Depth-Assisted Semantic Segmentation, Image Enhancement and Parametric Modeling

Chenxi Zhang

University of Kentucky, pigzcx@gmail.com

Click here to let us know how access to this document benefits you.

Recommended Citation

https://uknowledge.uky.edu/cs_etds/27
STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student’s advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student’s thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Chenxi Zhang, Student

Dr. Ruigang Yang, Major Professor

Dr. Miroslaw Truszczynski, Director of Graduate Studies
DEPTH-ASSISTED SEMANTIC SEGMENTATION, IMAGE ENHANCEMENT AND PARAMETRIC MODELING

DISSERTATION

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the College of Engineering at the University of Kentucky

By
Chenxi Zhang
Lexington, Kentucky

Director: Dr. Ruigang Yang, Professor of Computer Science Lexington, Kentucky 2014

Copyright © Chenxi Zhang 2014
This dissertation addresses the problem of employing 3D depth information on solving a number of traditional challenging computer vision/graphics problems. Humans have the abilities of perceiving the depth information in 3D world, which enable humans to reconstruct layouts, recognize objects and understand the geometric space and semantic meanings of the visual world. Therefore it is significant to explore how the 3D depth information can be utilized by computer vision systems to mimic such abilities of humans. This dissertation aims at employing 3D depth information to solve vision/graphics problems in the following aspects: scene understanding, image enhancements and 3D reconstruction and modeling.

In addressing scene understanding problem, we present a framework for semantic segmentation and object recognition on urban video sequence only using dense depth maps recovered from the video. Five view-independent 3D features that vary with object class are extracted from dense depth maps and used for segmenting and recognizing different object classes in street scene images. We demonstrate a scene parsing algorithm that uses only dense 3D depth information to outperform using sparse 3D or 2D appearance features.

In addressing image enhancement problem, we present a framework to overcome the imperfections of personal photographs of tourist sites using the rich information provided by large-scale internet photo collections (IPCs). By augmenting personal 2D images with 3D information reconstructed from IPCs, we address a number of traditionally challenging image enhancement techniques and achieve high-quality results using simple and robust algorithms.

In addressing 3D reconstruction and modeling problem, we focus on parametric modeling of flower petals, the most distinctive part of a plant. The complex structure, severe occlusions and wide variations make the reconstruction of their 3D models a challenging task. We overcome these challenges by combining data driven modeling techniques with domain knowledge from botany. Taking a 3D point cloud of an input flower scanned from a single view, each segmented petal is fitted with a scale-invariant morphable petal shape model, which is constructed from individually scanned 3D exemplar petals. Novel constraints based on botany studies are incorporated into
the fitting process for realistically reconstructing occluded regions and maintaining correct 3D spatial relations.

The main contribution of the dissertation is in the intelligent usage of 3D depth information on solving traditional challenging vision/graphics problems. By developing some advanced algorithms either automatically or with minimum user interaction, the goal of this dissertation is to demonstrate that computed 3D depth behind the multiple images contains rich information of the visual world and therefore can be intelligently utilized to recognize/understand semantic meanings of scenes, efficiently enhance and augment single 2D images, and reconstruct high-quality 3D models.

KEYWORDS: Semantic segmentation, image enhancement, 3D parametric modeling, Multiview stereo

Author’s signature: Chenxi Zhang
Date: December 18, 2014
To my beloved parents, Luning Zhang and Song Gao.
I would like to firstly thank my advisor, Dr. Ruigang Yang. I sincerely appreciate his insightful guidance, creative inspiration, and endless encouragement during my graduate studies. I have learned a lot from him about how to do research, present results, and interact with people. I am very grateful to Dr. Yang for giving me the freedom to pursue various research topics and raising my interest in the field of computer vision. His support made substantial impacts in my PhD study and dissertation work. I consider myself truly fortunate to have him being my advisor.

This dissertation is built based on part of the research projects I have done with several collaborators. Here I would like to thank my co-authors, all of whom have provided valuable efforts to different parts of the dissertation. In particular, I wish to express my deep gratitude to Scott Cohen, Brian Price, Oliver Wang, James Davis and Marc Pollefeys. It is extremely enlightening to discuss with them and our collaboration on a variety of projects is highly enjoyable.

Next I would like to express my gratitude to my committees, including Professors Nathan Jacobs, Fuhua Cheng and Samson Cheung. I appreciate them for reading my dissertation and offering thoughtful comments. I would also give my thanks to the staff of the Center for Visualization and Virtual Environments and the Director of Graduate Studies of the Computer Science Department, Professor Raphael Finkel and Professor Mirek Truszczynski who provided administrative supports.

Also many thanks to my colleagues in the GRAVITY lab: Liang Wang, Xinyu Huang, Miao Liao, Xianwang Wang, Jizhou Gao, Qing Zhang, Mao Ye, Bo Fu, Changpeng Ti, Chao Du, Yajie Zhao and Yongwook Song. Besides from the discussion we had and the invaluable support from them, I treasure all the wonderful moments we have spent together. They made my PhD time truly memorable.
Finally, I could never finish this dissertation without the support and unconditional love from my family. I would like to take this opportunity to mention my parents Song Gao and Luning Zhang, who have had the most significant impact on my life.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>iii</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>v</td>
</tr>
<tr>
<td>List of Tables</td>
<td>vii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>viii</td>
</tr>
<tr>
<td>Chapter 1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation and Contributions</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Guideline for Reading</td>
<td>8</td>
</tr>
<tr>
<td>Chapter 2 Related Work</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Automatic/Semi-automatic stereo matching</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Semantic segmentation and scene understanding</td>
<td>12</td>
</tr>
<tr>
<td>2.3 Internet Photo-based reconstruction and image enhancement</td>
<td>13</td>
</tr>
<tr>
<td>2.4 3D Flower Modeling</td>
<td>17</td>
</tr>
<tr>
<td>Chapter 3 High-quality Stereo Image/Video Matching via User Interaction</td>
<td>19</td>
</tr>
<tr>
<td>3.1 Overview and Problem Formulation</td>
<td>21</td>
</tr>
<tr>
<td>3.2 Interactive Stereo Correction</td>
<td>22</td>
</tr>
<tr>
<td>3.2.1 Model fitting tool</td>
<td>22</td>
</tr>
<tr>
<td>3.2.2 Object correspondence tool</td>
<td>24</td>
</tr>
<tr>
<td>3.2.3 Alignment tool</td>
<td>24</td>
</tr>
<tr>
<td>3.2.4 Smoothness brush tool</td>
<td>25</td>
</tr>
<tr>
<td>3.2.5 Discontinuity brush tool</td>
<td>25</td>
</tr>
<tr>
<td>3.3 Temporal Disparity Propagation and Fusion</td>
<td>26</td>
</tr>
<tr>
<td>3.3.1 Spatio-temporal Cost Volume Aggregation</td>
<td>26</td>
</tr>
<tr>
<td>3.3.2 Disparity propagation via tracking and 3D transformation</td>
<td>27</td>
</tr>
<tr>
<td>3.3.3 Fusion of Aggregated Cost and Propagated Disparity</td>
<td>30</td>
</tr>
<tr>
<td>3.4 Evaluation</td>
<td>31</td>
</tr>
<tr>
<td>3.4.1 York Stereo Dataset</td>
<td>32</td>
</tr>
<tr>
<td>3.4.2 Office Dataset</td>
<td>33</td>
</tr>
<tr>
<td>3.4.3 KITTI Vision Benchmark datasets</td>
<td>34</td>
</tr>
<tr>
<td>3.5 Summary</td>
<td>37</td>
</tr>
<tr>
<td>Chapter 4 Depth-based Semantic Segmentation of Urban Scenes</td>
<td>39</td>
</tr>
<tr>
<td>4.1 Depth Maps Recovery</td>
<td>40</td>
</tr>
<tr>
<td>4.2 Semantic Segmentation from Dense Depth Maps</td>
<td>42</td>
</tr>
<tr>
<td>4.2.1 Image over-segmentation</td>
<td>42</td>
</tr>
<tr>
<td>4.2.2 Features from dense depth map</td>
<td>43</td>
</tr>
</tbody>
</table>
4.1 Comparison of Pixel-wise percentage accuracy with [1]. Our dense depth map based approach gives best result on 7 classes. ‘Global’ is the percentage of pixels correctly classified. ‘Average’ is the average value of per-class accuracies. .............................................................. 52

