1.45	7.82	17.03	33.44	15.00	7.85	23.10	15.37	11.32	5.56	15.25	6.05
PB [88]	9.73	8.03	2.52	0.66	6.07	10.95	31.21	14.61	9.17	20.28	17.66	16.41	5.78	9.59	13.80
CDLLC	5.79	4.72	4.15	1.18	4.68	11.12	25.35	8.79	4.34	16.06	9.06	6.67	3.12	7.25	3.22
Ours (*Self.*)*	3.23	3.13	2.37	0.59	**1.79**	**3.52**	**10.06**	**4.66**	2.38	**6.70**	**4.71**	**3.88**	1.96	3.08	2.14
Ours	**3.09**	**3.11**	**2.36**	**0.54**	1.92	3.92	11.47	4.98	**2.21**	6.98	4.80	3.99	**1.92**	**2.96**	**2.00**

Table 3.4: Average Processing Time for CDLLC (sec.)

	Shock Filtering and Initialization	Iterative Reconstruction		Total Time
		Coeff. Estimation	L0 Smoothing	
125×150 Image	3.77	17.1	12.6	35.1

Table 3.5: Average Processing Time for EG (sec.)

	Patch NN search	MRF-inference	MRF-inference	Total Time
125×150	29.89	10.78	20.02	60.69

the percent of error pixels, which is a more common assessment over depth or disparity, the edge-guided method gives much better performance. Though popular in image restoration literature, the RMSE metric over disparity is dominated by wrong assignments around boundaries. Thus, a disparity map with blurry boundaries might generate a better RMSE score than a disparity map with a few pixels being assigned to a wrong foreground/background disparity value around the boundary. Therefore, to evaluate the accuracy of disparity estimation, the Percent of Error metric has been reported as a more fair measurement in stereo-related fields like stereo matching [44].

In terms of Percent of Error metric, the proposed Edge Guided (EG) method with an external training dataset performs best for all the testing images and is even better than the color-assisted approaches. It is worth mentioning that in EG, we obtain about a 29% error drop on average in terms of the percent error score compared to the best performer of the baseline methods for each scale and each testing image. The proposed Edge-Guided approach with self-similarity also achieves comparable results in terms of PE.

It is also interesting to see that the state-of-the-art single color image super-resolution methods such as SRF [55] and SRCNN [31] do not perform well in the depth image domain. For example,

our proposed methods based on self-similar edge patch match performs much better than that of SRF [55], which adopts transformed self-exemplars as well. The experimental results also indicate that the color image super-resolution techniques cannot be simply transferred to the depth domain as depth image super-resolution offers unique challenges compared to color image super-resolution.

Complexity. Table 3.4 and 3.5 list the processing time of our proposed algorithms for upscaling an image to about 450×375 with a scaling factor of 3. The simulations are carried out on MATLAB using a machine with a dual-core 3.1 GHz Intel i3-2100 CPU, 16.0 GB RAM. In the tables, we can see that, on average, the processing time of our algorithms is reasonable and the CDLLC method is significantly faster than the EG method.

3.5.2 Qualitative Results

We evaluate our proposed methods visually from Figure 3.10 to Figure 3.15. Among the figures, Figure 3.10 to 3.12 show the super-resolution results of the Middlebury data with zoomed cropped regions. Figure 3.13 shows the result of upscaling the depth from a laser scanner. From the figures, we can see that our proposed methods also generate more visually appealing results than the previously reported approaches. Boundaries in our results are generally sharper and smoother along the edge direction. Our proposed method also well preserves the structure of the scene in regions with fine details.

Moreover, for EG, we generally obtain visually similar results with the proposed method based on an external dataset and based on self-similarity. However, we can still notice subtle differences in some results such as in Figure 3.12 (top), where the one with self-similarity has some artifacts around the branch of the plant.

We also compare the view synthesis results using the upscaled depth images by different single image or depth super-resolution methods as shown in Figure 3.14. We first upsample the depth images from two different views using different methods. With the upscaled depth results, we synthesize a new color view in between using the view synthesis algorithm in [59] and measure the resultant Peak Signal Noise Ratio (PSNR) value for the synthesized results generated by depth from

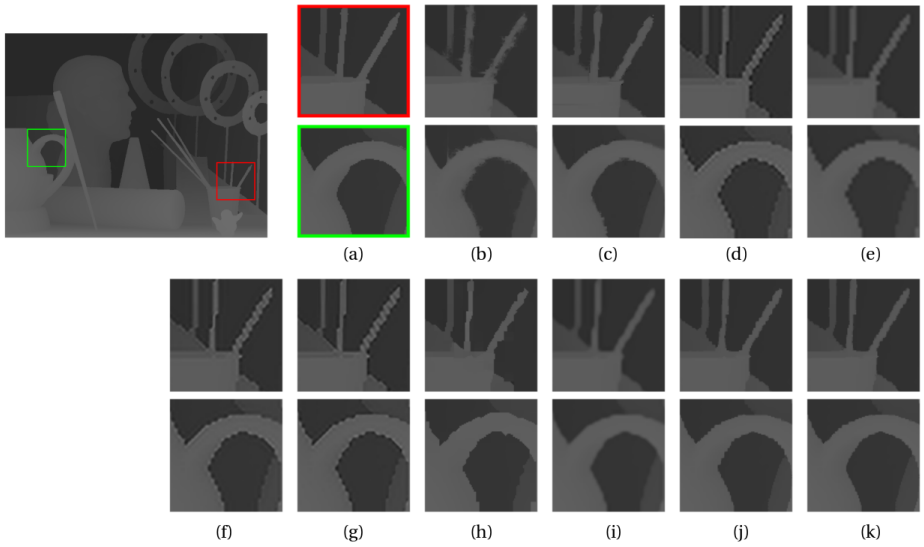


Figure 3.10: **Visual comparison of ArtL with cropped zoomed regions** (g = 3). (a) Ground truth. (b) NLM [103]. (c) TGVL2 [37]. (d) ScSR [153]. (e) K-SVD [158]. (f) SRCNN [31]. (g) SRF [55]. (h) PB [88]. (i) CDLLC. (j) EG without training data. (k) EG.

each method. We also synthesize the "Ground Truth" color image of the view using the ground truth high-resolution depth and color images provided in the dataset as a comparison. From it, we can see that the edge-guided method obtains the highest PSNR scores and better view synthesis results compared to other methods. The CDLLC is also one of the best performers compared with other methods.

To further demonstrate the effectiveness of our proposed approach, we show the 3D mesh constructed from the upscaled depth with different methods in Figure 3.15, in which the depth is captured by a PMD camera (TOF camera). From the figure, we can see that the edge-guided method yields sharp 3D boundaries as well as relatively smooth surfaces. It should be noted that in the CDLLC, the results suffer over smoothness while the result of method PB [88] has some severe block artifacts.

As a comparison between the two methods that we propose, in general, the EG method performs slightly better than the CDLLC method for depth super-resolution in terms of percentage

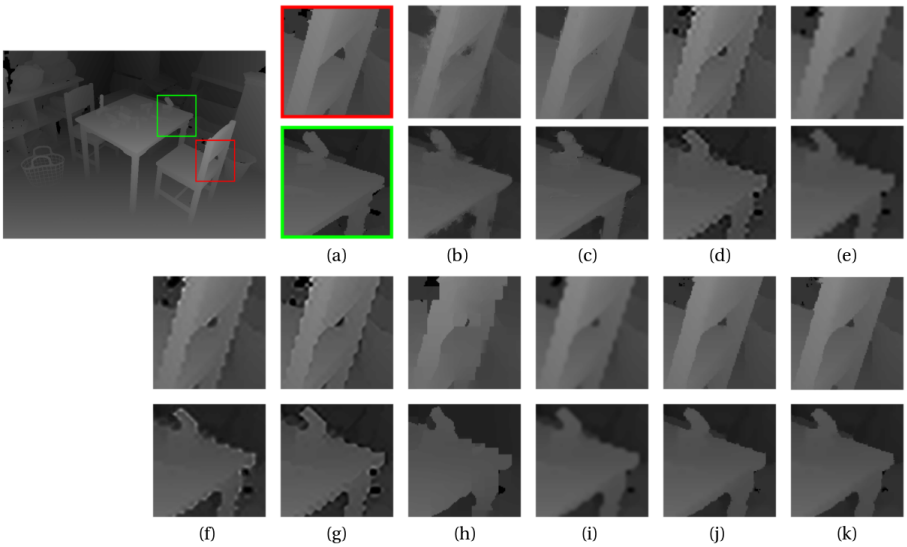


Figure 3.11: **Visual comparison of Playtable with cropped zoomed regions** ($g = 4$). (a) Ground truth. (b) NLM [103]. (c) TGVL2 [37]. (d) ScSR [153]. (e) K-SVD [158]. (f) SRCNN [31]. (g) SRF [55]. (h) PB [88]. (i) CDLLC. (j) EG without training data. (k) EG.

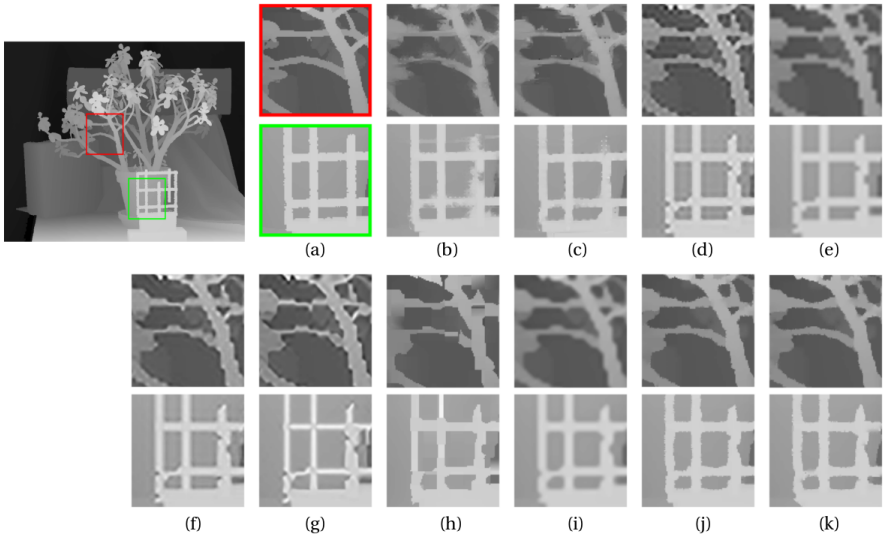


Figure 3.12: **Visual comparison of Jadeplant with cropped zoomed regions** ($g = 4$). (a) Ground truth. (b) NLM [103]. (c) TGVL2 [37]. (d) ScSR [153]. (e) K-SVD [158]. (f) SRCNN [31]. (g) SRF [55]. (h) PB [88]. (i) CDLLC. (j) EG without training data. (k) EG.

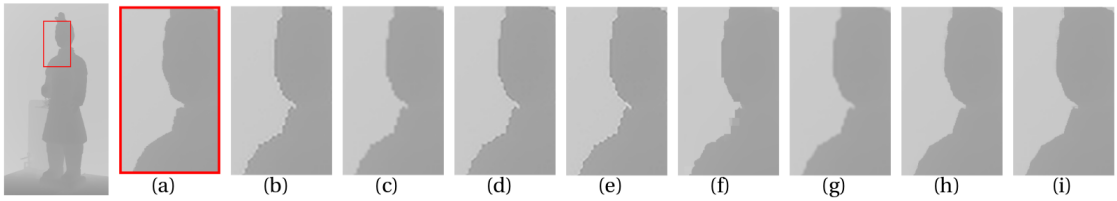


Figure 3.13: **Visual comparison of Laser Data with cropped zoomed regions** ($g = 4$). (a) Ground truth. (b) ScSR [153]. (c) K-SVD [158]. (d) SRCNN [31]. (e) SRF [55]. (f) PB [88]. (g) CDLLC. (h) EG without training data. (i) EG.

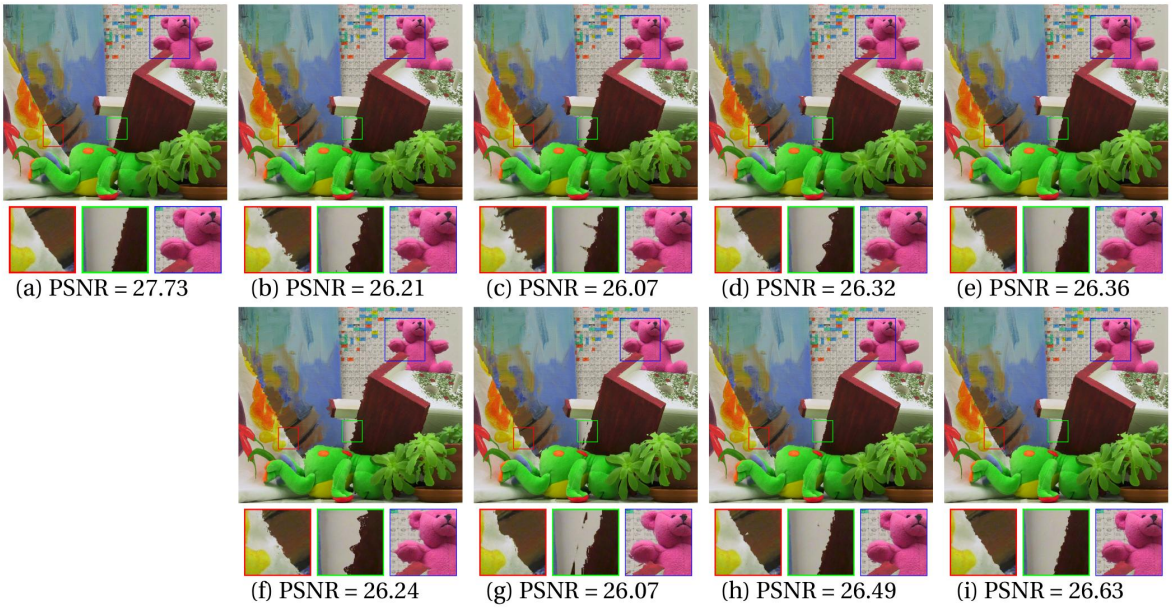


Figure 3.14: **Visual comparison of view synthesis result on depth images scaled by a factor of 4 with cropped zoomed regions.** (a) Ground truth. (b) ScSR [153]. (c) K-SVD [158]. (d) SRCNN [31]. (e) CDLLC. (f) SRF [55]. (g) PB [88]. (h) EG without training data. (i) EG.