4.2 Comparison of pixel-wise percentage accuracy with [1] in illumination variation test. .......................................................... 53
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Overview of the work in this dissertation. (a) Interactive stereo matching. (b) Depth-based semantic segmentation of urban scene. (c) Depth-based personal photo enhancement using internet photo collections. (d) Depth-based 3D flower parametric modeling.</td>
</tr>
<tr>
<td>3.1</td>
<td>User interface of our interactive stereo correction tools.</td>
</tr>
<tr>
<td>3.2</td>
<td>Overview of sparse feature and 3D transformation based disparity propagation.</td>
</tr>
<tr>
<td>3.3</td>
<td>Example of disparity propagation on Office 2 dataset. From left to right: key frame 1 (frame 10), key frame disparity correction (frame 10), intermediate frame (frame 20), propagated disparity map (frame 20).</td>
</tr>
<tr>
<td>3.4</td>
<td>Results of York stereo Lab1 sequence. <strong>First row:</strong> from left to right, 1) key frame in <em>blue</em> (frame 1); 2) user corrected disparity map (frame 1); 3) reference image in <em>red</em> (frame 12); 4) ground truth disparity map (frame 12). <strong>Second row:</strong> from left to right, result of 1) initial cost 2) spatiotemporal cost aggregation; 3) spatiotemporal cost aggregation+tracking and 3D transformation based propagation (<em>Ours</em>); 4) spatiotemporal aggregation+optical flow based propagation; 5) method from [2].</td>
</tr>
<tr>
<td>3.5</td>
<td>Results of York stereo Lab2 sequence. <strong>First row:</strong> from left to right, 1) key frame in <em>blue</em> (frame 20); 2) user corrected disparity map (frame 20); 3) reference image in <em>red</em> (frame 10); 4) ground truth disparity map (frame 10). <strong>Second row:</strong> from left to right, result of 1) initial cost 2) spatiotemporal cost aggregation; 3) spatiotemporal cost aggregation+tracking and 3D transformation based propagation (<em>Ours</em>); 4) spatiotemporal aggregation+optical flow based propagation; 5) method from [2].</td>
</tr>
<tr>
<td>3.6</td>
<td>Results of Office datasets. <strong>First row &amp; Third row:</strong> From left to right, 1) key frames in <em>blue</em> (<em>Office 1</em>: frame 18, <em>Office 2</em>: frame 10); 2) user corrected disparity map (<em>Office 1</em>: frame 18, <em>Office 2</em>: frame 10); 3) reference image in <em>red</em> (<em>Office 1</em>: frame 23, <em>Office 2</em>: frame 20). <strong>Second row &amp; Fourth row:</strong> 1) spatiotemporal cost aggregation+tracking and 3D transformation based propagation (<em>Ours</em>); 2) spatiotemporal cost aggregation+optical flow based propagation; 3) results from [3].</td>
</tr>
<tr>
<td>3.7</td>
<td>Results of KITTI Seq 0093 datasets. <strong>First row:</strong> from left to right, 1) key frame in <em>blue</em> (frame 153); 2) user corrected disparity map (frame 153). <strong>Second row:</strong> 1) reference image in <em>red</em> (frame 155); 2) result from initial cost. <strong>Third row:</strong> from left to right, result of 1) spatiotemporal cost aggregation; 2) spatiotemporal cost aggregation+tracking and 3D transformation based propagation (<em>Ours</em>). <strong>Fourth row:</strong> 1) spatiotemporal cost aggregation+optical flow based propagation; 2) method from [3].</td>
</tr>
</tbody>
</table>
3.8 Results of KITTI Seq 0046 datasets. **First row:** from left to right, 1) key frame in blue (frame 80); 2) user corrected disparity map (frame 80);
**Second row:** 1) reference image in red (frame 84); 2) result from initial cost. **Third row:** from left to right, result of 1) spatiotemporal cost aggregation; 2) spatiotemporal cost aggregation+tracking and 3D transformation based propagation (Ours). **Fourth row:** 1) spatiotemporal cost aggregation+optical flow based propagation; 2) method from [3].

4.1 Overview of our framework

4.2 Visualization of different steps and features in our algorithm.

4.3 (a)Labeled colors for 11 object classes. (b)Split of sequence as training or testing data.

4.4 Scene Parsing result samples. From top to bottom: test image, ground truth, dense depth map inferred segmentation. Note that our algorithm solely using dense depth map is able to achieve accurate segmentation and recognition of street scene.

4.5 Robustness to lighting condition changes. In the left figure, from top to bottom: test image, ground truth, segmentation results. The left two columns are examples of training by a dusk sequence and testing on day sequence. The right two columns show results when the training and the test sequence are swapped. Table 4.2 shows the overall accuracy and comparison to numbers reported in [1].

4.6 Refinement of segmentation by multi-view fusion. From left to right: test image, result before fusion, result after fusion, ground truth. Based on temporal consistency of video sequence, some inaccurate classifications in reference view are refined by multi-view fusion.

4.7 Preliminary result of changed-view training and testing. From the side-view video sequence, we can see that our algorithm trained based on front-view sequences still obtained acceptable results.

5.1 Overview of our photo enhancements. From left to right: original image, foreground segmentation, photometric enhancement, stereoscopic image conversion, field of view expansion, and geo-tagging. The saturation (over-exposure) in right corner of the original image is removed by photometric enhancement, which is used as input for all the following enhancement applications. The field of view expansion result is intentionally resized for better figure layout. All photo enhancements are realized using Internet photo collections of the same landmark, Notre-Dame.

5.2 For images that initially failed to register to a 3D model(first image in second row), we search for a match in the temporally neighboring images(second and third images in second row). After constraining their locations nearby, we are able to successfully register the input image to one of the reconstructed models(Among the three reconstructed models in first row, the input image is registered to the right one).
5.3 Segmentation results: (top left) the original image superimposed with the bounding box prior for Grabcut, (top right) red dots for foreground color seeds and blue dots for background color seeds, (bottom left) the result of Grabcut, (bottom right) the result of our automatic approach.

5.4 Example of completed Scene where user interaction is needed to separate different foreground layers.

5.5 Single-view personal photo to stereoscopic 3D conversion. Input images (top) are converted to stereo pairs (bottom) using depth information computed from IPCs. Note: As with any stereo 3D viewing situation, the perceived depth is related to the distance of the viewer to the screen. These images have been optimized for zoomed-in on screen display. Please view zoomed-in on screen for best results.

5.6 Comparison showing the effect of incorporating our 3D model $M$ into the stereoscopic conversion. a) Original image, b) result computed using a planar approximation to the background, c) result computed using $M$. In the latter case, the rocks on the right can be seen at their correct depth level. Please view zoomed-in on screen for best results.

5.7 Results from FOV expansion, (top) original images (bottom) FOV expanded images. For the third set, the expanded background is blurred with a low-pass filter to match the input.

5.8 Median color images from selected geometrically registered images with high photometric consistency.

5.9 Comparison showing the effect of our saturation removal using internet photo collections. a) Original image, b) Result from Photoshop color adjustment, c) Result from our method. Since saturated regions have no details, simple color adjustment cannot fix the problem. Please view zoomed-in on screen for best results.

5.10 Effect of our glare removal using internet photo collections. a) Original image with user selected glare region, b) Results using Photoshop Healing and Clone tools, c) Results from our method. Please view zoomed-in on screen for best results.

5.11 Geo- and textual tagging of personal photos using IPCs of Berlin and Notre Dame. Locations depicted with green pins correspond to the query images that are matched a 3D model, geo-tagged and geo-localized. Blue pins represent images from IPC.

5.12 Comparison showing how the results depend on the number of images in database. Row 1-3: Mount Rushmore Dataset. Row 4-6: Notre Dame Dataset. From top to bottom: 2D to 3D conversion, FOV expansion, photometric enhancement. Please view zoomed-in on screen for best results.
5.13 Performance related to query image resolution. a) Original image, b) Result from original image, c) Downsampled original image (X5), d) Result from downsampled (X5) original image. The expanded region of (d) is more consistent to the quality of (c), therefore looks more natural, while the expanded region of (b) is more blurry relative to the quality of (a), which looks a little bit unnatural. Please view zoomed-in on screen for best results.

5.14 Average time used for processing under our system and Photoshop.

5.15 User study result: Percentage of preference choices between our results and Photoshop (PS) results. The four rows represent four different applications. The expert-driven PS segmentation can be deemed as ground truth. Users’ visual evaluations show that our segmentation results are comparable to ground truth. For the other three applications, our results receive much higher evaluation scores.

5.16 Normalized evaluations of Our approach vs Photoshop.

5.17 Comparison showing the effect of our enhancements using internet photo collections. a) Original image, b) Result from Photoshop, c) Result from our method. From top to bottom: foreground segmentation, 2D to 3D conversion, FOV expansion and photometric enhancement. Please view zoomed-in on screen for best results. To see more photometric enhancement comparisons with Photoshop, please refer to Figure 5.9 and Figure 5.10. More comparison results are provided in supplemental materials.

5.18 Limitations of our approach: (a) Missing data behind the columns caused bad interpolated depth values, and consequently a low-fidelity synthesized view in the column areas. (b) Another example of field-of-view expansion. The green box is the original photo. Note that the flowing water from the fountain on both the left and right sides, and the fountain rocks are blurry.

6.1 From left to right: 1) Petal database for Lily species; (2) Input image; (3) Petal segmentation; (4) Scanned 3D data; (5) Reconstructed 3D model.

6.2 Petal database for Pansies (60 exemplars).

6.3 Left: Initial segmentation. Right: Final segmentation.

6.4 Four cases of finding correspondences on input shape. Red points stand for true correspondences, and green points are false correspondences.

6.5 Intermediate results of coarse-to-fine petal fitting. a) After initial alignment; b) After boundary alignment; c) After fitting with inner points; d) After adding relative depth constraints.

6.6 Example of severely occluded petal reconstruction. From left to right: 1) Scanned petal data with occlusion; 2) Reconstructed petal without texture; 3) Reconstructed petal with texture.

6.7 3D modeling of Lily species. From left to right: (a) Petal segmentation; (b) Scanned 3D data; (c) Reconstructed model without texture; (d) Reconstructed model with texture; (e)-(f): Model from different views. The orange example (last row) is scaled up for visualization purpose. Zoom-in for better visualization.
6.8 3D modeling of Pansie species. From left to right: (a) Petal segmentation; (b) Scanned 3D data; (c) Reconstructed model without texture; (d) Reconstructed model with texture; (e)-(f): Model from different views. Zoom-in for better visualization.

6.9 Constraint importance evaluation. From left to right: a) Scanned 3D data; b) Remove one constraint; c) With all constraints. From top to bottom: Comparisons with removing 1) root convergence constraint ($E_C$); 2) relative depth constraint (Eq. 6.16); 3) similarity constraint ($E_S$). Zoom-in for better visualization.

6.10 Comparison with leaf fitting method in [4]. From left to right: a) Scanned 3D data; b) Results from [4]; c) Our results. Zoom-in for better visualization.
Chapter 1 Introduction

When a 3D world is projected onto a 2D image, reconstructing back the contents of the real-world becomes an ill-posed problem which is extremely difficult to solve. Humans possess the remarkable ability of navigating and understanding the visual world by solving the inverse problem going from 2D to 3D. Computer vision, as a modern discipline, seeks to imitate such abilities of humans to reconstruct layouts, recognize objects and understand the geometric space and semantic meanings of the visual world. Those abilities of humans are remarkably attribute to humans’ perception of the depth information in 3D world. Therefore, it is natural to bring up a question: In what aspects and how can depth information assist a computer vision system?

There are a number of classical computer vision/graphics problems, of which the traditional solutions are 2D appearance based approaches, since 2D information is easy to obtain. Though remarkable progress has been achieved, the results are far from perfect. In recent years, tremendous progress has been achieved on the acquisition of 3D depth information. Obtaining high quality 3D depth information, either by passive or active stereo methods, is not a difficult task. The major objective that guided the research in this dissertation is to explore how 3D depth information can benefit traditional computer vision/graphics problems. We propose to design some novel computer vision/graphics algorithms that utilize the augmented 3D depth information for achieving a series of vision/graphics applications. Generally, we aim at covering a broad scope of vision/graphics problems, including segmenta-
tion/recognition, image enhancement, 3D parametric modeling, to demonstrate 3D depth information can provide better solutions. Specifically, we emphasize the usage of 3D depth information in solving problems in the following scenarios: semantic segmentation and scene understanding (joint segmentation and recognition) on 2D urban video sequence, personal photo enhancements using internet photo collections, high-quality 3D flower parametric modeling. For the first two scenarios, there have been priors works only using 2D appearance information. For the third one, though there have been tremendous work on 3D modeling, very few work has been carried out on flower modeling due to its complex structure and big self-occlusion. We believe that by employing the augmented 3D depth information, the solutions of those problems can be greatly improved.

The processing performed by the proposed system can be partitioned into two parts. The first part is the automatic depth map generation/acquisition. Based on different scenarios, we choose different methods for depth map generation/acquisition. For the problem of semantic segmentation on 2D video sequence, it is natural to estimate 3D depth information from 2D video sequence. For personal photo enhancements, since there is no way to automatically estimate the depth from a single image, we refer to internet photo collections taken under the same tourist sites for 3D scene reconstruction, and register personal photo to the reconstructed 3D model for depth recovery. For 3D flower modeling, high quality depth map is required, we therefore choose structured light scanning approaches to obtain accurate 3D depth information. Despite different methods for depth generation, we essentially employ image based triangulation method on multiple images (multi-view stereo) to automatically
compute the 3D depth information. To compensate for the depth inaccuracies from automatic stereo matching computation, a number of easy-to-use interactive tools are developed for correcting depth errors with minimum manual intervention. For video sequences, a novel propagation algorithm is designed to propagate the user corrected depths in key frames to intermediate frames.

The second part is the core research work in this dissertation. It aims at employing the generated 3D depth information to solve those vision/graphic sub-problems. For the first problem, we propose a framework for semantic segmentation on urban video sequence only using dense depth maps. For the second problem, we propose a framework to efficiently enhance and augment one’s own personal photos by employing the rich depth and photometric information contained in large scale internet photo collections. We specifically demonstrate algorithms for automatic foreground segmentation, 2D-to-3D conversion, field of view expansion and photometric enhancement. Our solution to the third problem focuses on developing computational methodology to realistically model flowers, specifically petals, by using captured depth maps with some Botany priors.

Figure 1.1 shows an overview of the work in this dissertation.

1.1 Motivation and Contributions

The research presented in this dissertation aims to explore how 3D depth information can benefit some typical computer vision/graphics problems. Towards this goal, we address three key scenarios for employing 3D depth information and contribute several novel algorithms that are motivated by the specific requirements and limitations
First, we address the problem of semantic segmentation of urban scenes from a monocular video sequence filmed at street level and propose an effective algorithm to address this particular problem. Semantic segmentation, which refers to the process of simultaneously classifying and segmenting objects in an image, is one of the fundamental problems of computer vision. A successful scene parsing system is of great benefit to a variety of vision applications, such as object recognition, automatic driver assistance and 3D urban modeling. While the task of segmentation traditionally relies on color information alone, using depth information has some obvious advantages.
Firstly it is invariant to lighting and/or texture variations; secondly it is invariant to camera pose and perspective changes. Therefore using depth can potentially enable successful segmentation independent of illumination or view, significantly expanding the range of operation conditions. Recently, advances in structure from motion techniques make it easier to obtain depth cues from video sequences. As a result there is notable progress in performing semantic segmentation using 3D cues. A pioneer work in using depth for outdoor scene segmentation is [1], in which the authors demonstrated that semantic segmentation is possible based solely on sparse 3D point clouds obtained from structure from motion (SFM) techniques. Given the success of [1], a natural question raised is whether dense 3D information can perform equally well, or, even better on this challenging task. The most distinct feature that differentiates our approach from existing solutions lies in the use of dense depth maps recovered via multi-view stereo matching techniques as cues to achieve accurate scene parsing. Our experiments indicate that this is true for street scene segmentation and recognition.

The second contribution of this dissertation is that we propose a framework to overcome the imperfections of personal photos of tourist sites using the rich information provided by large scale Internet photo collections. Given the growth of Internet photo collections we now have a visual index of all major cities and tourist sites in the world. However, it is still a difficult task to capture that perfect shot with your own camera when visiting these places, especially when your camera itself has limitations, such as a limited field of view. Our method deploys state-of-the-art techniques for constructing initial 3D models from photo collections. The same techniques are then used to register personal photos to these models, allowing us to augment
personal 2D images with 3D information. This strong available scene prior allows us to address a number of traditionally challenging image enhancement techniques, and achieve high quality results using simple and robust algorithms. Specifically, we demonstrate automatic foreground segmentation, mono-to-stereo conversion, field of view expansion, photometric enhancement, and additionally automatic annotation with geo-location and tags. Our method clearly demonstrates some possible benefits of employing the rich information contained in on-line photo databases to efficiently enhance and augment one’s own personal photos.