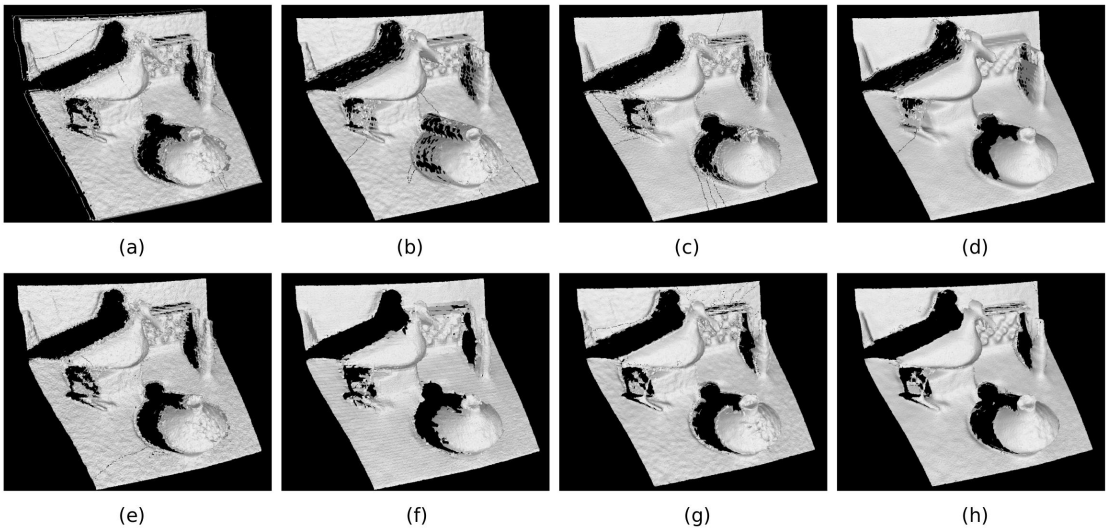


Figure 3.15: **Visual comparison of the 3D mesh from depth images scaled by a factor of 4.** (a) Ground truth. (b) ScSR [153]. (c) K-SVD [158]. (d) CDLLC. (e) SRF [55]. (f) PB [88]. (g) EG without training data. (h) EG.

error and visual result. This is because in depth images, which are usually relatively textureless, edges play a very important role. A sharper edge can lead to better depth visualization as well as better performance on depth related application such as view synthesis and 3D reconstruction. However, regarding the noise, CDLLC has a better performance in denoising and also gives better results in terms of RMSE evaluation, since the CDLLC algorithm also considers removing the present noise in depth at the same time as upscaling the depth image. Also, the CDLLC method requires significantly less computation compared to the EG method.

3.6 Conclusion

In this chapter, we propose two approaches for single depth image super-resolution: a Coupled Dictionary Learning-based approach with Local Constraint (CDLLC) and an Edge-Guided approach (EG). In CDLLC, we introduce a locality constraint in the coupled dictionary learning process to train a more robust dictionary. We also jointly reconstruct and smooth the high-resolution image

using an L0 gradient smooth constraint. Furthermore, we use an adaptively regularized Shock filter to tackle the jagged edge problem without introducing blurry artifacts around the depth discontinuities. In EG, we present a novel framework for single depth image super-resolution guided by a constructed high-resolution edge map. Motivated by the idea that edges are of particular importance in the textureless depth image, we convert the super-resolution problem from high-resolution texture prediction to high-resolution edge prediction. We construct the high-resolution edge map by casting it as a MRF labeling problem. Moreover, we also propose to incorporate self-similarity edge patch match during the edge prediction process, when an external training dataset is not available. Then guided by the edge map, we propose to interpolate the high-resolution depth image using a modified joint bilateral filter. From experimental results, our methods not only have better objective performance (i.e., for EG, it reduces 29% error on average compared with state-of-the-art methods in terms of Percent of Error score metric), but also help avoid artifacts introduced by direct texture prediction, reduce jagged artifacts, and preserve sharp edges.

Chapter 4

SEMANTIC INSTANCE ANNOTATION OF STREET SCENES BY 3D TO 2D LABEL TRANSFER

4.1 Introduction

The revolutionary success of high-capacity deep learning architectures [75, 86, 163] may flag the beginning of a paradigm shift in computer vision. Rather than developing methods for solving a certain task, future research could be directed towards teaching a "universal program" (e.g., a deep network) a mapping from input to output space. One fundamental question arising in this context is how the required ground truth labels for training these models can be generated at very large scales (i.e., > 100k images). While for some tasks large annotated datasets are already available today (e.g., image classification [110]), other tasks such as semantic segmentation of street scenes lack this information as human annotation is very labor-intensive. We refer to this phenomenon as the "curse of dataset annotation" (Figure 4.1).

One option to circumvent this problem is to exploit auxiliary tasks for which large annotated datasets are available. While generalization of the target domain can be achieved to some extent, discriminative cues which solve the auxiliary problem will dominate the learned representation [162]. A second option is the creation of synthetic datasets. Unfortunately, our community still lacks rich generative image formation models which are able to produce realistic and diverse imagery from the true underlying distribution of the 3D world we live in. In this chapter, we therefore propose an alternative approach which leverages additional 3D information to simplify the 2D annotation task.

Recently, applications such as autonomous cars and humanoid robots have attracted significant attention. For research in these applications, a street view video dataset with dense semantic labels will be very useful. Motivated by those needs, our work focuses on the challenging task of

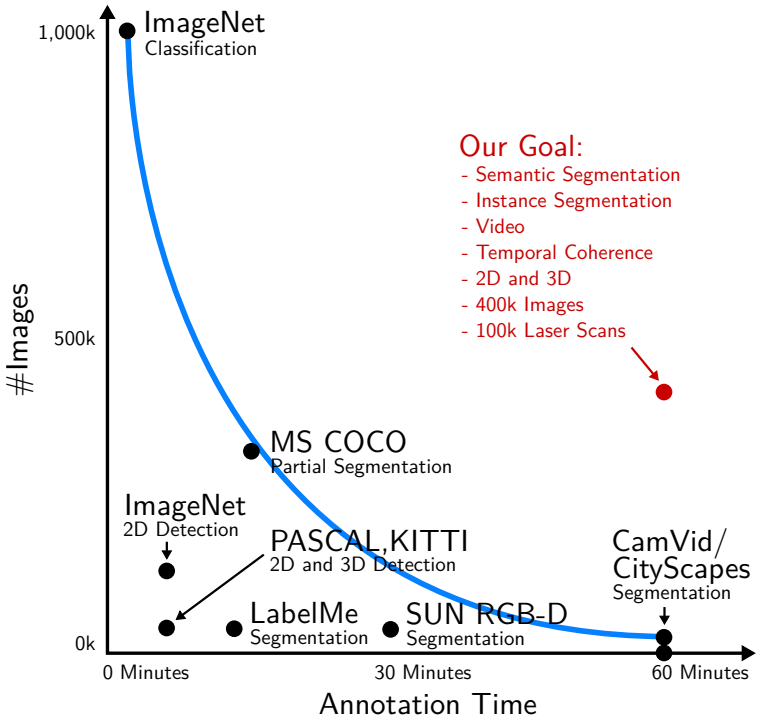


Figure 4.1: Curse of dataset annotation.

semantic and instance video annotation of street scenes for which pixelwise labeling requires up to minutes per image for a human annotator as acknowledged in [6]. Inspired by the easy usage of 3D modeling tools (Blender, SketchUp) we propose to annotate scenes directly in 3D and then transfer this knowledge back into the image domain. The required 3D information can be obtained from various sources including structure-from-motion (SfM), stereo, or laser scanners. This approach has several advantages over labeling in 2D: First, objects often project into several images of the video sequence, thus lowering annotation efforts considerably. Furthermore, the obtained 2D instance annotations are temporally coherent as they are associated with a single object in 3D. And finally, our 3D annotations might be useful by themselves for reasoning in 3D [159] or for enriching 2D annotations with approximate depth or coarse 3D geometry.

Unfortunately, obtaining dense and accurate 2D labels from sparse noisy point clouds and coarse 3D annotations is a challenging task in itself. Towards solving this problem, in this chapter, we propose a non-local multi-field CRF model which reasons jointly about semantic and instance

labels of all 3D points and all pixels in the image, as illustrated in Figure 4.2. This approach offers several advantages over methods which reason purely in 2D [5, 137]: Occluders and occludees which exhibit complex boundaries when projected onto the image plane (e.g. a tree in front of a building) are often easier to separate in 3D. Besides, our approach is not affected by missing labels due to occlusions or drift in optical flow. Furthermore, our model allows us to specify a tractable semantic instance loss for principled and efficient end-to-end parameter learning. And finally, the probabilistic nature of our model allows for estimating label uncertainties, which can be used to increase label accuracy when only a subset of the pixels require a label. In summary, we make the following two contributions in this chapter [148].

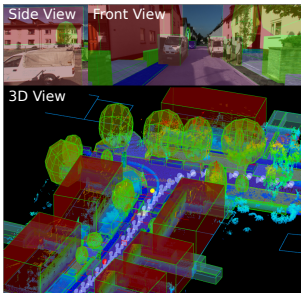
- We present a novel geo-registered dataset of suburban scenes recorded by a moving platform. The dataset comprises over 400k images and over 100k laser scans, and we provide semantic 3D annotations for all static scene elements.

- We propose a method which is able to transfer these labels from 3D into 2D, yielding pixel-wise semantic instance annotations. We demonstrate the potential of our approach in ablation studies and with respect to several 2D and 3D baselines.

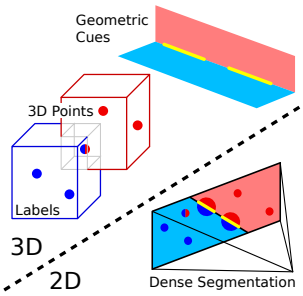
4.2 Related Work

In this section, we first review semi-supervised video annotation methods, followed by an overview of existing semantic and instance segmentation datasets.

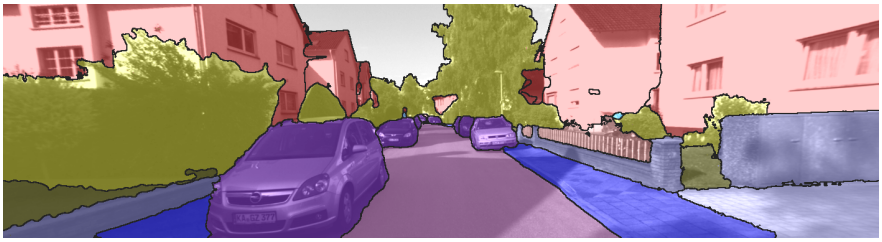
Methods: Compared to annotating individual images [49, 83, 150], using video sequences offers the advantage of temporal coherence between adjacent frames. Label propagation techniques exploit this fact by transferring labels from a sparse set of annotated key frames to all unlabeled frames based on color and motion information. While in some works a single foreground object is assumed [60, 134], here we focus on methods that can handle multiple object categories. Towards this goal, [6, 16] proposed a coupled Bayesian network based on video epitomes and semantic regions to propagate label information between two annotated key frames. To better account



(a) 3D annotation



(b) Label Transfer



(c) 2D annotation

Figure 4.2: **3D to 2D label transfer:** (a) We annotate all objects in 3D using bounding primitives. (b) Our model then transfers this information into 2D by jointly reasoning about 3D geometric cues, sparse 3D points, as well as image pixels. (c) This allows us to infer temporally consistent semantic instance annotations for every frame in the video.

for errors in label propagation, [96] proposed a hierarchy of local classifiers for this task and [5] leveraged a mixture-of-tree model for temporal association. The problem of selecting the most promising key frames for annotation has been considered in [137].

In contrast to the aforementioned methods which propagate labels in 2D, in this chapter we propose to annotate directly in 3D and then project these annotations into the 2D domain. While this approach requires a source of 3D information (e.g., SfM, stereo, laser), it is able to produce more accurate semantic and temporally consistent instance annotations. Furthermore, our experiments indicate that annotation in 3D is more time efficient than labeling in 2D as scene elements can be separated more easily and often project into many images of the input video sequence.

There exists little work on 3D to 2D label transfer. A notable exception is the approach of Chen

et al. [20], in which annotations from KITTI [43] as well as 3D car models are leveraged to infer separate figure-ground segmentations for all vehicles in the image. In comparison, our approach reasons jointly about all objects in the scene and also handles categories for which CAD models or 3D point measurements are unavailable (e.g., "Tree", "Sky"). In the context of street view image segmentation, Xiao et al. [142] present a hybrid method where annotated 3D points from structure-from-motion are projected onto superpixels in the image and users interactively correct wrong predictions with 2D scribbles. However, as no occlusion reasoning is performed, their method can only be applied to scenes with little variations in depth (e.g., facades). Other methods [14,90,94,95,97] which model the interaction between image pixels and 3D points focus primarily on improving classification performance or efficiency by exploiting multiple input modalities while our goal is to transfer ambiguous 3D primitive labels to every pixel in the image.

Datasets: While some datasets such as PASCAL VOC [33] or MS COCO [82] provide semantic labels for a subset of pixels in the image, here we focus on datasets with dense semantic annotations. Most of these datasets provide only a small number (about 1k) of accurately annotated indoor [118] or outdoor [47, 117] images. A notable exception is LabelMe [111] with more than 10k images labeled using crowdsourcing techniques. Compared to the smaller datasets, however, not all images are densely annotated, quality varies heavily amongst annotators, and polygons have been chosen over pixels for more efficient but less accurate representation.

A number of works have also considered the annotation of video sequences [15, 119, 141]. In [141], eight RGB-D sequences of indoor scenes have been manually annotated using an interactive tool which propagates 2D polygons from one frame to another. The recently proposed SUN RGB-D dataset [119] provides labeled 2D polygons as well as 3D cuboids for 10k RGB-D images captured indoors. For street scenes, less annotated data is available [8, 94, 95, 108, 136]. While KITTI [44] provides semantic information only for a few object categories , CamVid [15] offers pixel-accurate labels, but without instances and for a very limited number of frames. Very recently, the Cityscapes dataset [26] has been proposed with 5k manually annotated individual 2D images of

street scenes[1]. Our dataset differs from Cityscapes in that we provide temporally coherent semantic instance annotations at a much larger scale, as well as omnidirectional imagery, 3D laser scans, and 3D annotations which might also be directly useful for reasoning in 3D. While [26] focuses on inner-city scenes, our dataset comprises mainly suburban areas; thus both datasets complement each other.

4.3 Method

In this work, we are interested in generating semantic instance annotations for urban scenes at a large scale by transferring labels from sparse 3D point clouds into the images. In particular, we focus on static scene elements which dominate suburban scenes. Dynamic objects could be handled via 3D models [20, 92], but as our dataset comprises few dynamic objects we leave this extension for future work. This section describes our data collection efforts, our 3D annotation process, as well as the proposed label transfer model.

4.3.1 Data Collection

For our data collection, we equipped a station wagon with one 90° fisheye camera to each side and a 90° perspective stereo camera (baseline 60 cm) to the front. Furthermore, we mounted a Velodyne HDL-64E and a SICK LMS 200 laser scanning unit in pushbroom configuration on top of the roof. This setup is similar to the one used in KITTI [43,44], except that we gained a full 360° field of view due to the additional fisheye cameras and the pushbroom laser scanner, in comparison to KITTI, which only provides perspective images and Velodyne laser scans with a 26.8° vertical field of view. Approximate localization is provided by an integrated IMU/GPS measurement unit.

Using this setup, we recorded several suburbs of a mid-size city corresponding to over 400k images and 100k laser scans. We estimated all vehicle and camera poses using structure-from-motion [52]. More specifically, we minimized 3D reprojection errors based on all feature matches while regularizing against the GPS solution. This results in accurate georegistered camera poses.

[1] `http://www.cityscapes-dataset.net/`

While our label transfer approach does not assume geolocalization, geospatial information[2] can facilitate the 3D annotation task.