While some of these effects have been demonstrated or mentioned previously in different contexts, our system is the first of using information from IPCs to solve these problems for personal photos. Our main observation is that with the abundance of variations in viewpoint and illumination in IPCs, quite often high-quality results can be obtained with relatively simple methods. One main challenge we have to overcome is the limited completeness and accuracy found in state-of-the-art reconstruction methods due to occlusion and noise in the IPCs. An additional challenge when enhancing a personal photo is that we typically only have one view of the foreground in the personal photo. Therefore, its geometry can not be reconstructed. To overcome these limitations we propose novel methods to estimate or interpolate the missing depth values.

With the set of techniques we have adopted and developed in this paper, we envision a system where a user can upload his/her trip photos, taken with a regular camera, into a photo-editing system. The system will query previously reconstructed 3D models (from IPCs) and register each photo with the appropriate data set. After
registration, geo-location as well as tags can be obtained, and a foreground and back-
ground segmentation can be automatically computed. Then, users can interactively
enhance their trip photos in various ways. With more and more photos being stored
online, we predict that such a system will with time become more and more robust to
use, require less human interaction, and be able to accomplish an increasing number
of applications over a large range of locations.

The third contribution of this dissertation is an innovative solution to a challenging
3D modeling task, in particular 3D flower petal modeling. Plants modeling is one of
the most difficult tasks in computer vision and graphics community because of their
complex geometry and appearance. Flower, as the most distinctive part of a plant,
has fine structures and wide variations, which makes reconstructing their 3D models
a challenging task. Existing 3D modeling techniques for plants and vegetation are
usually designed for large scale structures, such as trees, foliage, or based on pure
synthesis given some pre-defined rules and templates. The biggest challenge for flower
modeling is occlusion. The tight formation of flower petals make segmentation and
3D reconstruction a very challenging task. In order to make this modeling problem
tractable, we develop a unique pipeline that incorporate domain-specific knowledge.
More specifically, the shape space of petals (the most dominant components of a
flower) can be learned from individually scanned petals and their relative spatial
layout can be known a priori from botany study.

To our knowledge, our system is the first to focus on flower modeling, petals in
particular, from 3D point cloud. The key contributions of our work can be summa-
ized as: 1) a novel petal fitting algorithm that is robust to significant occlusions;
2) a robust scheme for flower petal segmentation, by extending a two-region level-set formulation to multiple regions; 3) a scale-invariant morphable petal shape model which can handle wider shape variations within a species, or even across species.

It should be emphasized that our reconstruction pipeline generates a *parametric model*, which is particularly suited for measurement, editing, and animation. For example, one could easily apply a geometric morphing between two models, or make global changes to the shapes by varying shape parameters. While these are not explored in the context of this paper, we believe our approach will enable more research in the modeling and animation of an intrinsic class of objects, flowers, with applications in botany, entertainment, and visual simulations.

1.2 Guideline for Reading

This dissertation is organized as following. In chapter 2 we review existing related work on automatic/semi-automatic stereo matching, semantic segmentation, image enhancement and 3D plant modeling that are most relevant to the algorithms proposed in this dissertation. Chapters 3, 4, 5, and 6 contain the core material of this dissertation. Chapter 3 demonstrates a stereo video matching system that allows user interaction to obtain high quality, dense disparity maps on key frames and then intelligently propagates the user input and key frame disparities to automatically produce high quality disparity maps on intermediate frames. Chapter 4 addresses the problem of semantic segmentation of urban video sequence using dense depth maps. Experimental results from various data sets show that our method outperforms traditional 2D appearance based methods, or sparse 3D based method. In Chapter 5
we present our algorithms for personal photo enhancements using internet photo collections. We demonstrate four types of challenging enhancements to the personal photograph on seven different tourist landmarks from man-made architecture to natural scene. Chapter 6 describes a framework for data-driven high quality 3D flower modeling with Botany priors. In the experiment section we validate and evaluate our method using several flower datasets with severe occlusions and large shape/size variations. Finally, in Chapter 7 we conclude the dissertation with discussions on possible directions for future developments. In this dissertation, Chapters 3 is a joint work with Brian Price, Scott Cohen and Ruigang Yang, first presented in 3DV 2013. Chapter 4 describes a joint work with Liang Wang and Ruigang Yang, first presented in ECCV 2010. Chapter 5 is a joint work with Jizhou Gao, Oliver Wang, Pierre Georgel, Ruigang Yang, James Davis, Jan-Michael Frahm and Marc Pollefeys, first presented in IEEE TVCG 2013. Chapter 6 describes a joint work with Mao Ye, Bo Fu and Ruigang Yang, first presented in IEEE CVPR 2014.

---

1 3rd Joint 3DIM/3DPVT Conference
2 European Conference on Computer Vision
3 IEEE Transactions on Visualization and Computer Graphics
4 IEEE Computer Vision and Pattern Recognition Conference
Chapter 2 Related Work

The problems of semantic segmentation, image enhancement, 3D modeling and visualization have received tremendous interest in computer vision and graphics communities. And multiview stereo based depth acquisition is an important method in my work for solving those problems. We will briefly summarize existing works in each of the three research areas.

2.1 Automatic/Semi-automatic stereo matching

In terms of automatic stereo matching, there are quite a few algorithms that aim to explore the spatial-temporal redundancy to improve the task of stereo matching, optical flow estimation, or both. Some prior works use object motion/3D motion estimation over temporal sequences for robust stereo matching [5–7]. [8] uses the concept of scene flow to recover a temporally consistent 3D model and its motion. Zhang et al. extends traditional stereo matching windows in the temporal domain to account for object motion [9]. Various formulations have been developed to estimate both scene geometry and object motion in a unified process (e.g., [10][13]). More recently, Sizintssev and Wildes developed a unique space-time stereo algorithm that introduces the concept of spatiotemporal quadric element (stequel) as the matching primitive [2].

All of these above methods rely on the data contained in a local window in the space-time image volume for correspondence matching, where typical problems such
as textureless areas or non-Lambertian objects still exist. To address this, additional constraints have been introduced, for example changing the illumination conditions [14] or assuming a prior 3D model [15]–[18], or leveraging segmentation [19]–[20].

In this paper we present an alternative approach to resolve the ambiguity: user interaction coupled with a novel propagation method to automatically fix the problems for the majority of frames that are not marked. The use of our interactive tools in fact assigns meaningful labeling information for objects in the scene, and therefore the propagation can be done on a per-segment level, effectively handling large textureless or specular regions.

There have been existing works in consistent propagation of user edits throughout video [21] and semi-automatic generation of depth maps for stereoscopic video. [22]–[24] develop systems where users can paint depth onto a 2D image via sparse scribbles on key frames and propagate them into per-pixel depth maps through the whole image or video sequence. However, all of those systems only propagate a user’s empirically specified depth/relative depth values. In our work, we propagate interactively corrected disparity values from stereo matching, which represents more accurate 3D geometry of the scenes. Another interactive 2D to 3D conversion system [25] uses sparse structure from motion (SFM) and user supplied data for depth computing on a per-segment level, and propagates depths/segments using an MRF model across the temporal domain. Our proposed method propagates dense depth/segment information using a combination of spatial-temporal cost aggregation and tracking based disparity propagation scheme on a per-segment level.

Liu et al. [26] developed a system to annotate videos for ground-truth optical flow.
Our system can also be used in a similar way to generate ground-truth disparity maps semi-automatically, in particular for objects with rigid motion. Different from [26], we develop a richer set of tools that are tailored specifically for stereo pairs. Also we developed a unified MRF formulation to incorporate user input and propagation information.

2.2 Semantic segmentation and scene understanding

Recently, many efforts have been made to achieve accurate semantic segmentation and classification. Traditional approaches employ 2D appearance information, such as color, texture, shape [27, 29] and have achieved impressive results. However, a drawback of appearance-based features is that they may change dramatically under different imaging conditions. For example, in day time and night, summer and winter, a scene may have different appearances. With recent advances in 3D imaging, 3D structure information has been exploited for semantic segmentation and recognition [1, 30]. Specifically, when the input is a video sequence rather than a still image, the available motion-based cues contain a large amount of information that can be used for segmentation and recognition. They are invariant to appearance changes. In [28], Liu et al. proposed a novel nonparametric approach for scene parsing using dense scene alignment. Their method is based on the alignment of testing image and its best matching image in the database by SIFT flow. However their formulation requires a large amount of training data of around 2700 fully annotated images [31].

The approaches most related to ours are [1] and [30]. Our method is inspired by the work of [1] with the following distinctions: First, we propose to use dense (per-
depth map information for street scene segmentation, while in [1] they proposed to utilize sparse structure from motion point clouds; and in [30] they combined structure from motion and appearance descriptors together. Second, although both of the two previous approaches have achieved impressive results, some of the features, such as height above camera, is defined as the relative height between object and camera, therefore dependent on camera pose. Note there are two types of common camera configuration: front-view camera (as used in [1]) and side-view camera (as used in [30]). Training using one of the datasets and testing on the other is likely to lead to fail in both approaches. In our approach, because we only use per-pixel depth information without any dependence on camera pose or appearance, our segmentation algorithm can be easily formulated in a view-independent fashion. Applying our approach we can get satisfactory segmentation results from a video sequence using training data captured under a different configuration from testing data. Finally, while it is shown in [1] that motion-derived information leads to results comparable to existing state-of-the-art appearance based techniques. We demonstrate in this paper that semantic segmentation using only dense depth information outperforms both sparse structure from motion based method and appearance based method, or even the combination of sparse depth with appearance.

2.3 Internet Photo-based reconstruction and image enhancement

There are three main fields of related work: Internet photo-based reconstruction, image enhancement, and image segmentation.
**Internet photo-based reconstruction**  With the prevalence of consumer cameras and large scale on-line photograph storage sites, 3D modeling from IPCs has become a hot topic in recent years [32–35]. Snavely et al. [32] presented pioneering work using photographs from IPCs to compute a 3D model reconstruction and recovered camera poses. Furukawa and Ponce [34] presented efficient clustering and filtering algorithms for parallel reconstruction that enforced inter-cluster consistency constraints over the entire reconstruction. Subsequently, Agarwal et al. [33] and Frahm et al. [35] advanced the state-of-the-art of city scale reconstruction from IPCs, with both improved geometric accuracy and computational performance. Our work adds onto these recent advancements to better perform incremental updates to these models, creating new sources of prior information for use in personal photo enhancement and augmentation.

**Image enhancement**  There have been several techniques for image editing using large quantities of images downloaded from the web, such as colorizing gray-scale images [36, 37], enhancing CG images [38], and enhancing face images using good example prior photos of the same person [39]. Most related to our work is image completion [40–43]. Although impressive results are presented, these completion methods focus only on image inpainting tasks, while our field of view expansion task is more general. Additional differences exist, for example, Hays and Efros [40] used semantically similar images for the completion task, but did not attempt to recreate the original scene. Whyte et al. [41] used photos of the same scene, but only applied homography corrections for the geometric registration of images. Our work on the other
hand, combines 3D geometry information with homographies for more accurate image registration. Garg et al. [42] established a theoretical upper bound on the number of basis images to model real-world scenes and demonstrated some related applications including occluder removal and view expansion. However, their approach is limited when there are many large random and different foreground objects appearing in the images, requires manually segmenting a large set of images to learn the appearance bases, and is only able to output pure background landmark images with no foreground object appears. Instead, we introduce a novel use of content-aware scaling for challenging cases where there are many foreground occluders present. Our method is able to automatically segment foreground objects that exist in many personal photos, and create high quality field of view expansions in more general situations. [44] proposes a technique for intrinsic image extraction from photo collections and therefore can be used for lighting transfer between images. However, it cannot handle our saturation removal problems, since intrinsic image cannot be accurately extracted for saturated parts. Our proposed methods can solve the problem by replacing the saturated parts with content from properly exposed photographs in IPCs.

Our stereoscopic 3D creation application is inspired by work in the area of virtual view synthesis. View synthesis from multi-view data is a well established area. Most work involves computing depth from multi-view input, and using a depth-image based rendering (DIBR) model to create novel views [45,46]. While our depth information is computed from multi-view data, it is all computed a-priori, and mapped to a single query image, which is used for view synthesis. In addition, so as to leverage sparse data, and avoid disocclusion problems, we use a robust image-domain warping
method originally presented for artist-driven 2D to 3D conversion [47].

There are also works on estimating geographic information from images. [48] computes location distributions by low-level image matching to a geo-referenced database. [49] uses some travel priors to develop the chronological order of the images to find the location of images. Our geo-tagging method does not need such priors, and can obtain more precise geo-tags instead of geo-location distributions.

There has also been related work in the area of 3D model-based photo enhancement of landscapes and cityscapes [50]. Their work augments digital terrain and urban models with user interaction to register images to the 3D model, while we can achieve fully automatic 3D model selection and image registration. More significantly, we propose different enhancement applications from their work and our enhancement applications benefit not only from the reconstructed 3D models, but also from photographic appearance and other information that can be gained from large scale IPCs.

**Image foreground segmentation**  Interactive image segmentation brings a user’s prior knowledge of the location, size, color, and depth boundaries to segment a target object from an image, for example via a user-provided bounding box [51, 52] and strokes [53-54]. However, even simple labeling tasks such as dragging a bounding box may still be daunting when dealing with lots of images. We leverage the opportunity that IPCs registered to the same model allow, enabling us to measure color consistency between the personal photo and IPC and filter out foreground and background color seeds, obtaining a high quality segmentation.
2.4 3D Flower Modeling

Due to their importance in the real world, there are many approaches for modeling plants. They can be roughly divided into two categories: rule-based modeling and data-driven modeling. Rule-based methods use compact rules and grammars for building models of plants. As a prime work, a series of approaches based on the idea of L-system were developed \[55\, 59\]. The modeling of plant organs, such as leaves and petals, is a much less studied problem. Fowler and colleagues \[60\] developed a collision-based model for the spiral phyllotaxis effect, where plant organs are arranged in spiral patterns. Mundermann and collaborators \[61\] used leaf silhouettes to estimate leaf skeleton and further build leaf shape models. Fuhrer and colleagues \[62\] studied how to model and render small hairs on plants. Reunions and collaborators \[63\] developed procedural algorithms to model a number of leaf venation patterns. A related work in flower modeling is an interactive system by Ijiri and collaborators \[64\]. It has a graphical user interface for users to sketch flower models based on botanical constraints. However those work summarized above are purely rule-based, for which the realism and accuracy depend on the understanding of flower development and the effort of the modelers.

Recently with the proliferation of digital cameras and 3D scanning devices, there have been a number of data-driven approaches developed specific for tree, small plants, or foliage \[4\, 65\, 69\]. Typically major tree branches are detected or interactively traced from 3D point clouds or images. Leaves are synthesized based on separate scanning, some heuristics, or mesh-fitting, so that the final model is visually
similar to the input data. Approaches in this category aim to faithfully reconstruct the 3D model of plants based on the input. They usually focus on plants with a large number of leaves and the general structure of the whole plant. From algorithm perspective, the most related works are from Quan [65] and Bradley [4]. Quan et al. use similar modeling procedures to ours, composed of an interactive leaf segmentation and template based model fitting. Their approach requires multi-view data in which the entire plant is captured, while in our case we only use data from a single view because multi-view data do not provide significant more converges due to the tight formation of flower petals. Both [65] and [4] use an exemplar leaf mesh to fit to the dense point clouds non-rigidly. [4] further learns a statistical model for continuing fitting other leaves, as well as for leaf synthesis when occlusions are too big. We instead require a shape database for flower petals to handle the significant occlusions in flowers.

After a survey of existing plant modeling techniques, we note that flowers, despite being the most significant focus of study for identification, are the least frequently studied, probably due to its complex structure and significant self-occlusions. Our method uses both a data-driven approach and knowledge in botany to handle these challenges.
Chapter 3 High-quality Stereo Image/Video Matching via User Interaction

Stereo matching is one of the fundamental problems in computer vision. While much progress has been made over the last few decades of research, as demonstrated recently in [70], even the state of the art algorithms are quite fragile in practice. For example, large textureless areas cause ambiguity in matching, and objects with specular reflectance properties violate basic assumptions in matching. These situations, which are common in our everyday environment, all lead to erroneous disparity maps. In this paper we introduce a set of interactive tools to correct disparity maps generated by automatic approaches.

Our desire to allow human interaction is also motivated in part by the recent proliferation of consumer 3D displays and stereoscopic cameras. Similar to 2D image/video editing, there is a great need for new tools for stereoscopic image/video editing. Correcting disparity is a key element to allow editing with the added dimension. These new applications differ from the traditional stereo applications such as robotics in that user interaction is usually acceptable.

When the input is a stereoscopic video, manually tweaking each frame becomes too tedious. A better way is to edit on key frames and propagate the corrected disparity maps over time. While using optical flow may be the first choice, in practice it is doomed to fail. Stereo matching is just a restricted version of optical flow. With the same scene object, it is very unlikely that the 2D optical flow estimation will succeed
in places where the 1D stereo matching has failed (and therefore requires correction).