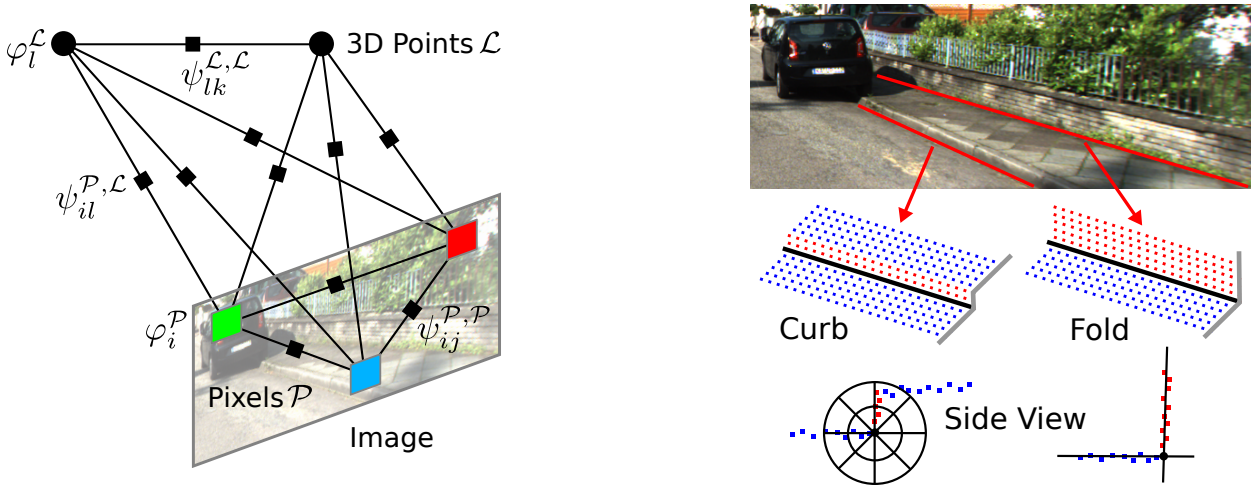
4.3.2 Annotation

We augmented our dataset with 3D annotations in the form of bounding primitives; i.e., we placed cuboids and ellipsoids around objects in 3D and assigned a semantic label to each of them. More specifically, we asked a group of annotators to tightly enclose the 3D points belonging to an object by the respective primitive. For this purpose, we developed a 3D annotation tool based on WebGL (see Figure 4.2a) which visualizes the 3D points and the two camera views, and provides tools to facilitate navigation and annotation. To enable efficient annotation, our primitives are rough approximations of the true object shapes and thus are allowed to overlap in 3D (see Figure 4.2b). For stuff categories (e.g., "Road", "Sidewalk", "Grass") we allowed users to draw 2D polygons in bird's eye view which are then extruded into 3D to better approximate the shape and to facilitate annotation. Ambiguities are resolved using our label transfer method described in the following section. Annotating a single batch comprising 200 laser scans and 800 images required about 3 hours.

4.3.3 Model

Given sparse point clouds and 3D annotations, we are interested in generating dense semantic instance annotations for all images. Towards this goal, we propose a CRF model which reasons jointly about the labels of the 3D points and all pixels in the image, leveraging the calibration and registration described in Section 4.3.1. Note that our 3D annotations are sparse and noisy; i.e., 3D points can carry none, one, or multiple labels due to overlapping bounding primitives in 3D. The algorithm described in this section is designed to resolve these situations and infers marginal estimates for all 3D points and pixels in the image. In order to make our approach more robust in regions where appearance is not discriminative, we investigate additional geometric cues of the

[2] `http://www.openstreetmap.org/`



(a) Graphical model

(b) 3D curbs and folds

Figure 4.3: **Label transfer model.** (a) Factor graph representation of our graphical model for 3D to 2D label transfer. Our approach estimates marginal distributions for all pixels $\mathcal{P}$ and 3D points $\mathcal{L}$. (b) We localize 3D geometric structures such as folds and curbs to improve segmentation boundaries between the categories "Road", "Sidewalk" and "Wall".

3D point cloud such as 3D surface folds and curbs (see Figure 4.3b). If detected, these cues can provide accurate boundaries between semantic classes in the image.

More formally, let $\mathcal{P}$, $\mathcal{L}$ and $\mathcal{F}$ denote the set of image pixels, sparse 3D points from laser/stereo, and detected 3D fold or curb segments, respectively. For each pixel $i \in \mathcal{P}$ and each 3D point $l \in \mathcal{L}$, we specify random variables s_i and s_l taking values from the set of semantic (or instance) labels $\{1, \ldots, S\}$, where S denotes the number of classes. For instance inference, we assign a unique ID to each object which projects into the image. Thus, semantic and instance inference can be treated equally under our model and we will refer to both as "semantic labels" in the following.

Let $\mathbf{s} = \{s_i | i \in \mathcal{P}\} \cup \{s_l | l \in \mathcal{L}\}$ denote the set of semantic labels. Dropping all dependencies on the image and point cloud for clarity we specify our CRF in terms of the following Gibbs energy

function:

$$E(\mathbf{s}) = \sum_{i\in\mathcal{P}} \varphi_i^{\mathcal{P}}(s_i) + \sum_{l\in\mathcal{L}} \varphi_l^{\mathcal{L}}(s_l) + \sum_{m\in\mathcal{F}} \sum_{i\in\mathcal{P}} \varphi_{mi}^{\mathcal{F}}(s_i) \tag{4.1}$$
$$+ \sum_{i,j\in\mathcal{P}} \psi_{ij}^{\mathcal{P},\mathcal{P}}(s_i, s_j) + \sum_{l,k\in\mathcal{L}} \psi_{lk}^{\mathcal{L},\mathcal{L}}(s_l, s_k) + \sum_{i\in\mathcal{P},l\in\mathcal{L}} \psi_{il}^{\mathcal{P},\mathcal{L}}(s_i, s_l)$$

with unary potentials $\varphi(\cdot)$ and pairwise potentials $\psi(\cdot)$. For notational clarity, we omit all conditional dependencies on the input images, 3D points and 3D annotations.

Pixel Unary Potentials: The pixel unary potentials $\varphi_i^{\mathcal{P}}(s_i)$ encode the likelihood of pixel i taking label s_i

$$\varphi_i^{\mathcal{P}}(s_i) = w_1^{\mathcal{P}}(s_i)\, \xi_i^{\mathcal{P}}(s_i) - \, w_2^{\mathcal{P}}(s_i) \log p_i^{\mathcal{P}}(s_i) \tag{4.2}$$

where $w_1^{\mathcal{P}}$ and $w_2^{\mathcal{P}}$ denote learned feature weights. Our first constraint $\xi_i^{\mathcal{P}}(s_i)$ determines the set of admissible labels and is obtained by projecting the 3D bounding primitives (which are an upper bound on the objects' extent) into the image. We formulate the constraint via a binary feature $\xi_i^{\mathcal{P}}(s_i) \in \{0, 1\}$ which takes 0 for pixel i if its ray passes through a primitive of class s_i, and 1 otherwise.

In addition, we leverage appearance information by projecting all non-occluded sparse 3D points into all adjacent frames of the image sequence and training a pixel-wise classifier [117] based on these projections. This results in a per-pixel probability distribution over semantic labels $p_i^{\mathcal{P}}(s_i)$. The intuition behind this feature is that regions of the same semantic class are similar in adjacent frames and thus yield highly discriminative cues for the current frame.

3D Point Unary Potentials: The 3D point unary potentials $\varphi_l^{\mathcal{L}}(s_l)$ encode the likelihood of 3D point l taking label s_l:

$$\varphi_l^{\mathcal{L}}(s_l) = -w^{\mathcal{L}}(s_l)\, \xi_l^{\mathcal{L}}(s_l) \tag{4.3}$$

where $\xi_l^{\mathcal{L}}(s_l)$ denotes a feature which takes 0 if the 3D point l lies within a 3D primitive of class s_l, and 1 otherwise. As the "sky" class cannot be modeled with primitives we set $\xi_l^{\mathcal{L}}(s_l)$ to 0 if s_l takes the label "sky". Additionally, we create "virtual sky points" at infinity for all pixels whose ray doesn't intersect any 3D primitive. Note that these pixels must correspond to sky regions as we assume that each object is completely contained in one or several bounding 3D primitive(s).

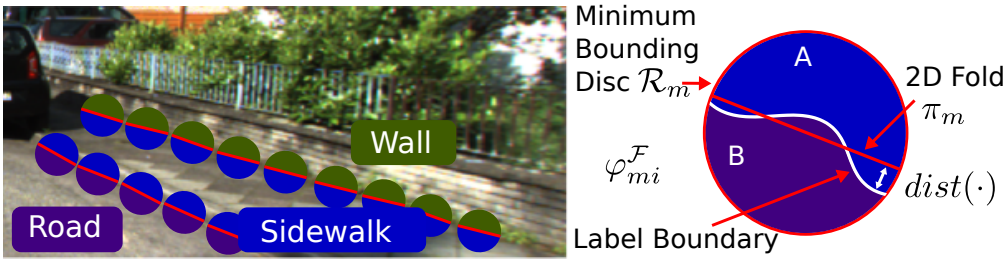


Figure 4.4: **Geometric unary potentials.** Left: We encourage label changes at 3D curbs or folds after projection into the image domain. Right: This constraint ($\varphi^{\mathcal{F}}_{mi}$) is implemented by pixel unary potentials inside each minimum bounding disc $\mathcal{R}_m$ around each 2D curb or fold segment m.

Geometric Unary Potentials: We encourage label changes at curbs or folds which we detect in 3D using plane fitting, as described in Appendix A. Given the projections into 2D, we introduce the following constraint:

$$\varphi^{\mathcal{F}}_{mi}(s_i) = w^{\mathcal{F}} \frac{[\mathbf{p}_i \in \mathcal{R}_m \wedge \nu_m(\mathbf{p}_i) \neq s_i]}{\exp\{dist(\mathbf{p}_i, \boldsymbol{\pi}_m)\}} \tag{4.4}$$

Here, $[\cdot]$ is the Iverson bracket, $\mathbf{p}_i$ denotes the 2D location of pixel i and $\mathcal{R}_m$ represents a 2D disc around curb or fold segment m projected into 2D (yielding a line segment $\boldsymbol{\pi}_m$) as illustrated in Figure 4.4. ν_k is a function which takes as input a pixel location and returns the semantic label predicted by fold m. More specifically, we project the 3D fold into 2D and compute the majority label at its two sides from the sparse projected 3D points. The denominator in Eqn. 4.4 ensures a penalty decay towards the disc boundaries.

Pixel Pairwise Potentials: Our dense pairwise term encourages semantic label coherence and connects all pixels in the image via Gaussian edge kernels

$$\begin{aligned}\psi^{\mathcal{P},\mathcal{P}}_{ij}(s_i, s_j) \;=\; & w^{\mathcal{P},\mathcal{P}}_1(s_i, s_j) \exp\left\{-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\,\theta^{\mathcal{P},\mathcal{P}}_1}\right\} \\ + & w^{\mathcal{P},\mathcal{P}}_2(s_i, s_j) \exp\left\{-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|^2}{2\,\theta^{\mathcal{P},\mathcal{P}}_2} - \frac{\|\mathbf{c}_i - \mathbf{c}_j\|^2}{2\,\theta^{\mathcal{P},\mathcal{P}}_3}\right\}\end{aligned} \tag{4.5}$$

where $\mathbf{p}_i$ is the 2D location of pixel i and $\mathbf{c}_i$ denotes its color value. Further, $w^{\mathcal{P},\mathcal{P}}_1$ and $w^{\mathcal{P},\mathcal{P}}_2$ are learned pairwise feature weights and $\theta^{\mathcal{P},\mathcal{P}}$ parametrizes the kernel width.

3D Pairwise Potentials: Similarly, we apply a Gaussian edge kernel to encourage label consistency between 3D points based on their 3D location and surface normals

$$\begin{aligned} \psi_{lk}^{\mathcal{L},\mathcal{L}}(s_l, s_k) &= w^{\mathcal{L},\mathcal{L}}(s_l, s_k) \\ &\times \exp\left\{ -\frac{\|\mathbf{p}_l^{3d} - \mathbf{p}_k^{3d}\|^2}{2\,\theta_1^{\mathcal{L},\mathcal{L}}} - \frac{(n_l - n_k)^2}{2\,\theta_2^{\mathcal{L},\mathcal{L}}} \right\} \end{aligned} \tag{4.6}$$

where $\mathbf{p}_l^{3d}$ is the 3D location of point l and n_l denotes the vertical (up) component of its normal. We use the normal's z-component as it is the most discriminative cue for indicating label changes between horizontal (e.g., road, sidewalk) and vertical (e.g., side of car, wall) surfaces. We estimate the respective normals using principle component analysis in a local neighborhood around each 3D point.

2D/3D Pairwise Potentials: Finally, we encourage coherence between all 3D points and the image pixels

$$\psi_{il}^{\mathcal{P},\mathcal{L}}(s_i, s_l) = w^{\mathcal{P},\mathcal{L}}(s_i, s_l) \exp\left\{ -\frac{\|\mathbf{p}_i - \boldsymbol{\pi}_l\|^2}{2\,\theta^{\mathcal{P},\mathcal{L}}} \right\} \tag{4.7}$$

where $\boldsymbol{\pi}_l$ denotes the projection of the 3D laser or stereo point l onto the image plane. Importantly, we project only points into the image which are likely to be visible. We determine these points by meshing the 3D point cloud using the ball-pivoting method of Bernardini et al. [10], considering only 3D points in front of the mesh. We also tried state-of-the-art multi-view reconstruction approaches [62] for mesh generation, but obtained better results with the described meshing approach.

4.4 Learning and Inference

This section describes inference and parameter estimation in our label transfer model.

Inference: At test time, we are interested in estimating the marginal distribution of each semantic or instance label in $\mathbf{s}$ under our model, specified by the Gibbs distribution defined in Eqn. 4.1. The most likely configuration can then be estimated by variable-wise maximization of these marginals. As our graphical model is loopy, exact inference in polynomial time is intractable. We resort to variational inference and approximate the probability distribution on $\mathbf{s}$ by replacing it

with a factorized mean field distribution $Q(\mathbf{s}) = \prod_{i \in \mathcal{P} \cup \mathcal{L}} Q_i(s_i)$. This mean field approximation can be computed efficiently using bilateral filtering [74]. As our model comprises three sets of densely connected variables (namely $\mathcal{P}$, $\mathcal{L}$ and $\mathcal{P} \leftrightarrow \mathcal{L}$), we exploit the algorithm of [67, 138] which generalizes [74] to multiple fields.

Learning: We employ empirical risk minimization in order to learn the parameters in our model, considering the univariate logistic loss, defined as $\Delta(s) = -\log(P(s))$ where $P(\cdot)$ denotes the marginal distribution at the respective site. Let us subsume all model parameters into $\Theta = \{w_1^{\mathcal{P}}, w_2^{\mathcal{P}}, w^{\mathcal{L}}, w^{\mathcal{F}}, w_1^{\mathcal{P},\mathcal{P}}, w_2^{\mathcal{P},\mathcal{P}}, w^{\mathcal{P},\mathcal{L}}, w^{\mathcal{L},\mathcal{L}}\}$. We define our minimization objective $f(\Theta)$ as the regularized univariate logistic loss:

$$f(\Theta) \;=\; \sum_{n=1}^{N} \sum_{i \in \mathcal{P}} -\log\left(Q_{n,i}(s_{n,i}^*)\right) + \lambda\, C(\Theta) \tag{4.8}$$

Here, N is the number of training images, $s_{n,i}^*$ denotes the ground truth semantic label and $Q_{n,i}(\cdot)$ is the approximate margin at pixel i in image n, calculated via mean field approximation. $C(\Theta)$ is a quadratic regularizer on the parameter vector Θ. We whiten all features and use a single value λ which we select via cross-validation on the training set. For learning the instance segmentation parameters we exploit the same loss $f(\Theta)$ as for semantic segmentation. In order to associate 2D ground truth instances with 3D instances we project all visible 3D points into the image and find a consensus via the majority vote which gave good results in practice. As the number of instances per semantic class varies between images, we learn intra- and inter-class pairwise potentials using parameter tying.

We optimize the objective function $f(\Theta)$ using stochastic gradient descent. The derivative of $f(\Theta)$ wrt. $Q_{n,i}(s_{n,i})$ is given by

$$\frac{\partial f(\Theta)}{\partial Q_{n,i}(s_{n,i})} = \begin{cases} -Q_{n,i}(s_{n,i}^*)^{-1} & \text{if } s_{n,i} = s_{n,i}^* \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

We obtain $\partial Q / \partial \Theta$ using auto differentiation of our mean field inference algorithm. For faster convergence, we make use of the ADADELTA algorithm [157] with decay parameter 0.95 and

$\epsilon = 10^{-8}$, and randomly sample a batch of 16 training images at each iteration for which all gradients can be computed in parallel.