With our proposed editing tools, a user’s editing in the key frames not only corrects the disparity map but also conveys important information about how pixels are grouped together. By relying on this segmentation information, we can propagate disparity maps on a per-segment basis instead of a per-pixel basis. The per-segment transfer is modeled as a 3D rigid transformation. A key insight is that we can estimate such a 3D transformation from sparse feature points reliably tracked across time and views. This allows us to accurately propagate dense disparity values across time from large regions that contain only a handful of trackable feature points, such as low-texture regions or specular objects.

In terms of technical contributions, we formulated the user interaction and disparity map propagation under a unified Markov Random Field (MRF) framework. The user’s input and/or the propagated disparity maps affect the data or smoothness costs in a global MRF objective function, and improved disparity maps are obtained by re-optimizing the modified objective function.

With our novel and practical system, disparity errors in stereo pairs can be easily corrected, yielding high-quality results not possible with fully automatic methods. In particular, our novel disparity propagation approach significantly reduces the amount of manual labeling required for correcting stereoscopic videos. We believe our system is an important step towards widespread consumer stereo image/video editing, for which the final quality is paramount.
3.1 Overview and Problem Formulation

Given a stereo video \( I_t = \{I_l, I_r\}_t \), where \( I_l \) and \( I_r \) are the left and right frames, the goal of our system is to compute the dense disparity map \( D_t \) of each reference frame, say \( I_l \) at each time \( t \). For notation clarity, we will omit the subscript \( t \) when describing our stereo model.

In our stereo matching model, in addition to the input images, there exists a propagated disparity map on the reference view, denoted by \( \hat{D} \), from some key frames where the disparities are of high quality and confidence. We formulate our stereo model as a MAP-MRF problem. Under the assumption that \( \hat{D} \) is independent of the image formation process of stereo image pair \( I \), the posterior probability over \( D \) given \( I \) and \( \hat{D} \) can be written as \( P(D | I, \hat{D}) \propto P(I | D)P(\hat{D} | D)P(D) \). Since maximizing the posterior is equivalent to minimizing its negative log likelihood, our objective is to find a disparity map \( D \) to minimize the following global energy function:

\[
E(D) = -\ln(P(I | D)) - \ln(P(D)) - \ln(P(\hat{D} | D))
\]

\[= E_{data}(D) + E_{smooth}(D) + E_{prop}(D).\]  

(3.1)

We define the data term as the weighted sum of absolute difference of color and gradient between corresponding pixels given an assigned disparity value \( d_p \) for pixel \( p \):

\[
E_{data}(D) = \sum_{p \in I} \Phi(p, d_p)
\]

\[
\Phi(p, d_p) = \alpha\|I_{l,p} - I_{r,p-d_p}\| + (1 - \alpha)\|\nabla_x I_{l,p} - \nabla_x I_{r,p-d_p}\|
\]

(3.2)

For smoothness cost, we use the linear truncated model defined upon a 4-connected neighborhood system \( N_4 \):

\[
E_{smooth}(D) = \sum_{p \in I} \sum_{q \in N_4(p)} W(d_p, d_q)
\]

\[
W(d_p, d_q) = \rho_{pq} \cdot \min(|d_p - d_q|, \tau)
\]

(3.3)
where $\rho_{pq}$ is the rate of increase in discontinuity cost and $\tau$ controls the limit of cost. The propagation term plays a regulation role and will be described in section [3.3.3].

While we can use any stereo algorithm to optimize the cost function defined in eq. (3.1), our current implementation uses hierarchical belief propagation (HBP) [71]. Given a new sequence, our system initially calculates $D_t$ for each frame. Then the user can refine $D_t$ using the interactive tools described in the next section, and finally the entire sequence will be optimized again using updated values for eq. (3.1).

### 3.2 Interactive Stereo Correction

Here we introduce the various tools we have developed to allow easy and effective disparity map correction. Figure [3.1](image) shows a screenshot of our interface. A fundamental tool through all our editing is a stroke-based graph-cut selection tool [72] for users to quickly select a certain object in images. The user interaction in our framework is incorporated as constraints modifying $E_{data}(D)$ and $E_{smooth}(D)$, as opposed to being applied as a post-process.

#### 3.2.1 Model fitting tool

The model fitting tool is used to encourage a certain object to be close to a specific geometric model. In this paper we specially handle two types of shapes, plane and sphere, although our method extends to any parametrically-defined model. After selecting a certain object, users can specify the geometric model for that object. Sift feature detection and matching is used to establish 2D correspondences between left and right images, which are thereafter triangulated to obtain 3D point positions for
Figure 3.1: User interface of our interactive stereo correction tools.

a RANSAC based model fitting. Predicted disparity values of the object can be obtained by projecting the fitted model to image plane.

Let $d_p^m$ be the predicted disparity value of pixel $p$ from model fitting, which should be encouraged in disparity assignment to $p$. Therefore, we change the data cost of assigning to $p$ disparities around $d_p^m$ as:

$$
\Phi(p, d_p) = \Phi(p, d_p) - \rho(d_p)
$$

(3.4)

$$
\rho(d_p) = \begin{cases} 
\delta(1 - \frac{|d_p - d_p^m|}{\Delta d}) & |d_p - d_p^m| \leq \Delta d \\
0 & \text{otherwise}
\end{cases}
$$

(3.5)

where $\delta$ is a large positive number. As we cannot guarantee that the predicted disparity values are all perfectly accurate, we use a linear model on cost change to allow preference in a disparity range$(d_p^m - \Delta d, d_p^m + \Delta d)$. This scheme can also

23
propagate a range of potential good disparities to other frames in spatiotemporal cost aggregation stage (Sec [3.3.1]). Allowing for a range of disparities is more useful in this sense because the disparities of corresponding objects in other frames might change due to camera or object motion.

3.2.2 Object correspondence tool

Often stereo algorithms will generate disparities that match a part of an object in one image to a different object in the other. To address this, the object correspondence tool allows users to specify object correspondence by selecting the object in the left and right views. This is incorporated as constraints that bias the disparity assignments to map the object in one view to the corresponding object in the other view. The tool especially works well on thin structures. Let $O_l$ and $O_r$ be the selected corresponding objects in the left and right views. We change the data cost of assigning disparity $d_p$ to pixel $p \in O_l$ or $p \in O_r$ as:

$$\Phi(p,d_p) = \Phi(p,d_p) + \delta \text{ if } p \in O_l, p - d_p \notin O_r$$
$$\Phi(p,d_p) = \Phi(p,d_p) + \delta \text{ if } p \in O_r, p + d_p \notin O_l$$

(3.6)

where $\delta$ is a large positive number.

3.2.3 Alignment tool

The alignment tool allows users to indicate the rough disparities of large regions at a single time. By overlapping and blending the left and right views then shifting them relative to one another, users can easily see which regions are well aligned. Painting on these regions allows the user to mark the approximate disparity of the entire region very intuitively. Let $d_p^a$ be the predicted disparity value of pixel $p$ determined
by the alignment. The data cost for $p$ is the same as in Equations 3.4 and 3.5 but substituting $d^a_p$ for $d^m_p$.

### 3.2.4 Smoothness brush tool

Stereo matching algorithms may create erroneous edges in the disparity map. This often happens when there is an image edge in a region where the true disparities vary smoothly. Users can correct this using the smoothness brush to indicate regions in the image where the object surface is smooth. Different from the tools described before, this tool incorporates the interaction constraints by changing the rate of increase in discontinuity cost $\rho_{pq}$ in the energy function in eq. (3.3). The user may adjust $\rho_{pq}$ for a given region. Let $\rho_{old}$ be the original rate of increase in discontinuity cost and $\rho_{new}$ be the adjusted value for the user-specified region $O$. $\rho_{pq}$ is changed as:

$$
\rho_{pq} = \begin{cases} 
\rho_{new} & p, q \in O, q \in N_4(p) \\
\rho_{old} & \text{otherwise}
\end{cases}
$$

### 3.2.5 Discontinuity brush tool

Stereo matching algorithms, especially those based on global optimization like belief propagation, have the tendency to smooth over depth discontinuities. To preserve these discontinuities, we design a discontinuity brush tool to enforce discontinuity along certain object boundaries. Users mark parts of the object boundary generated from selection tool, where depth discontinuities occur and can specify a minimum disparity difference along the discontinuity boundary.

Let $\partial O$ be the discontinuity boundary of a selected object $O$. Denote $\Delta d$ as the interactively adjusted minimum disparity difference between two neighbor pixels
(p, q), where p ∈ ∂O, q ̸∈ O. The discontinuity constraints are incorporated by changing the smoothness cost as:

For p ∈ ∂O, q ̸∈ O, q ∈ N₄(p):

\[
\mathcal{W}(d_p, d_q) = \begin{cases} 
+\infty & |d_p - d_q| < \Delta d \\
\kappa & \text{otherwise}
\end{cases}
\]  

(3.8)

where κ is a constant.