4.5 Experimental Evaluation

In this section, we first evaluate our method in ablation studies and with respect to several label transfer baselines. Besides, we exploit the uncertainty in our predictions to increase accuracy for semi-dense predictions. Finally, we show some qualitative results of our method.

As input to our method, we accumulate all laser measurements in a common world coordinate system and augment them with 3D points from stereo matching [53]. To reduce outliers, we consider only points up to 15 m distance, and apply left-right as well as forward-backward consistency checks over 5 frames. We fuse all 3D points into one global point cloud and remove all points which are closer than 5 cm to their nearest neighbor.

For evaluation, we manually annotated 160 images from 8 different suburbs with dense pixel-wise ground truth. From the 160 frames, 120 frames have been labeled in equidistant steps of 5 frames for comparison with 2D label transfer methods. We learn the parameters in our and the baseline models using 2-fold cross validation at the sequence level to avoid any bias caused by the correlation of adjacent frames within a sequence.

4.5.1 Quantitative Evaluation

This section presents our quantitative evaluation. We compare our method with respect to several baselines on semantic and instance segmentation tasks.

Semantic Segmentation: As semantic segmentation is a more established task with a larger number of baselines available, we focus on this task first. For evaluation, we map the 27 semantic labels in our 3D annotations to the most frequently occurring 14 categories, “Road”, “Parking”, “Sidewalk”, “Terrain”, “Building”, “Vegetation”, “Car”, “Trailer”, “Caravan”, “Gate”, “Wall”, “Fence”, “Box”, and “Sky”. The color coding is shown in Figure 4.5 and Figure 4.6. We report the frequency of these classes in Table 4.1. As a metric we leverage the common Jaccard Index, calculated as the intersection-over-union w.r.t. the estimated and the ground truth pixels. We

Figure 4.5: Color coding of semantic labels.

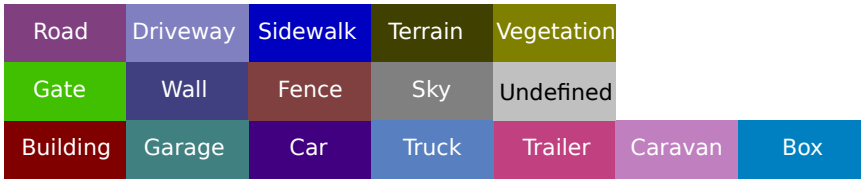


Figure 4.6: Color coding of instance labels.

measure overall performance by the average Jaccard Index (JI) weighted by the class frequency and the average pixel accuracy (Acc).

The upper half of Table 4.2 shows results of several 2D to 2D label transfer methods on all 120 equidistantly labeled frames. Here, the task is to predict the center frame from two annotated images (± 5 frames corresponding to 0.5 seconds of driving or ~ 5 meters travel distance).

Our first baseline ("Label Prop.") is the label transfer approach presented in [137]. To ensure that all baselines have access to the same information, we do not select frames in an active fashion but use equidistantly spaced labels for all methods (the driving speed during recording was nearly constant). We construct a second baseline ("Sparse Track. + GC") using the feature tracking approach of [124] to propagate semantic labels from the two closest labeled frames to the target frame. To densify the label map, we apply graph cuts (GC) with contrast sensitive edge potentials [13].

In order to evaluate the value of 3D information, we implemented a third baseline ("3D Prop.

Semantic Labels	**Instance Labels**	**Frequency (%)**	**Definition**
Road	Road	11.55	
Sidewalk	Sidewalk	6.91	
Driveway	Driveway	1.29	
Terrain	Terrain	1.40	Grass, Soil, Stone
Building	Building, Garage	29.04	
Vegetation	Vegetation	25.03	
Car	Car, Truck	9.61	
Trailer	Trailer	0.49	
Caravan	Caravan	0.49	
Box	Box	0.27	Box, Trashbin, Vendingmachine
Wall	Wall	3.63	
Fence	Fence	1.81	
Gate	Gate	0.44	
Sky	Sky	7.80	
Undefined	Undefined	0.26	Motorcycle, Bicycle, Pedestrian, Rider, BigPole, SmallPole, TrafficLight, TrafficSign, Lamp

Table 4.1: **Mapping between Instance Labels and Semantic Labels.** Frequencies are specified in percentage of pixels.

+ GC") which works similar to the previous one, but replaces the sparse tracking part with correspondences obtained by transferring pixels of the two closest labeled frames to the target image via the visible vertices of our 3D mesh followed by graph cuts propagation. While this method takes advantage of 3D information, the propagation is purely done in 2D. Finally, we train the model of Krähenbühl et al. [74] ("Fully Conn. CRF") on all annotated adjacent frames of the *test sequence* in order to segment the respective target frame.

From the 2D label transfer baselines, the mesh transfer method which uses projected 3D information performs best. Furthermore, and maybe surprisingly, the fully connected CRF model which does not use correspondence information performs on par or even better than special purpose label transfer methods. According to our experiments, this is caused by the fact that optical flow (as used in [124, 137] for propagating labels) often fails for street scenes like ours due to large displacements, perspective distortions, textureless regions and challenging lighting conditions. On the other hand, the fully connected model performs weaker for less frequent or textureless classes such as "Trailer" or "Box".

The bottom half of Table 4.2 compares the proposed method with respect to several 3D to 2D label transfer baselines which in contrast to the 2D to 2D label transfer methods exploit our 3D annotations but do not require equidistantly labeled 2D annotations. Since there exists little prior work on this task, we construct a set of baselines from our model for this purpose. As evidenced by our results, simply projecting 3D primitives or meshes into the image and smoothing via GC does not perform well due to the crude approximation of the geometry ("3D Primitives + GC", "3D Mesh + GC"). Better results are obtained when projecting the visible 3D points followed by spatial propagation ("3D Points + GC").

Finally, we observe that all baselines are outperformed by the proposed method (last row) in almost all categories. Importantly, note that the 2D methods require every 10th frame to be labeled, while our method (as well as the other 3D baselines) require 3D annotations in the form of 3D primitives. Assuming 60 minutes annotation time per image, this amounts to 20 hours of annotation time per batch of 200 frames when labeling one 2D image every 10th frame, while the respective 3D annotations for this scene can be obtained in less than 3 hours. Note that labeling

Method	Road	Park	Swlk	Terr	Bldg	Vegt	Car	Trler	Carvn	Gate	Wall	Fence	Box	Sky	JI	Acc
Label Prop. [137]	93.4	51.8	73.5	58.3	80.2	69.9	61.5	22.4	42.3	30.6	45.3	45.7	32.5	89.6	74.4	84.4
Spas Trk. + GC [124]	89.6	37.1	69.0	54.2	84.6	79.5	78.2	2.5	35.3	3.2	38.9	32.9	7.0	91.0	77.8	87.3
3D Prop. + GC	91.3	44.5	74.0	62.4	86.2	81.8	81.6	5.2	38.6	12.7	47.4	42.0	15.0	88.8	80.2	88.9
Fully Conn. CRF [74]	88.5	37.8	68.4	55.8	85.5	79.8	76.8	2.5	30.6	2.9	38.3	32.4	0.0	**92.8**	77.9	87.4
3D Primitives + GC	78.7	46.4	43.9	46.5	54.9	55.4	55.1	72.3	54.6	51.0	40.2	52.3	40.2	55.4	56.4	72.1
3D Mesh + GC	92.1	66.4	72.4	66.1	69.1	74.9	87.7	88.9	88.5	61.9	51.4	60.7	30.4	46.4	72.5	82.6
3D Points + GC	93.1	72.4	78.9	72.4	81.9	77.2	88.1	92.4	91.1	70.2	66.8	68.7	62.4	69.9	80.5	89.0
Proposed Method	**95.3**	**80.6**	**86.4**	**81.0**	**90.9**	**86.9**	**91.5**	**94.9**	**91.8**	**73.6**	**78.9**	**79.4**	**73.0**	91.0	**89.2**	**94.2**

Table 4.2: **Comparison to Label Transfer Baselines on Semantic Segmentation Task.** We compare our method to 2D label transfer baselines (top) and to 3D to 2D label transfer baselines (bottom) on 120 consecutive images. See text for details.

each frame of the sequence manually would require 200 hours. This gain multiplies with the frame rate and the number of cameras (our setup comprises four).

Ablation Study: We evaluate the importance of the individual components of our model in Table 4.3 (top). Starting with the appearance classifier trained on the projected sparse 3D points ($p^{\mathcal{P}}$), we incrementally add the terms related to the 3D points ($\varphi^{\mathcal{L}}$,$\psi^{\mathcal{P},\mathcal{L}}$), the semantic pairwise term between pixels ($\psi^{\mathcal{P},\mathcal{P}}$), the 3D primitive constraints ($\xi^{\mathcal{P}}$), the 3D pairwise constraints ($\psi^{\mathcal{L},\mathcal{L}}$) and finally the remaining terms ($\varphi^{\mathcal{F}}_{mi}$) as specified in Eqn. 4.1. We note that each component is able to increase performance. As expected, we obtain the largest improvement by reasoning about the relationship between points in 3D and pixels in the image. Integrating 3D fold and curb detections increases overall performance only slightly, but improves boundaries, in particular between road and sidewalk.

Semi-dense Inference: Often, it is not necessary to label all pixels in every image for training a semantic segmentation model. In this section, we therefore leverage our model's awareness of label uncertainty to estimate semi-dense label maps with high accuracy. To quantify uncertainty, we

Method	Road	Park	Swlk	Terr	Bldg	Vegt	Car	Trler	Carvn	Gate	Wall	Fence	Box	Sky	JI	Acc
LA	92.2	64.6	77.9	67.5	85.2	81.9	81.7	85.7	81.5	46.8	62.1	60.3	49.4	83.1	82.1	90.0
LA+3D	95.0	76.9	85.5	73.3	87.9	84.3	89.4	88.2	90.2	68.8	74.6	74.0	63.7	83.4	86.2	92.5
LA+PW	92.5	68.6	79.5	73.1	87.3	84.2	84.1	89.9	85.9	48.7	66.2	64.9	54.5	86.6	84.4	91.4
LA+PW+CO	93.0	72.7	81.2	73.8	87.7	84.5	85.7	90.9	88.4	57.7	70.4	69.6	57.6	86.9	85.2	92.0
LA+PW+CO+3D	93.2	78.6	85.0	76.3	90.6	86.7	89.1	90.9	92.7	68.5	77.8	78.9	67.8	90.7	88.2	93.7
+ 3D PW	94.9	80.1	85.9	80.0	90.6	87.0	91.2	91.3	93.8	72.6	78.1	78.5	69.3	90.8	88.8	94.0
Full Model	95.4	80.1	87.1	80.0	90.6	87.0	91.2	91.3	93.9	72.6	78.4	78.6	69.4	90.8	89.0	94.1
Full Model (90%)	98.1	92.3	94.7	92.4	95.3	93.5	96.5	95.8	97.6	83.7	90.7	90.7	84.0	94.6	94.9	97.4
Full Model (80%)	98.8	95.3	96.7	94.9	96.8	95.5	97.5	96.4	98.5	86.4	93.7	93.4	87.9	96.4	96.6	98.2
Full Model (70%)	**99.2**	**96.8**	**97.9**	**96.4**	**97.5**	**96.8**	**97.9**	**97.2**	**99.0**	**88.1**	**95.0**	**94.6**	**90.1**	**97.2**	**97.5**	**98.7**

Table 4.3: **Ablation Study on Semantic Segmentation Task.** This table shows the importance of the different components in our model on all 160 images. The components are abbreviated as follows: LA = local appearance ($p^{\mathcal{P}}$), PW = 2D pairwise constraints ($\psi^{\mathcal{P},\mathcal{P}}$), CO = 3D primitive constraints ($\xi^{\mathcal{P}}$), 3D = 3D points ($\varphi^{\mathcal{L}}$,$\psi^{\mathcal{P},\mathcal{L}}$), 3D PW = 3D pairwise constraints ($\psi^{\mathcal{L},\mathcal{L}}$), Full Model = all potentials including folds. Percentages denote fractions of estimated pixels. See text for details.

Method	Road	Park	Sdwlk	Terr	Bldg	Vegt	Car	Trler	Carvn	Gate	Wall	Fence	Box	Sky	JI	Acc
LA+3D	94.5	74.7	83.5	73.4	80.7	84.5	86.3	90.8	90.9	66.3	74.7	75.6	63.1	81.9	83.5	91.0
LA+PW+CO	92.8	70.3	79.8	73.9	64.9	84.6	82.2	90.7	87.1	51.7	67.8	66.6	24.7	88.0	78.4	87.4
LA+PW+CO+3D	94.6	78.4	84.2	78.4	86.3	87.6	90.8	93.0	93.3	70.9	77.6	79.4	68.6	91.1	87.5	93.3
+ 3D PW	95.1	**80.6**	85.3	**79.3**	86.4	**87.9**	91.5	93.0	93.6	**73.6**	78.1	79.0	70.4	90.7	87.9	93.5
Full Model	**95.7**	**80.6**	**86.9**	79.2	**86.4**	**87.9**	**91.5**	**93.1**	**93.6**	**73.6**	**78.5**	**79.1**	**70.5**	**90.7**	**88.1**	**93.6**

Table 4.4: **Ablation Study on Instance Segmentation Task** using the same abbreviations as in Table 4.3. See text for details.

measure the entropy of the label marginal distribution at every pixel. Sorting all pixels according to their entropy allows us to predict the most certain regions in the image.

Table 4.3 (bottom) and Figure 4.7 show our results when predicting only those parts of the image. Note how this helps to boost our performance to 94.9% JI and 97.4% accuracy when predicting at 90% pixel density. In contrast, uncertainty is not directly accessible in most of the baseline models as they are deterministic or rely on MAP estimates.

Instance Segmentation: As time consistent 2D instance ground truth is hard to obtain, most existing 2D label transfer methods focus on the semantic segmentation problem. Therefore, we chose to evaluate instance segmentation performance in an ablation study. We annotated the classes "Building", "Car", "Trailer", "Caravan" and "Box" with instances in our 2D ground truth. While the remaining classes (e.g., "Road", "Sky") do not admit unambiguous instance labels, we also report their performance as our model reasons about all instance and semantic classes jointly. Table 4.4 shows our results. Note how the instance segmentation results are on par with the semantic segmentations, demonstrating our model's intra-class separation ability.

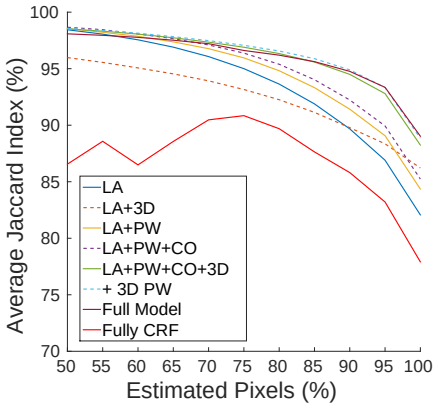
4.5.2 Qualitative Evaluation

Figure 4.8 illustrates our dense inference results qualitatively for 6 different scenes in terms of semantic instance segmentation. The last row shows the error maps where colors indicate the true label (see Figure 4.5 and Figure 4.6 for color coding). While the proposed method is able to delineate most object boundaries satisfyingly, some challenges remain. In particular, errors occur in low-contrast image regions with overlapping 3D annotations (Scene 1: car/road boundary) and in regions where 3D points are absent due to sensor occlusion (Scene 4: building roof). Another source of errors are inherent label ambiguities which occur for porous objects such as fences or trees (Scene 6: tree boundary) where even 2D ground truth annotation is a difficult and ambiguous task. Finally, manual 2D annotations also contain errors, in particular at complex boundaries which are hard to delineate (Scene 4: trees, Scene 5: hedge). However, note that our semi-dense inference is able to successfully identify those regions as shown in Figure 4.8 - Figure 4.9. Furthermore, in Figure 4.8 - Figure 4.9, we provide additional qualitative results as well as our inference results

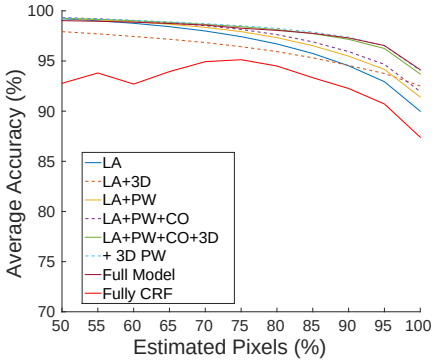
for 3D points. Figure 4.10 shows the visual comparison with baseline methods. In Figure 4.11, we also show the accumulated 3D semantic point cloud inferred from our model.

4.6 Conclusion

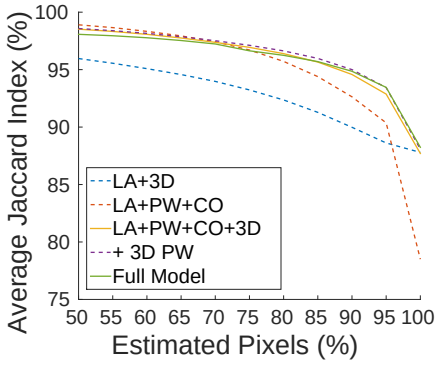
We present a method for semantic instance labeling of large datasets from annotated 3D primitives. In the presence of 3D data, our method yields better results compared to several state-of-the-art 2D label transfer baselines while lowering annotation time. Furthermore, our method results in temporally consistent instance labels, and explicitly exposes label uncertainty. We also propose a novel dataset (which we will make available), comprising 400k images, laser point clouds, and annotations for all objects.



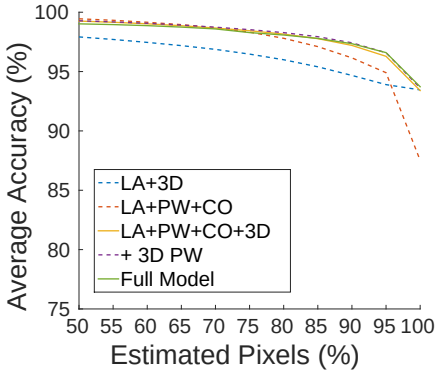
(a) Jaccard Index (Semantic)



(b) Accuracy (Semantic)

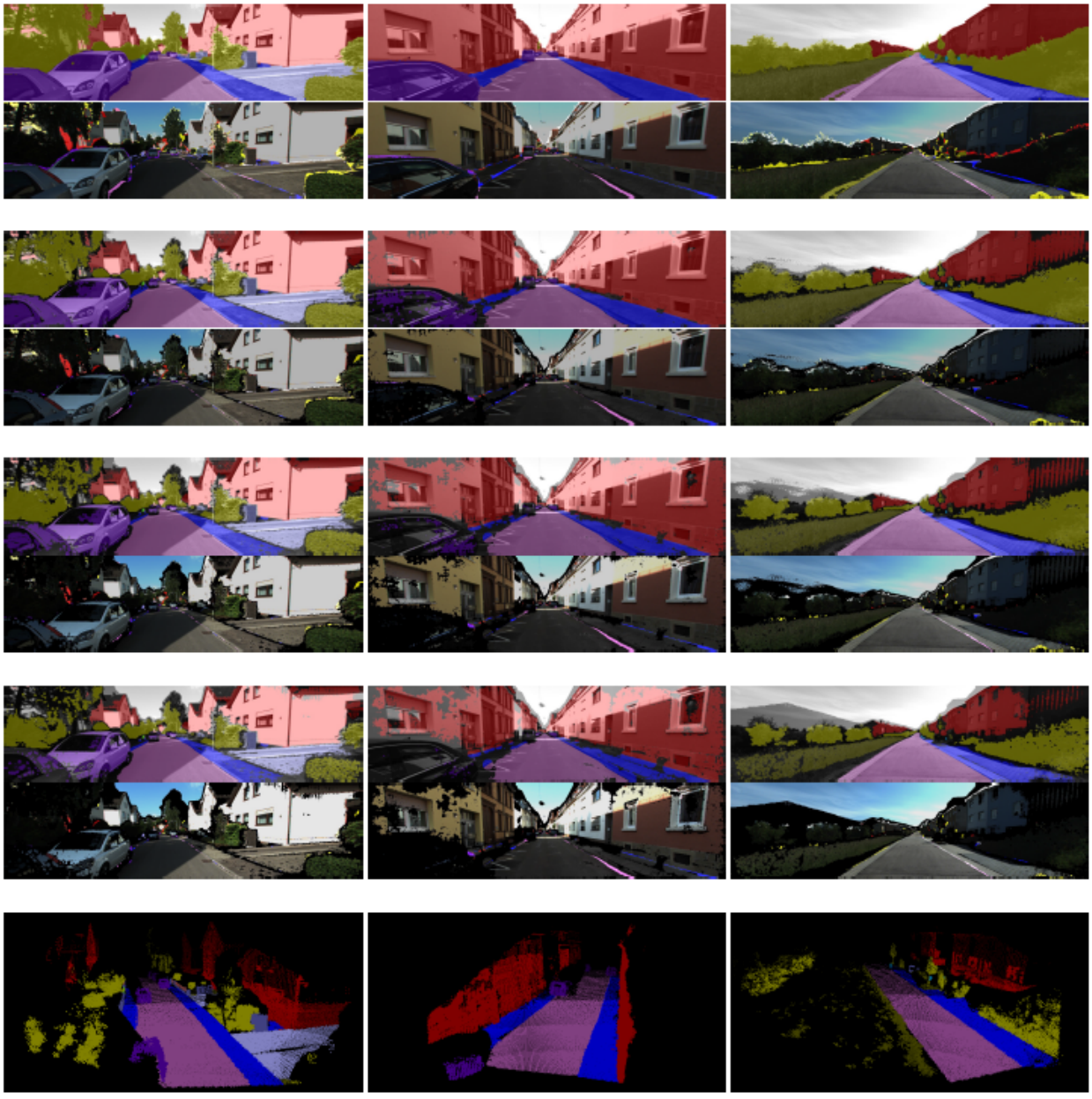


(c) Jaccard Index (Instance)



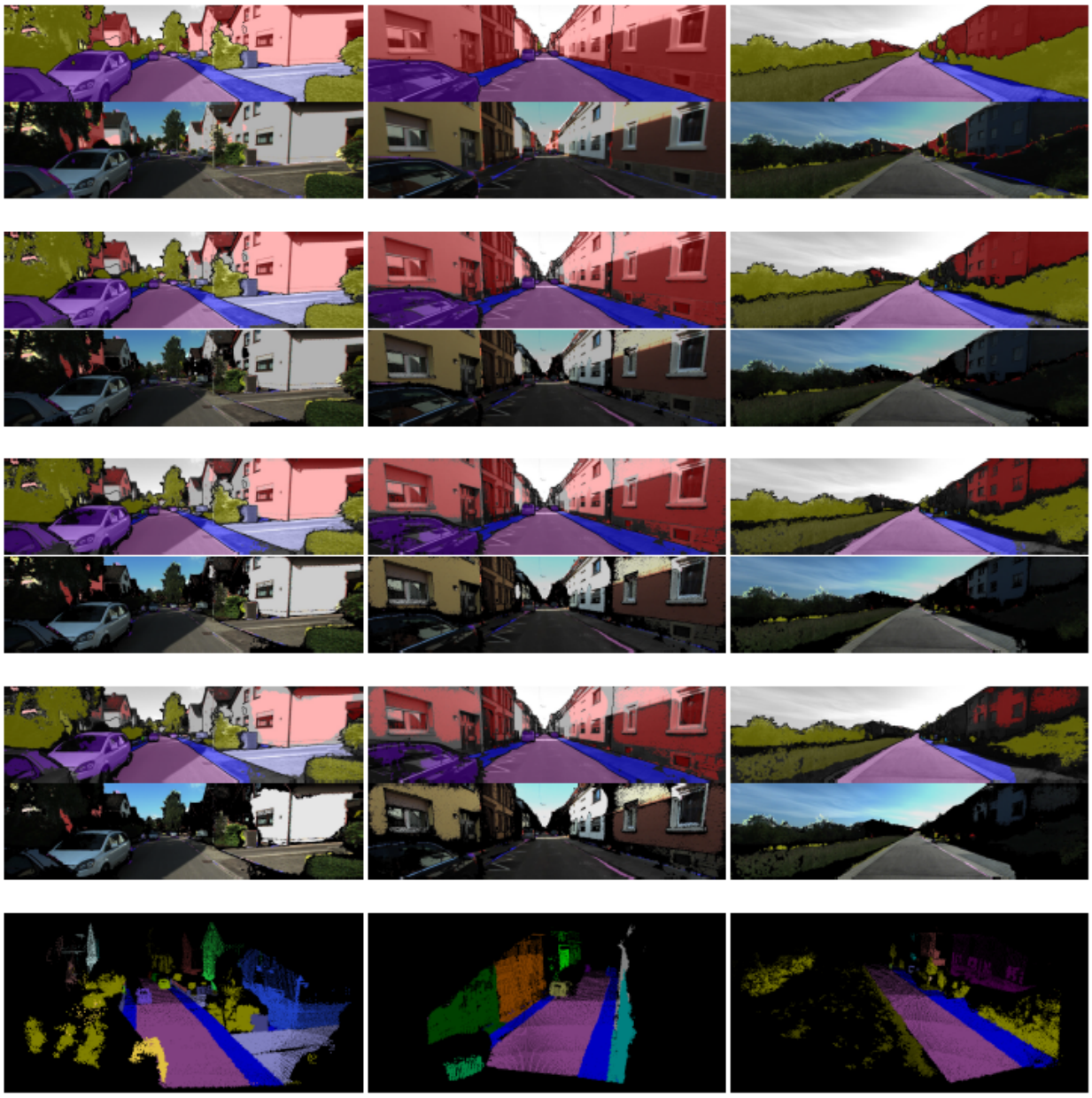
(d) Accuracy (Instance)

Figure 4.7: **Performance wrt. estimated pixels.** This figure shows the average Jaccard Index (a, c) and the average accuracy (b, d) for semantic segmentation (top, including the "Fully Conn. CRF" baseline) and instance segmentation (bottom) when estimating only a fraction of the pixels which is selected according to the uncertainty/entropy in our predictions.



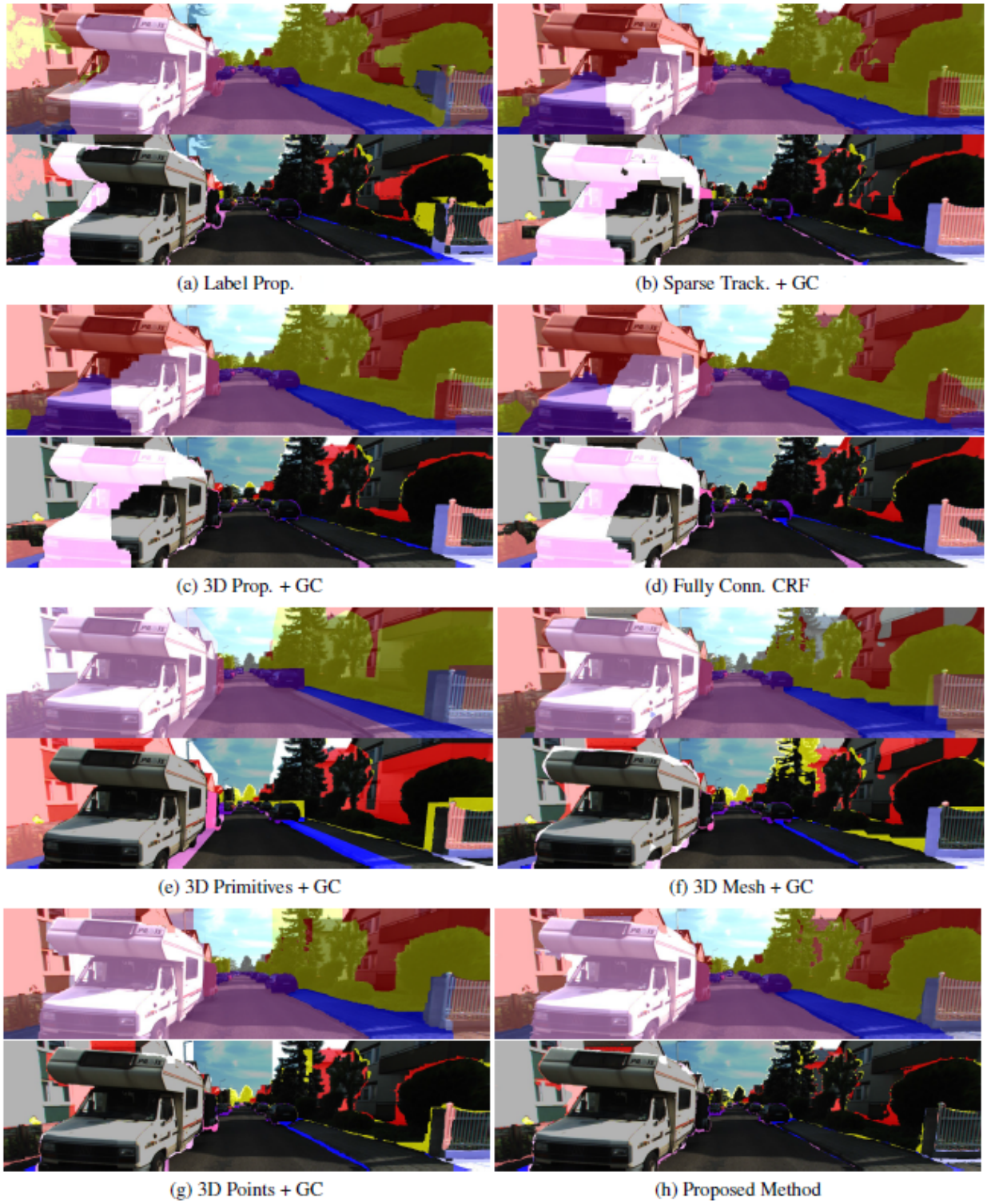
(a) Scene 1 (b) Scene 2 (c) Scene 3

Figure 4.8: **Qualitative semi-dense semantic results.** Each subfigure shows from top-to-bottom: the input image with inferred semantic segmentation and the errors with respect to 2D ground truth annotation where colors indicate the groundtruth label. The second, third and fourth row of subfigures show the results at 90%, 80% and 70% density, respectively. The last row shows the corresponding semantic 3D point cloud.



(a) Scene 1 (b) Scene 2 (c) Scene 3

Figure 4.9: **Qualitative semi-dense instance results.** Each subfigure shows from top-to-bottom: the input image with inferred instance segmentation and the errors with respect to 2D ground truth annotation where colors indicate the groundtruth semantic label. The second, third and fourth row of subfigures show the results at 90%, 80% and 70% density, respectively. The last row shows the corresponding instance 3D point cloud (random colors).