3.3 Temporal Disparity Propagation and Fusion

In this section, we describe our pipeline for propagating refined disparity maps in key frames to the entire sequence. It consists of three major steps: 1) propagate the corrected cost volume in the temporal domain by applying spatial-temporal cost volume aggregation; 2) propagate the corrected disparity maps in the temporal domain by using sparse feature tracking and disparity warping in 3D space; 3) re-optimize the disparity map. We now discuss each of these steps in detail.

3.3.1 Spatio-temporal Cost Volume Aggregation

Inspired by [73], we propagate the corrected cost volumes of key frames to neighbor frames by applying a spatio-temporal cost volume aggregation. Firstly, a three-dimensional cost volume \( C^d \) is constructed for each disparity \( d \). Elements in the cost volume are the costs of assigning disparity \( d \) to each voxel \( i = (x, y, t) \), where \( (x, y) \) are the image coordinates and \( t \) the temporal coordinate of voxel \( i \). After the cost volume for each disparity \( d \) is established, we filter the cost volume. More precisely, the filtered cost value of assigning disparity \( d \) to each voxel \( i \) is a weighted average of all voxels in cost volume \( C^d \):
\[ C'_{i,d} = \sum_j W_{i,j}(I^s)C_{j,d} \] (3.9)

where \( W_{i,j} \) is the filter weight of a neighbor voxel \( j \) to reference voxel \( i \). It depends on input image sequence \( I^s \). Here we follow the idea of [74] and use the weights from the guided image filter [75]. The reason we choose the guided image filter lies in its edge-preserving properties, runtime independent of the filter size, as well as the simplicity of extending it to temporal domain. The weight \( W_{i,j} \) is given by:

\[
W_{i,j} = \frac{1}{|\omega|^2} \sum_{k:(i,j) \in \omega_k} (1 + (I_i^s - \mu_k)^T(\Sigma_k + \epsilon U)^{-1}(I_j^s - \mu_k))
\] (3.10)

Here, \( I_i \) and \( I_j \) are \( 3 \times 1 \) (color) vectors and \( \mu_k \) and \( \Sigma_k \) are mean vector and covariance matrix computed over the a spatio-temporal window \( \omega_k \) with size \( \omega_x \times \omega_y \times \omega_t \), centered at voxel \( k \). \(|\omega|\) is the total number of voxels in \( \omega_k \) and \( \epsilon \) is a smoothness parameter. \( U \) is the \( 3 \times 3 \) identity matrix.

To maximize the influence from the user-corrected cost volume, we filter the reference frame cost volume only with those frames’ costs that have been changed. The filtered cost volume will be used as the new data cost \( E_{\text{data}} \) in eq. 3.1.

### 3.3.2 Disparity propagation via tracking and 3D transformation

We especially want to deal with challenging scenes with big textureless regions or specular objects. While they can be corrected manually in key frames, propagating their disparity values over time is challenging since they are difficult to track. So instead of trying to propagating them pixel-by-pixel, we will use the group information implicitly provided by the user to propagate them on a per-segment basis. The hope is that there are still a few sparse feature points that can be tracked.
Let \((I^t_l, I^t_r)\) and \((I^{t+1}_l, I^{t+1}_r)\) be the stereo image pairs at time \(t\) and \(t+1\). First, we use SIFT feature detection and matching to find 4-tuple spatial-temporal correspondences among \((I^t_l, I^t_r, I^{t+1}_l, I^{t+1}_r)\), where each 4-tuple correspondence, \((p^{l,t}_i, p^{r,t}_i, p^{l,t+1}_i, p^{r,t+1}_i)\), contains one point from each image. For simplicity purpose, we demonstrate our algorithm using the left view. After establishing 4-tuple correspondence, we compute the 3D position \(P^{l,t}_i\) of the 2D pixel \(p^{l,t}_i\) and 3D position \(P^{l,t+1}_i\) of 2D pixel \(p^{l,t+1}_i\).

We then compute the 3D transformation between frame \(t\) and frame \(t+1\) on a per-segment basis, using the object segmentation generated from the interactive stereo correction stage. We also find that warping an object to a neighboring frame is more robust than warping the whole scene. Let \(P^t_{sparse}\) be the set of 3D positions of sparse features in one object under camera \(t\), and \(P^{t+1}_{sparse}\) be the set of corresponding 3D positions under camera \(t+1\). There is a relative transformation \(R\) and \(T\) between \(P^t_{sparse}\) and \(P^{t+1}_{sparse}\):

\[
P^{t+1}_{sparse} = RP^t_{sparse} + T \tag{3.11}
\]

We use a least squares solution of \(R\) and \(T\) to transform the dense 3D point clouds \(P^t_{dense}\) in the object under camera \(t\), to 3D coordinates \(P^{t+1}_{dense}\) under camera \(t+1\).

\[
P^{t+1}_{dense} = RP^t_{dense} + T \tag{3.12}
\]

By projecting \(P^{t+1}_{dense}\) to frame \(t+1\), we can obtain the corresponding positions and disparity values of the object in frame \(t+1\). To compensate for inaccurate propagation caused by tracking or user’s correction, we use color consistency across both time and views to measure the confidence of propagated disparity. Let \(m^t_i\) be a point in the left frame \(t\) warped to \(m^{t+1}_i\) in left frame \(t+1\) with predicted disparity value \(d\). The
confidence value \( C(m_t^{t+1}, d) \) of \( m_t^{t+1} \) with disparity \( d \) is computed using normalized cross correlation (NCC) as:

\[
C(m_t^{t+1}, d) = NCC(N_l(m_t^t), N_l(m_t^{t+1})) + NCC(N_r(m_t^{t+1}), N_r(m_t^{t+1} - d))
\]  

(3.13)

where \( N_l(p) \) or \( N_r(p) \) is a patch centered at \( p \) in left or right images. In (3.13), the first NCC measures color consistency of the left frames at \( t \) and \( t+1 \) under the 3D transformation, and the second NCC between left and right frames at \( t+1 \) under disparity \( d \). For regions not covered after propagation, the confidence value is assigned as 0. Figure 3.2 shows our disparity propagation scheme based on sparse tracking and 3D transformation.

Figure 3.2: Overview of sparse feature and 3D transformation based disparity propagation.

After all the selected objects in frame \( t \) are transformed to frame \( t + 1 \), a predicted disparity map of frame \( t + 1 \) is obtained along with its confidence map. A bilateral interpolation is applied to fill in holes within objects in disparity maps caused by
transformation to new frame. The predicted disparity map will be used as regularization priors in the MRF stereo formulation when computing the disparity map of frame $t+1$. The propagation continues from frame $t+1$ to the next frame. Figure 3.3 shows an example of our high-quality disparity propagation, in which both the camera and the person in the scene have rotation movements.

Figure 3.3: Example of disparity propagation on Office 2 dataset. From left to right: key frame 1(frame 10), key frame disparity correction(frame 10), intermediate frame(frame 20), propagated disparity map(frame 20).

3.3.3 Fusion of Aggregated Cost and Propagated Disparity

To deal with inaccuracy in tracking and non-rigid deformation over time, we treat the propagated disparity map $\hat{D}$ as a soft constraint. The propagation regularization term $E_{prop}(D)$ is defined as:

$$E_{prop}(D) = \lambda_p \sum_{p \in I} \Psi(d_p, \hat{d}_p) C(\hat{d}_p)$$  \hspace{1cm} (3.14)

where the function $\Psi(d_p, \hat{d}_p)$ penalizes the disparity assignment $d_p$ when deviating from the propagated disparities $\hat{d}_p$. $\lambda_p$ is a coefficient that controls the weight of the propagation energy. $C(\hat{d}_p)$ is the associated confidence map of the propagation. The penalty function $\Psi(x, y)$ is defined using the Total Variance model [76] as:

$$\Psi(x, y) = -\ln((1 - \eta) \exp(\frac{-|x - y|}{\gamma}) + \eta).$$  \hspace{1cm} (3.15)
where $\gamma$ and $\eta$ control the sharpness and upper-bound of the robust function. Since the propagated disparity priors and aggregated data cost both contain strong information from good disparity maps, we handle the conflict between them in a smooth way by making $\gamma$ a function of propagated disparity value $\hat{d}_p$ and a winner-takes-all (WTA) solution $\tilde{d}_p$ of pixel $p$ from aggregated data cost:

$$\gamma(\hat{d}_p, \tilde{d}_p) = \gamma_{\text{min}}(1 + |\hat{d}_p - \tilde{d}_p|)$$ (3.16)

where $\gamma_{\text{min}}$ is the min value of $\gamma(\hat{d}_p, \tilde{d}_p)$ in the case of $\hat{d}_p = \tilde{d}_p$. The larger the difference between $\hat{d}_p$ and $\tilde{d}_p$ is, the larger $\gamma(\hat{d}_p, \tilde{d}_p)$ is, which makes $\Psi$ smoother. This makes sense because large deviation from the aggregated data cost probably indicates a non-perfect propagation. In this case we do not want $\Psi$ to be too sharp around $\hat{d}_p$, and vice versa.