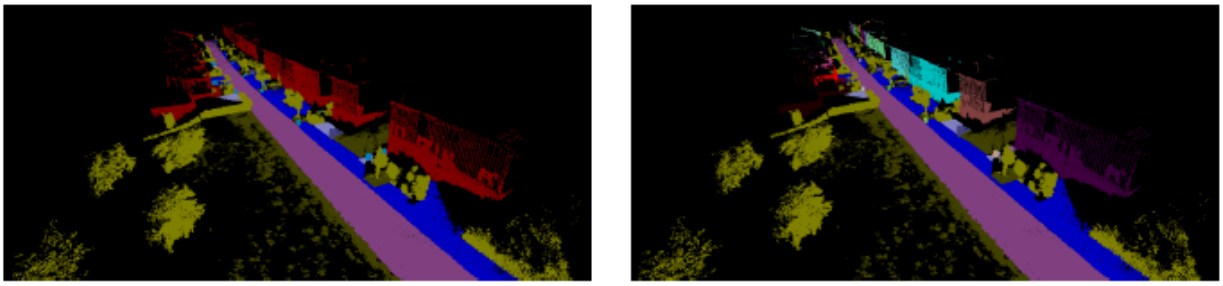
(a) Label Prop. (b) Sparse Track. + GC

(c) 3D Prop. + GC (d) Fully Conn. CRF

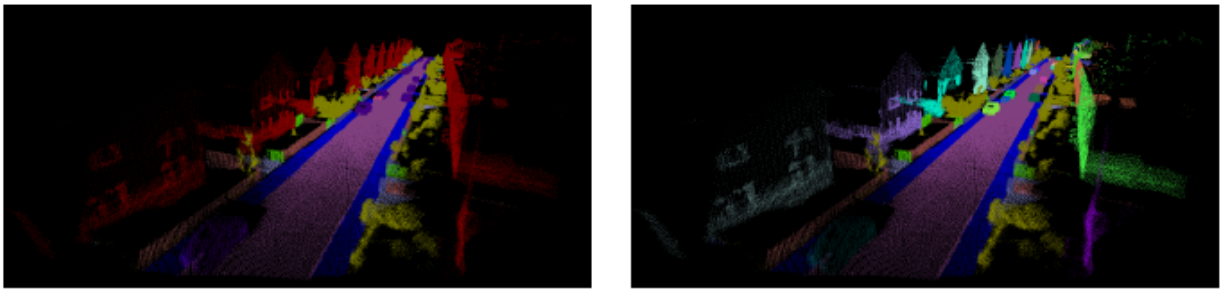
(e) 3D Primitives + GC (f) 3D Mesh + GC

(g) 3D Points + GC (h) Proposed Method

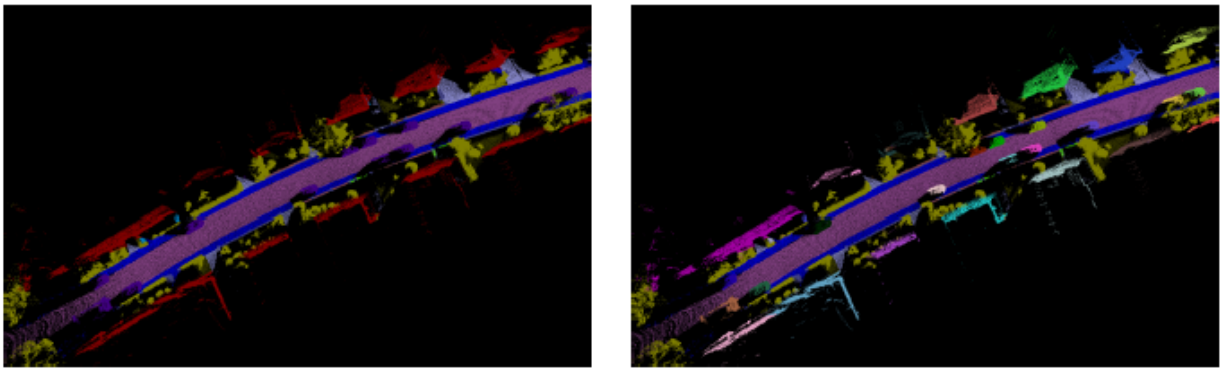
Figure 4.10: **Comparison to baselines.** Each subfigure shows from top-to-bottom: the input image with inferred semantic segmentation and the errors with respect to 2D ground truth annotation, where colors indicate ground truth labels.



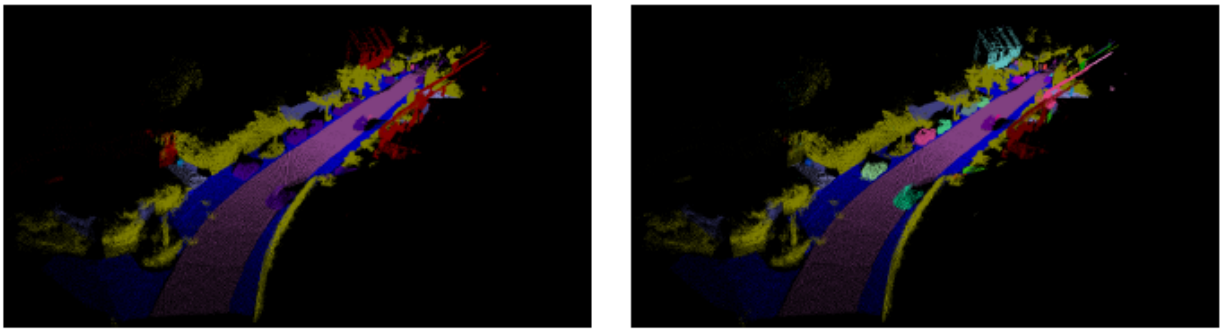
(a) Point Cloud 1



(b) Point Cloud 2



(c) Point Cloud 3



(d) Point Cloud 4

Figure 4.11: **Inferred 3D point clouds.** Left: Semantic results. Right: Instance results (random colors).

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Summary

In this dissertation, we present new techniques and applications related to depth. A brief summary is provided as follows:

In Chapter 2, we propose a robust and efficient 3D point cloud fine registration approach by enhancing ICP with a new cost function that balances the significance of structural and photometric features with dynamically adjusted weights to improve the error minimization process. We also introduce a novel outlier rejection method, which adaptively sets the outlier distance threshold in each ICP iteration, while taking into account both the 3D structural features of the object and the spatial distances of the SIFT feature pairs. For experiments, we show that our contributions can achieve superior results than other related methods, both in terms of registration accuracy and efficiency. In particular, we demonstrate our approach in several challenging scenarios, involving objects with symmetrical structures and alignment with large camera view displacements.

In Chapter 3, we propose two approaches from different aspects of single depth image super-resolution: a Coupled Dictionary Learning-based approach with Local Constraint (CDLLC) and an Edge-Guided approach (EG). In CDLLC, a robust coupled dictionary learning method with locality coordinate constraints is introduced to reconstruct the corresponding high resolution depth map. We also incorporate an adaptively regularized shock filter to simultaneously reduce the jagged noise and sharpen the edges. Furthermore, a joint reconstruction and smoothing framework is proposed with an L0 gradient smooth constraint, making the reconstruction more robust to noise. In the edge-guided approach, a novel framework for the single depth image super-resolution is proposed. In the framework, the upscaling of a single depth image is guided by a high-resolution edge map, which is constructed from the edges of the low-resolution depth image through a Markov

random field optimization in a patch synthesis based manner. We also explore the self-similarity of patches during the edge construction stage, when limited training data are available. With the guidance of the high-resolution edge map, we propose to upsample the high-resolution depth image through a modified joint bilateral filter. From experimental results, our method not only has better objective performance (i.e., for EG, it reduces 29% error on average compared with state-of-the-art methods in terms of Percent of Error score metric), but also helps avoid artifacts introduced by direct texture prediction, reduces jagged artifacts, and preserves sharp edges.

In Chapter 4, we propose a novel dataset comprising 400k images, laser point clouds, and annotations for all objects in street views. We also present a method for semantic instance labeling of large datasets from annotated 3D primitives. In the presence of 3D data, our method yields better results compared to several state-of-the-art 2D label transfer baselines while lowering annotation time. Furthermore, our method results in temporally consistent instance labels, and explicitly exposes label uncertainty.

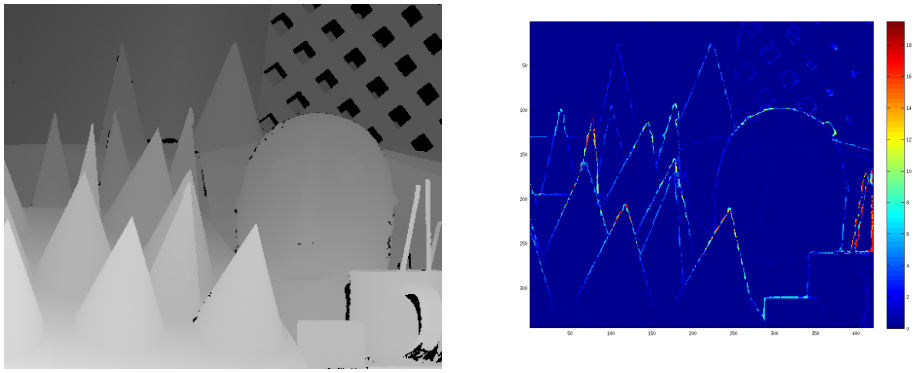
5.2 Future Work

In what follows, we propose some possible future works and directions related to the depth processing:

- Joint edge reconstruction and edge-guided super-resolution
- Depth video super-resolution
- Joint inference on multiple sequential frames
- Semantic segmentation on dynamic scene

5.2.1 Joint edge reconstruction and edge guided super resolution

From the previous depth super-resolution results, we can see that the edge plays an important role in helping super-resolution. In the edge-guided method, although we reconstruct a high-resolution



(a) (b)

Figure 5.1: **Error map of the super-resolution result of the edge-guided method.** (a) Original depth image. (b) Corresponding error map (encoded with jet color).

edge to help bilateral interpolation, most of the errors are still located around the edges in the final reconstructed high-resolution depth as shown in the Figure 5.1. Therefore, the possible future work could be estimating a more accurate edge map.

One possibility is to jointly estimate the high-resolution edges and super resolved depth image. On one hand, the estimated edges could help depth image interpolation. On the other hand, the interpolated depth image will benefit the edge reconstruction as well.

The other possibility is to use the pre-registered high-resolution color image to help edge estimation. The high-resolution edge can be inferred by the combination of the high-resolution color image and the low-resolution input depth map.

5.2.2 Depth video super-resolution

As the future work, we also aim to extend our single depth image super-resolution idea to spatio-temporal domain (i.e., upsampling depth videos). We plan to apply our algorithm on each frame of video sequences. Since we know the 3D structure of the scene from the depth images as well, we can register the point cloud of each view and figure out the correspondences across different views.

Therefore, we can jointly infer the upsampled depth images across different frames based on the correspondences. Since depth images are more independent of light/view variances compared to color images, the depth continuity is less sensitive to view changes in different frames, and thus, depth edges are more temporally consistent. So, promising results on video sequences are also expected.

5.2.3 Joint inference on multiple sequential frames

In our semantic/instance labeling method, since the obtained 3D instance annotations are associated with a single object in 3D, it already imposes somewhat soft temporal consistency constraint in our model. However, we can further impose the temporal constraint by jointly inferring the label of multiple frames in a sliding window manner. One potential problem is the trade-off between computation and the labeling quality, as the introduction of multiple fields will add significant computational burden during inference and learning.

It would also be interesting to see if the large scale dataset that we created will boost the performance for learning-based methods such as the fully convolutional networks [86].

5.2.4 Semantic segmentation on dynamic scenes

While creating a large scale dataset for semantic and instance segmentation, we only focused on the static scene elements which dominate suburban scene. One possibility of our future work is to extend our semantic segmentation method to dynamic scenes. We propose to use the objectness idea for handling the dynamic objects [20, 93]. Moreover, the incorporation of instance loss functions [1] will be another promising direction for the future.

Appendix A

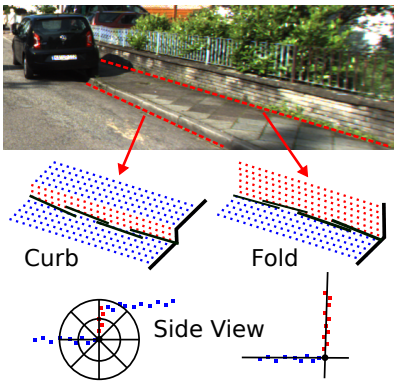
FOLD AND CURB DETECTION

We detect folds and curbs in the 3D point cloud for disambiguating the semantic class at object boundaries. We first extract all relevant object class boundaries by thresholding the gradient over semantic classes in the annotated 3D point cloud (i.e., we sweep a 3D gradient operator over the semantic 3D point cloud). For each boundary point, we fit two perpendicular 3D planes and extract their intersection in terms of a 3D fold (see Figure 4.3b, right). The sole exception are boundaries between road and sidewalk for which we detect the bottom part (of the curb) by training an SVM (Support Vector Machine) on shape context features [9] (see Figure 4.3b, left). Due to the small elevation of the curb and the noise in the 3D data we found this to perform better than 3D plane fitting in terms of separating the objects in 3D.

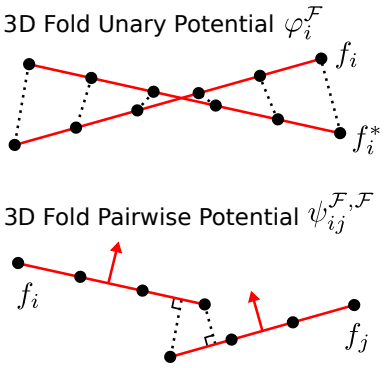
As the fold detections are noisy, we model the true fold location as a random variable and penalize the deviation of the estimate f from the detection f^* while encouraging continuity/smoothness. We associate a random variable $f_i \in \mathbb{F}$ with each 3D fold or curb $i \in \mathcal{F}$ which specifies the location and orientation of the fold segment in 3D. We discretize the set of possible fold segments for each detection by sampling from a local neighborhood around the parameters of the detection, i.e., we have $\mathbb{F} = \{1, \ldots, F\}$, where F is the number of discrete sample points. Each sample is associated with the corresponding fold segment parameters. We formulate a CRF model for optimizing the placement of fold/curb segments with an energy function which encourages smoothness of adjacent segments in 2D:

$$E(\mathbf{f}) = \sum_{i \in \mathcal{F}} \varphi_i^{\mathcal{F}}(f_i) + \sum_{i,j \in \mathcal{F}} \psi_{ij}^{\mathcal{F},\mathcal{F}}(f_i, f_j) \tag{A.1}$$

3D Fold/Curb Unary Potentials: The unary potential for the 3D fold segments and curbs is



(a) 3D Fold Detection



(b) 3D Fold Potentials

Figure A.1: **Illustration of the fold/curb detection in our model.** (a) Geometric structures such as folds and curbs are detected in the 3D point cloud by fitting planes and training a classifier based on shape context. (b) We model the uncertainty in the folds by introducing an auxiliary random variable f_i with each of them and connect adjacent folds to encourage smoothness.

specified by a quadratic loss on the deviation of the estimated fold f_i from its 3D detection f_i^*:

$$\varphi_i^{\mathcal{F}}(f_i) = w^{\mathcal{F}} \sum_{c \in \mathcal{C}} \|\boldsymbol{\kappa}_i(f_i, c) - \boldsymbol{\kappa}_i(f_i^*, c)\|_2^2 \tag{A.2}$$

Here, $\mathcal{C} \subset [0, 1]$ is a finite set of 1D control points along the fold segment and $\boldsymbol{\kappa}_i(f_i, c)$ returns the corresponding 3D point. The potential is illustrated in Figure A.1b (top).

3D Fold/Curb Pairwise Potentials: For smoothing the boundaries, we introduce a pairwise term which encourages continuity between neighboring fold segments and curbs

$$\psi_{ij}^{\mathcal{F},\mathcal{F}}(f_i, f_j) = \begin{cases} \phi_{ij}^{\mathcal{F},\mathcal{F}}(f_i, f_j) & \text{if } (i,j) \in \mathcal{N}_{\mathcal{F}} \\ 0, & \text{otherwise} \end{cases} \tag{A.3}$$

where smoothness of neighboring folds is defined via

$$\begin{aligned}\phi_{ij}^{\mathcal{F},\mathcal{F}}(f_i, f_j) &= w_1^{\mathcal{F},\mathcal{F}}\left(1 - \frac{|\boldsymbol{\pi}_i(f_i)^T \cdot \boldsymbol{\pi}_j(f_j)|}{\|\boldsymbol{\pi}_i(f_i)^T\|_2 \, \|\boldsymbol{\pi}_j(f_j)\|_2}\right) \\ &\quad + w_2^{\mathcal{F},\mathcal{F}} dist\left(\boldsymbol{\pi}(\boldsymbol{\kappa}_i(f_i, 1)), \boldsymbol{\pi}_j(f_j)\right) \\ &\quad + w_2^{\mathcal{F},\mathcal{F}} dist\left(\boldsymbol{\pi}(\boldsymbol{\kappa}_j(f_j, 0)), \boldsymbol{\pi}_i(f_i)\right)\end{aligned}$$

and $\mathcal{N}_{\mathcal{F}}$ denotes the set of neighboring folds in 3D, i.e., folds for which the endpoint of one fold segment is within a small distance from the startpoint of the next segment. The 3D point $\boldsymbol{\kappa}(\cdot,\cdot)$ is defined as above, $\boldsymbol{\pi}(\cdot)$ projects a point or fold segment from 3D to 2D, and $dist(\cdot,\cdot)$ denotes the shortest distance of a 2D point to a 2D fold segment. We use scaled normals to represent fold segments $\boldsymbol{\pi}_i(f_i)$ in 2D (i.e., $\boldsymbol{\pi}_i(f_i)^T\mathbf{p} = 1$ for all pixels $\mathbf{p} \in \mathbb{R}^2$ on the 2D fold). This potential is illustrated in Figure A.1b (bottom).

Inference: Eqn. A.1 corresponds to a non-loopy pairwise CRF as folds are connected in chains, e.g., along the sidewalk-road boundary. We obtain a global minimizer of the corresponding Gibbs energy via belief propagation. The parameters of the model have been set empirically to yield smooth results.

VITA

Jun Xie was born in Hangzhou, China. She received the B.S. degree in electrical and computer engineering from Shanghai Jiao Tong University, Shanghai, China, in 2011. She is currently a fifth year Ph.D. student in electrical engineering department from the University of Washington, Seattle. Her research interests include computational photography, interactive computer graphics and computer vision.

BIBLIOGRAPHY

[1] Faruk Ahmed, Daniel Tarlow, and Dhruv Batra. Optimizing Expected Intersection-over-Union with Candidate-Constrained CRFs. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2014)*, 2015. 95

[2] Henrik Andreasson and Achim Lilienthal. Vision Aided 3D Laser Scanner Based Registration. In *European Conference on Mobile Robots (ECMR)*, 2007. 4

[3] Henrik Andreasson and Todor Stoyanov. Real Time Registration of RGB-D Data using Local Visual Features and 3D-NDT Registration. In *Semantic Perception, Mapping and Exploration Workshop (SPME)*, May 2012. 4

[4] Mathieu Aubry, Sylvain Paris, Samuel W. Hasinoff, Jan Kautz, and Frédo Durand. Fast Local Laplacian Filters: Theory and Applications. *ACM Trans. Graph.*, 33(5):167:1–167:14, Sep 2014. 32

[5] Vijay Badrinarayanan, Ignas Budvytis, and Roberto Cipolla. Mixture of trees probabilistic graphical model for video segmentation. *International Journal of Computer Vision*, 110(1):14–29, 2014. 69, 70

[6] Vijay Badrinarayanan, Fabio Galasso, and Roberto Cipolla. Label Propagation in Video Sequences. In *CVPR*, 2010. 68, 69

[7] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM Trans. on Graphics*, 28(3), Jul 2009. 49

[8] Jens Behley, Volker Steinhage, and Armin B. Cremers. Performance of histogram descriptors for the classification of 3D laser range data in urban environments. In *ICRA*, 2012. 71

[9] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, 2002. 96

[10] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999. 77

[11] Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992. 4, 11

[12] Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. Sparse Iterative Closest Point. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*, SGP '13, pages 113–123, 2013. 4, 17, 18, 23, 24, 25, 26

[13] Yuri Boykov and Vladimir Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:1124–1137, 2004. 80

[14] Gabriel Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and Recognition using SfM Point Clouds. In *ECCV*, 2008. 71

[15] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic Object Classes in Video: A High-definition Ground Truth Database. *Pattern Recognition Letters*, 30(2):88–97, 2009. 71

[16] Ignas Budvytis, Vijay Badrinarayanan, and Roberto Cipolla. Label propagation in complex video sequences using semi-supervised learning. In *BMVC*, 2010. 69

[17] Ayan Chakrabarti, A. N. Rajagopalan, and Rama Chellappa. Super-Resolution of Face Images Using Kernel PCA-Based Prior. *IEEE Trans. on Multimedia*, 9(4):888–892, June 2007. 29

[18] Hong Chang Chang, Yeung Dit-Yan, and Yimin Xiong. Super-resolution through neighbor embedding. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004. 29

[19] Rick Chartrand and Wotao Yin. Iteratively reweighted algorithms for compressive sensing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3869–3872, Mar 2008. 4, 17, 18, 23, 24, 26

[20] Liang-Chieh Chen, Sanja Fidler, Alan L. Yuille, and Raquel Urtasun. Beat the MTurkers: Automatic Image Labeling from Weak 3D Supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 71, 72, 95

[21] Yang Chen and Gerard Medioni. Object modeling by registration of multiple range images. In *IEEE International Conference on Robotics and Automation*, pages 2724–2729 vol.3, Apr 1991. 4, 11, 12

[22] Dmitry Chetverikov, Dmitry Stepanov, and Pavel Krsek. Robust euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23:299–309, 2005. 4, 17, 18, 23, 24, 26

[23] Sunghyun Cho and Seungyong Lee. Fast Motion Deblurring. *ACM Trans. Graph.*, 28(5):145:1–145:8, Dec 2009. 37

[24] Ouk Choi and Seung-Won Jung. A Consensus-Driven Approach for Structure and Texture Aware Depth Map Upsampling. *IEEE Transactions on Image Processing*, 23(8):3321–3335, Aug 2014. 31

[25] Haili Chui, Anand Rangarajan, Jie Zhang, and Christiana Morison Leonard. Unsupervised Learning of an Atlas from Unlabeled Point-Sets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):160–172, Jan 2004. 5

[26] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset. In *CVPR Workshop on The Future of Datasets in Vision*, 2015. 71, 72

[27] Antonio Criminisi, Toby Sharp, Carsten Rother, and Patrick P'erez. Geodesic Image and Video Editing. *ACM Trans. Graph.*, 29(5):134:1–134:15, Nov 2010. 32, 36

[28] Yan Cui, Sebastian Schuon, Sebastian Thrun, Didier Stricker, and Christian Theobalt. Algorithms for 3D Shape Scanning with a Depth Camera. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(5):1039–1050, May 2013. 30

[29] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance Capture from Sparse Multi-view Video. *ACM Trans. Graph.*, 27(3):98:1–98:10, Aug 2008. 4

[30] James Diebel and Sebastian Thrun. An Application of Markov Random Fields to Range Sensing. In *NIPS*, pages 291–298, 2005. 31

[31] Chao Dong, ChenChange Loy, Kaiming He, and Xiaoou Tang. Learning a Deep Convolutional Network for Image Super-Resolution. *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, 2014. v, 29, 54, 58, 59, 60, 62, 63, 64

[32] Sebastien Druon, Marie-Jose Aldon, and Andre Crosnier. Color Constrained ICP for Registration of Large Unstructured 3D Color Data Sets. In *IEEE International Conference on Information Acquisition*, pages 249–255, Aug 2006. 5, 15

[33] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 71

[34] Zeev Farbman, Raanan Fattal, and Dani Lischinski. Diffusion Maps for Edge-aware Image Editing. *ACM Trans. Graph.*, 29(6):145:1–145:10, Dec 2010. 32

[35] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans. on Graphics*, 27(3):67:1–67:10, 2008. 32

[36] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science, 2004. 47

[37] David Ferstl, Christian Reinbacher, Rene Ranftl, R. Matthias, and Horst Bischof. Image Guided Depth Upsampling Using Anisotropic Total Generalized Variation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 993–1000, Dec 2013. v, 30, 31, 55, 58, 59, 62, 63

[38] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24(6):381–395, June 1981. 4

[39] Mario Frank, Matthias Plaue, Holger Rapp, Ullrich Köthe, Bernd Jähne, and Fred A Hamprecht. Theoretical and Experimental Error Analysis of Continuous-Wave Time-Of-Flight Range Cameras. *Optical Engineering*, 48(1), 2009. 32

[40] William T. Freeman, Thouis R. Jones, and Egon C. Pasztor. Example-based Superresolution. *Computer Graphics and Applications*, 22(2):56–65, 2002. 29, 31

[41] Jingjing Fu, Shiqi Wang, Yan Lu, Shipeng Li, and Wenjun Zeng. Kinect-Like Depth Denoising. *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS)*, 2012. 32

[42] Xinbo Gao, Kaibing Zhang, Dacheng Tao, and Xuelong Li. Joint Learning for Single-Image Super-Resolution via a Coupled Constraint. *IEEE Trans. on Image Processing*, 21(2):469–480, 2012. 29

[43] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets Robotics: The KITTI Dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 71, 72

[44] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 60, 71, 72

[45] Guy Gilboa, Nir Sochen, Yehoshua Y. Zeevi, G Gilboa, and Yehoshua Y. Zeevi. Image Enhancement and Denoising by Complex Diffusion Processes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(8):1020–1036, 2004. 37, 46

[46] Daniel Glasner, Shai Bagon, and Michal Irani. Super-resolution from a single image. *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2009. 30

[47] Stephen Gould, Richard Fulton, and Daphne Koller. Decomposing a Scene into Geometric and Semantically Consistent Regions. In *ICCV*, 2009. 71

[48] Sébastien Granger and Xavier Pennec. Multi-scale EM-ICP: A Fast and Robust Approach for Surface Registration. In *Proceedings of the 7th European Conference on Computer Vision-Part IV*, ECCV '02, pages 418–432, 2002. 5

[49] Matthieu Guillaumin, Daniel Küttel, and Vittorio Ferrari. ImageNet Auto-Annotation with Segmentation Propagation. *International Journal of Computer Vision*, 110(3):328–348, 2014. 69

[50] Yoav HaCohen, Raanan Fattal, and Dani Lischinski. Image upsampling via texture hallucination. *Proc. IEEE Int. Conf. on Computational Photography (ICCP)*, pages 1–8, Mar 2010. 30

[51] Lulu He, Sen Wang, and Thrasyvoulos N. Pappas. 3D surface registration using Z-SIFT. In *IEEE International Conference on Image Processing (ICIP)*, pages 1985–1988, Sep 2011. 5

[52] Lionel Heng, Bo Li, and Marc Pollefeys. CamOdoCal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In *IEEE International Conference on Intelligent Robots and Systems*, 2013. 72

[53] Heiko Hirschmüller. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008. 79

[54] Michael Hornacek, Christoph Rhemann, Margrit Gelautz, and Carsten Rother. Depth Super Resolution by Rigid Body Self-Similarity in 3D. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 31, 49, 54, 56

[55] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single Image Super-Resolution from Transformed Self-Exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. v, 30, 54, 58, 59, 60, 61, 62, 63, 64, 65

[56] Benjamin Huhle, Timo Schairer, Philipp Jenke, and Wolfgang Straßer. Robust Non-Local Denoising of Colored Depth Data. *CVPR Workshop on TOF Camera based Computer Vision*, 2008. 32

[57] Ryo Inomata, Kenji Terabayashi, Kazunori Umeda, and Guy Godin. Lecture Notes in Computer Science. In *Advances in Visual Computing*, volume 6938 of *Lecture Notes in Computer Science*, pages 325–336. Springer Berlin Heidelberg, 2011. 4

[58] Shahram Izadi, Andrew Davison, Andrew Fitzgibbon, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Dustin Freeman. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, 2011. 30

[59] Ankit K. Jain, Lam C. Tran, Ramsin Khoshabeh, and Truong Q. Nguyen. Efficient stereo-to-multiview synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 889–892, May 2011. 61

[60] Suyog Dutt Jain and Kristen Grauman. Supervoxel-Consistent Foreground Propagation in Video. In *ECCV*, 2014. 69

[61] Mayoore Jaiswal, Jun Xie, and Ming-Ting Sun. 3D object modeling with a Kinect camera. In *Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA)*, pages 1–5, Dec 2014. 4

[62] Michal Jancosek and Tomás Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3121–3128, 2011. 77

[63] Bing Jian and Baba C. Vemuri. Robust Point Set Registration Using Gaussian Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, Aug 2011. 5

[64] Andrew Edie Johnson and Sing Bing Kang. Registration and integration of textured 3-D data. In *International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pages 234–241, May 1997. 5, 15, 17, 18, 23, 24, 25, 26

[65] Xie Jun, Hsu Yu-Feng, Feris Rogerio Schmidt, and Sun Ming-Ting. Fine registration of 3D point clouds fusing structural and photometric information using an RGB-D camera. *Journal of Visual Communication and Image Representation*, 32:194–204, 2015. 6

[66] Ali H. Kashani, William S. Owen, Nicholas Himmelman, Peter D. Lawrence, and Robert A. Hall. Laser Scanner-based End-effector Tracking and Joint Variable Extraction for Heavy Machinery. *Int. J. Rob. Res.*, 29(10):1338–1352, sep 2010. 4

[67] Martin Kiefel and Peter Gehler. Human Pose Estimation with Fields of Parts. In *ECCV*, 2014. 78

[68] Kwang I. Kim and Younghee Kwon. Single-image super-resolution using sparse regression and natural image prior. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(6):1127–1133, June 2010. 29

[69] Vladimir G. Kim, Yaron Lipman, and Thomas Funkhouser. Blended Intrinsic Maps. *ACM Trans. Graph.*, 30(4):79:1–79:12, Jul 2011. 5

[70] Georg Klein and David Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '07, pages 1–10, 2007. 4

[71] Jan Knopp, Mukta Prasad, Geert Willems, Radu Timofte, and Luc Van Gool. Hough Transform and 3D SURF for Robust Three Dimensional Classification. In *Proceedings of the 11th European Conference on Computer Vision*, pages 589–602, 2010. 23, 24

[72] Vladimir Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, Oct 2006. 48

[73] Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. Joint Bilateral Upsampling. *ACM Trans. on Graphics*, 26(3), Jul 2007. 30

[74] Philipp Krähenbühl and Vladlen Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *NIPS*, 2011. 78, 82, 83

[75] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012. 67

[76] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1817–1824, May 2011. 6, 18

[77] Kevin Lai and Dieter Fox. Object Recognition in 3D Point Clouds Using Web Data and Domain Adaptation. *IJRR*, 29(8):1019–1037, Jul 2010. 1

[78] Manuel Lang, Oliver Wang, Tunc Aydin, Aljoscha Smolic, and Markus Gross. Practical temporal consistency for image-based graphics applications. *ACM Trans. Graph.*, 31(4):34:1–34:8, Jul 2012. 30

[79] Rafael Lemuz-López and Miguel Arias-Estrada. Iterative Closest SIFT Formulation for Robust Feature Matching. In *Proceedings of the Second International Conference on Advances in Visual Computing*, ISVC'06, pages 502–513, 2006. 5, 12, 17, 18, 23, 24, 25, 26

[80] Anat Levin, William T Freeman, and Frédo Durand. Understanding Camera Trade-Offs Through a Bayesian Analysis of Light Field Projections. *Proceedings of the 10th European Conference on Computer Vision (ECCV)*, pages 88–101, 2008. 32

[81] Jing Li, Zhichao Lu, Gang Zeng, Rui Gan, and Hongbin Zha. Similarity-Aware Patchwork Assembly for Depth Image Super-resolution. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3374–3381, June 2014. 31

[82] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 71

[83] Ce Liu, Jenny Yuen, and Antonio Torralba. Nonparametric Scene Parsing via Label Transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2368–2382, 2011. 69

[84] Ming Yu Liu, Oncel Tuzel, and Yuichi Taguchi. Joint Geodesic Upsampling of Depth Images. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 30

[85] Kai-Han Lo, Yu-Chiang Frank Wang, and Kai-Lung Hua. Joint trilateral filtering for depth map super-resolution. In *Visual Communications and Image Processing (VCIP), 2013*, pages 1–6, Nov 2013. 30

[86] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 67, 95

[87] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov 2004. 5

[88] Oisin Mac Aodha, Neill D. F. Campbell, Arun Nair, and Gabriel J. Brostow. Patch Based Synthesis for Single Depth Image Super-Resolution. *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, pages 71–84, 2012. v, 31, 32, 54, 55, 56, 58, 59, 62, 63, 64, 65

[89] Mona Mahmoudi and Guillermo Sapiro. Sparse Representations for Range Data Restoration. *IEEE Trans. on Image Processing*, 21(5):2909–2915, May 2012. 31

[90] Andjelo Martinovic, Jan Knopp, Hayko Riemenschneider, and Luc Van Gool. 3D All The Way: Semantic Segmentation of Urban Scenes from Start to End in 3D. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 71

[91] Nicolas Mellado, Dror Aiger, and Niloy J. Mitra. Super 4PCS Fast Global Pointcloud Registration via Smart Indexing. *Computer Graphics Forum*, 33(5):205–215, 2014. 5, 23, 24

[92] Moritz Menze and Andreas Geiger. Object Scene Flow for Autonomous Vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 72

[93] Moritz Menze, Christian Heipke, and Andreas Geiger. Discrete Optimization for Optical Flow. In *GCPR*, 2015. 95

[94] Daniel Munoz, Andrew J. Bagnell, and Martial Hebert. Co-inference Machines for Multimodal Scene Analysis. In *ECCV*, 2012. 71

[95] Daniel Munoz, Andrew J. Bagnell, Nicolas Vandapel, and Martial Hebert. Contextual Classification with Functional Max-Margin Markov Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 71

[96] Naveen Shankar Nagaraja, Peter Ochs, Kun Liu, and Thomas Brox. Hierarchy of Localized Random Forests for Video Annotation. In *Pattern Recognition*, pages 21–30, 2012. 70

[97] Sarah Taghavi Namin, Mohammad Najafi, Mathieu Salzmann, and Lars Petersson. A Multimodal Graphical Model for Scene Analysis. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015. 71

[98] Stanley Osher and Leonid I. Rudin. Feature-oriented Image Enhancement using Shock Filters. *SIAM Journal on Numerical Analysis*, 27:919–940, 1990. 37

[99] Maks Ovsjanikov, Quentin Mérigot, Facundo Mémoli, and Leonidas J Guibas. One Point Isometric Matching with the Heat Kernel. *Comput. Graph. Forum*, 29(5):1555–1564, Jul 2010. 6

[100] Arbeláez Pablo, Maire Michael, Fowlkes Charless, and Malik Jitendra. Contour Detection and Hierarchical Image Segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, May 2011. 46

[101] Gaurav Pandey, James R. McBride, and Ryan M. Eustice. Ford campus vision and lidar data set. *The International Journal of Robotics Research*, 30:1543–1552, 2011. 5

[102] Sylvain Paris, Samuel W Hasinoff, and Jan Kautz. Local Laplacian Filters: Edge-aware Image Processing with a Laplacian Pyramid. *ACM Trans. Graph.*, 30(4):68:1–68:12, Jul 2011. 32, 36

[103] Jaesik Park, Hyeongwoo Kim, Yu-Wing Tai, Michael S. Brown, and Inso Kweon. High Quality Depth Map Upsampling for 3D-TOF Cameras. *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2011. v, 31, 55, 58, 59, 62, 63

[104] Mark Pauly, Markus Gross, and Leif P. Kobbelt. Efficient simplification of point-sampled surfaces. In *IEEE Visualization, 2002.*, pages 163–170, Nov 2002. 14

[105] Anna Petrovskaya, Anna Petrovskaya, Sebastian Thrun, and Sebastian Thrun. Model based vehicle detection and tracking for autonomous urban driving. *Autonomous Robots*, 26(2-3):123–139, 2009. 1

[106] Jeff M. Phillips, Ran Liu, and Carlo Tomasi. Outlier Robust ICP for Minimizing Fractional RMSD. In *Sixth International Conference on 3-D Digital Imaging and Modeling*, pages 427–434, Aug 2007. 4

[107] A. N. Rajagopalan, Arnav Bhavsar, Frank Wallhoff, and Gerhard Rigoll. Resolution Enhancement of PMD Range Maps. In *Proceedings of the 30th DAGM Symposium on Pattern Recognition*, pages 304–313, 2008. 30

[108] Hayko Riemenschneider, András Bódis-Szomorú, Julien Weissenberg, and Luc Van Gool. Learning Where to Classify in Multi-view Semantic Segmentation. In *ECCV*, 2014. 71

[109] Szymon Rusinkiewicz and Marc Levoy. Efficient Variants of the ICP Algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2001. 4

[110] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 1409.0575, 2015. 67

[111] Bryan Russell, Antonio Torralba, Kevin Murphy, and William Freeman. LabelMe: A Database and Web-Based Tool for Image Annotation. *IJCV*, 77:157–173, 2008. 71

[112] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast Point Feature Histograms (FPFH) for 3D registration. In *IEEE International Conference on Robotics and Automation*, pages 3212–3217, May 2009. 23, 24

[113] Faisal Salem and Andrew E. Yagle. Non-Parametric Super-Resolution Using a Bi-Sensor Camera. *IEEE Trans. on Multimedia*, 15(1):27–40, 2013. 30

[114] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nesic, Xi Wang, and Porter Westling. High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth. *German Conference on Pattern Recognition*, 2014. 57

[115] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. Journal of Computer Vision*, 47:7–42, Apr 2002. 57

[116] Sebastian Schuon, Christian Theobalt, James Davis, and Sebastian Thrun. Lidarboost: Depth Superresolution for tof 3D Shape Scanning. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009. 30

[117] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context. *International Journal of Computer Vision*, 81:2–23, 2009. 71, 75

[118] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor Segmentation and Support Inference from RGB-D Images. In *ECCV*, 2012. 71

[119] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 567–576, 2015. 71

[120] Nikolce Stefanoski, Can Bal, Manuel Lang, Oliver Wang, and Aljoscha Smolic. Depth estimation and depth enhancement by diffusion of depth features. *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pages 1247–1251, 2013. 30

[121] Jrgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 573–580, oct 2012. 18

[122] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A Concise and Provably Informative Multi-scale Signature Based on Heat Diffusion. In *Proceedings of the Symposium on Geometry Processing*, SGP '09, pages 1383–1392, 2009. 6

[123] Jian Sun, Jiejie Zhu, and Marshall F. Tappen. Context-constrained hallucination for image super-resolution. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 231–238, Jun 2010. 30, 49

[124] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense Point Trajectories by GPU-Accelerated Large Displacement Optical Flow. In *ECCV*, 2010. 80, 82, 83

[125] Yu-Wing Tai, Wai-Shun Tong, and Chi-Keung Tang. Perceptually-Inspired and Edge-Directed Color Image Super-Resolution. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1948–1955, 2006. 30

[126] Hiroyuki Takeda, Sina Farsiu, and Peyman Milanfar. Deblurring Using Regularized Locally Adaptive Kernel Regression. *IEEE Trans. on Image Processing*, 17(4):550–563, Apr 2008. 37

[127] Gary K. L. Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C. Langbein, Yonghuai Liu, David Marshall, Ralph R. Martin, Xian-Fang Sun, and Paul L. Rosin. Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7), Jul 2013. 4

[128] Yi Tang, Pingkun Yan, Yuan Yuan, and Xuelong Li. Single-image super-resolution via local learning. *Int. Journal of Mach. Learn. Cybern.*, 2(1):15–23, 2011. 29

[129] Art Tevs, Martin Bokeloh, Michael Wand, Andreas Schilling, and Hans Peter Seidel. Isometric registration of ambiguous and partial data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1185–1192, June 2009. 6

[130] Ivana Tosic and Sarah Drewes. Learning Sparse Representations of Depth. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):941–952, 2011. 31

[131] Ivana Tosic and Sarah Drewes. Learning Joint Intensity-Depth Sparse Representations. *IEEE Trans. on Image Processing*, 23(5):2122–2132, May 2014. 31

[132] Joel A. Tropp and Anna C. Gilbert. Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit. *IEEE Transactions on Information Theory*, pages 4655–4666, 2007. 42

[133] Chih-Yun Tsai, De-An Huang, Min-Chun Yang, Li-Wei Kang, and Yu-Chiang Frank Wang. Context-aware Single Image Super-resolution Using Locality-constrained Group Sparse Representation. *Proc. IEEE Int. Conf. on Visual Communications and Image Processing (VCIP)*, 2012. 34, 56

[134] David Tsai, Matthew Flagg, Atsushi Nakazawa, and James M Rehg. Motion Coherent Tracking Using Multi-label MRF Optimization. *International Journal of Computer Vision*, 100(2):190–202, 2012. 69

[135] Antoine Vacavant, Adelarde Albouy-Kissi, and Pierre-Yves Menguy. Fast Smoothed Shock Filtering. *Proc. IEEE Conf. on Pattern Recognition (ICPR)*, 2012. 37

[136] Julien P. C. Valentin, Sunando Sengupta, Jonathan Warrell, Ali Shahrokni, and Philip H. S. Torr. Mesh based semantic modelling for indoor and outdoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 71

[137] Sudheendra Vijayanarasimhan and Kristen Grauman. Active Frame Selection for Label Propagation in Videos. In *ECCV*, 2012. 69, 70, 80, 82, 83

[138] Vibhav Vineet, Glenn Sheasby, Jonathan Warrell, and Philip H. S. Torr. PoseField: An Efficient Mean-Field Based Method for Joint Estimation of Human Pose, Segmentation, and Depth. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2013. 78

[139] Shenlong Wang, Lei Zhang, Yan Liang, and Quan Pan. Semi-Coupled Dictionary Learning with Applications to Image Super-Resolution and Photo-Sketch Synthesis. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2216–2223, 2012. 29, 55, 56

[140] Kai M. Wurm, Henrik Kretzschmar, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. Identifying vegetation from laser data in structured outdoor environments. In *Robotics and Autonomous Systems*, volume 62, pages 675–684, 2014. 1

[141] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. SUN3D: A Database of Big Spaces Reconstructed Using SfM and Object Labels. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 71

[142] Jianxiong Xiao and Long Quan. Multiple view semantic segmentation for street view images. In *IEEE International Conference on Computer Vision (ICCV)*, 2009. 71

[143] Jun Xie, Cheng-Chuan Chou, Rogerio Schmidt Feris, and Ming-Ting Sun. Single depth image super resolution and denoising via coupled dictionary learning with local constraints and shock filtering. *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*, 2014. 33

[144] Jun Xie, R Feris, and Ming-Ting Sun. Edge Guided Single Depth Image Super Resolution. *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 2014. 33

[145] Jun Xie, Rogerio Schmidt Feris, and Ming-Ting Sun. Edge-Guided Single Depth Image Super Resolution. *IEEE Transactions on Image Processing*, 25(1):428–438, Jan 2016. 33

[146] Jun Xie, Rogerio Schmidt Feris, Shiaw-Shian Yu, and Ming-Ting Sun. Joint Super Resolution and Denoising From a Single Depth Image. *Multimedia, IEEE Transactions on*, 17(9):1525–1537, Sep 2015. 33

[147] Jun Xie, Yu-Feng Hsu, R S Feris, and Ming-Ting Sun. Fine registration of 3D point clouds with iterative closest point using an RGB-D camera. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2904–2907, May 2013. 6

[148] Jun Xie, Martin Kiefel, Ming-Ting Sun, and Andreas Geiger. Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 69

[149] Zhiwei Xiong, Dong Xu, Xiaoyan Sun, and Feng Wu. Example-based super-resolution with soft information and decision. *IEEE Trans. on Multimedia*, 15(6):1458–1465, 2013. 29

[150] Jia Xu, Alexander G. Schwing, and Raquel Urtasun. Tell Me What You See and I will Show You Where It Is. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 69

[151] Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. Image Smoothing via L0 Gradient Minimization. *ACM Trans. on Graphics*, 30(6):174:1–174:12, 2011. 32, 44

[152] Chih-Yuan Yang and Ming-Hsuan Yang. Fast Direct Super-Resolution by Simple Functions. *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2013. 29

[153] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image Super-Resolution Via Sparse Representation. *IEEE Trans. on Image Processing*, 19(11):2861–2873, 2010. v, 29, 34, 41, 55, 56, 58, 59, 62, 63, 64, 65

[154] Min Chun Yang and Yu Chiang Frank Wang. A Self-Learning Approach to Single Image Super-Resolution. *IEEE Trans. on Multimedia*, 15(3):498–508, Apr 2013. 29

[155] Qingxiong Yang, Ruigang Yang, James Davis, and David Nistér. Spatial-depth Super Resolution for Range Images. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007. 30

[156] Kai Yu, Tong Zhang, and Yihong Gong. Nonlinear Learning using Local Coordinate Coding. *Advances in Neural Information Processing Systems (NIPS)*, 2009. 34, 35

[157] Matthew Zeiler. ADADELTA: An adaptive learning rate method. *arXiv preprint*, 1212.5701, 2012. 78

[158] Roman Zeyde, Michael Elad, and Matan Protter. On Single Image Scale-up using Sparse Representations. *Curves and Surfaces*, 6920:711–730, 2010. v, 29, 55, 56, 58, 59, 62, 63, 64, 65

[159] Hongyi Zhang, Andreas Geiger, and Raquel Urtasun. Understanding High-Level Semantics by Modeling Traffic Patterns. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 68

[160] Zhengyou Zhang. Iterative Point Matching for Registration of Free-form Curves and Surfaces. *Int. J. Comput. Vision*, 13(2):119–152, Oct 1994. 4, 15, 17, 18, 23, 24, 26

[161] Ziming Zhang, Lubor Ladicky, Philip H.S. Torr, and Amir Saffari. Learning Anchor Planes for Classification. *Advances in Neural Information Processing Systems (NIPS)*, 2011. 34

[162] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object Detectors Emerge in Deep Scene CNNs. *International Conference on Learning Representations (ICLR)*, 1412.6856, 2015. 67

[163] Yukun Zhu, Raquel Urtasun, Ruslan Salakhutdinov, and Sanja Fidler. segDeepM: Exploiting Segmentation and Context in Deep Neural Networks for Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 67

[164] Maria Zontak and Michal Irani. Internal statistics of a single natural image. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 977–984, June 2011. 30