With both forward and backward propagation, each frame has two $\hat{D}$ maps. $E_{\text{prop}}(D)$ is calculated as the weighted sum of the two, with weights inversely proportionally to how far the reference frame is from the two key frames. With all terms in eq. (3.1) updated, our system then refines the disparity map via another round of HBP.

### 3.4 Evaluation

We have implemented our system and here we show the results on three different datasets. The first dataset is composed of two laboratory scenes from York University [2]. The second set contains two scenes that we captured in indoor office environments. The third dataset from [70] is captured in outdoor environments. We use the
following parameter settings to generate all results: \( \{\alpha, \rho_{pq}, \tau, \delta, \kappa, \omega_z, \omega_y, \omega_t, \epsilon, \lambda_p, \gamma_{\text{min}}, \eta\} = \{0.15, 1, 10, 10^{11}, 100, 9, 9, 4, 0.0001, 10^9, 2, 0.005\} \). For the datasets shown in our paper, only the first and last frames are corrected with our interaction tools. It takes about 5-10 minutes to refine a disparity map depending on the complexity of the scene. The average processing time on computing disparity map from propagated information is around 10-20 seconds per frame.

### 3.4.1 York Stereo Dataset

The first dataset contains two laboratory scenes (\(\text{Lab1} 28\) frames, \(\text{Lab2} 40\) frames), which include planes slanted in depth, textureless/repetitive texture regions (central part of the scene), and complicated objects with non-trivial 3D boundaries and non-Lambertian materials (e.g., the teddy bear and gargoyle). The stereo camera makes horizontal and in-depth movements, and out-of-plane rotation.

In Figure 3.4 and Figure 3.5, the first three images of the second row in each figure show intermediate results that are progressively refined by adding more constraints to the optimization process. The first one is the disparity map from initial costs. The second uses aggregated data costs, which brings improvement, but still have inaccurate regions (noisy boundaries). The third image incorporates tracking based propagations to aggregated costs and generates high quality results. HBP [71] is used for optimization at each stage.

We also compare our results with two other methods. One is replacing our tracking based propagations by optical flow based propagation (2nd row, 4th column) using a top optical flow algorithm [77], the other is from a recent state-of-the-art spatial-
temporal stereo algorithm \cite{2} (2nd row, 5th column). With our semi-automatic refinement system, the results are visibly improved – which is of course expected since we require user interaction. Most noticeable improvements are in textureless regions or regions that can be fixed by our model fitting tools. Another advantage of our algorithm is improved preservation of discontinuity boundaries. This is because our selection tool can produce precise segmentation boundaries in key frames, and our tracking and 3D transformation based propagation can transfer the depth discontinuity well.

\subsection*{3.4.2 Office Dataset}

This dataset includes two indoor office scenes, both of which contain large textureless regions (white wall and table) and moving objects. The stereo camera makes horizontal rotation movements. Sequence \textit{Office 1} (1st row) has a length of 10 frames
and contains a moving object (rolling ball), and Sequence Office 2 (2nd row) has 20 frames and contains a rotating person sitting on a chair. As shown in Figure 3.6, we again make comparisons with two other methods, optical flow based propagation (5th column) and an automatic stereo algorithm [3] (6th column). Such comparisons again demonstrate the good performance of our method in dealing with textureless scenes, on which even the state-of-the-art stereo matching or optical flow algorithms would fail.

### 3.4.3 KITTI Vision Benchmark datasets

The last dataset is captured outdoors with a moving stereo camera on a car. Uncontrolled outdoor scenes are much more challenging for stereo matching. We show two sequences with specular objects and low-texture regions. As shown in Figure 3.7, in Seq0093 there are multiple specular surfaces (windows on left wall and back of right vehicle) in this scene. Automatic stereo algorithm fails in these regions because
Figure 3.6: Results of Office datasets. First row & Third row: From left to right, 1) key frames in blue (Office 1: frame 18, Office 2: frame 10); 2) user corrected disparity map (Office 1: frame 18, Office 2: frame 10); 3) reference image in red (Office 1: frame 23, Office 2: frame 20). Second row & Fourth row: 1) spatiotemporal cost aggregation+tracking and 3D transformation based propagation (Ours); 2) spatiotemporal cost aggregation+optical flow based propagation; 3) results from [3].

of wrong matching information. The ground as well as the side of the vehicle are textureless and neither optical flow or automatic stereo can succeed. But via sparse feature tracking, our algorithm can intelligently propagate the corrected disparity map. Note that because of large motion between neighbor frames in this sequence,
Figure 3.7: Results of KITTI Seq 0093 datasets. **First row**: from left to right, 1) key frame in blue (frame 153); 2) user corrected disparity map (frame 153). **Second row**: 1) reference image in red (frame 155); 2) result from initial cost. **Third row**: from left to right, result of 1) spatiotemporal cost aggregation; 2) spatiotemporal cost aggregation+tracking and 3D transformation based propagation (**Ours**). **Fourth row** 1) spatiotemporal cost aggregation+optical flow based propagation; 2) method from [3].

We choose temporally close key frame and reference frame to guarantee trackable sparse features, as well as demonstrating on specular objects which disappear in next frame.

Figure 3.8 shows the results of another outdoor scene Seq0046. Our algorithm still achieve much higher quality results, especially on low-texture regions (ground, building facade), where state-of-the-art automatic stereo matching and optical flow algorithms both fail.
3.5 Summary

In this chapter we present a system to semi-automatically produce disparity maps from a stereo sequence. Using state-of-the-art stereo algorithms, it first automatically estimates an initial disparity map for each frame. The user can then correct disparity maps in key frames using a set of novel tools. A key component of our tools is the ability to select a region of interest in one or both views for (i) fitting low order geometric models, (ii) specifying areas in the two views that match one another, (iii) marking depth discontinuities, and (iv) identifying areas where more smoothness in
the disparities is needed. The corrected disparity maps are propagated to the remaining frames using an object-based 3D transformation that is robust to mismatches. We treat user input and propagation results as soft constraints and optimize the final disparity map with an integrated MRF framework. We demonstrated our system with a number of challenging sequences. Future work includes finding a more intelligent way to deal with dis-occlusion regions, as well as non-rigid object propagation.
This chapter introduce the first problem we solve using depth information. We present a framework for semantic scene parsing and object recognition based on dense depth maps. Five view-independent 3D features that vary with object class are extracted from dense depth maps at a superpixel level for training a classifier using randomized decision forest technique. Our formulation integrates multiple features in a Markov Random Field (MRF) framework to segment and recognize different object classes in query street scene images. We evaluate our method both quantitatively and qualitatively on the challenging Cambridge-driving Labeled Video Database (CamVid). The result shows that only using dense depth information, we can achieve overall better accurate segmentation and recognition than that from sparse 3D features or appearance, or even the combination of sparse 3D features and appearance, advancing state-of-the-art performance. Furthermore, by aligning 3D dense depth based features into a unified coordinate frame, our algorithm can handle the special case of view changes between training and testing scenarios. Preliminary evaluation in cross training and testing shows promising results.

Figure 4.1: Overview of our framework
Figure 4.1 shows an overview of our scene-parsing pipeline. The proposed algorithm starts from generating dense depth maps by plane swiping method. An over-segmentation is applied to video sequences and 3D information is obtained using dense depth maps. Five discriminative 3D features are extracted from dense depth maps and combined together to build a randomized decision forest classifier to obtain accurate semantic scene segmentation. The primary contribution is that we demonstrate a scene parsing algorithm that uses only dense 3D depth information to outperform the combination of sparse 3D features and color. Moreover, we demonstrate that by transferring those 3D features to a common coordinate system that is independent on camera pose, view-independent semantic scene can be achieved. Training in one type of camera view and application in a changed camera view is now possible.

In summary, our dense depth based segmentation algorithm lends itself well for real-world applications in which the viewpoint, lighting and object textures are likely to be significantly different from those captured in the training database.

4.1 Depth Maps Recovery

Stereo reconstruction of dense depth maps from a video sequence has long been a research topic in computer vision. Recently, there have been great advances that are based on high-quality stereo matching algorithms and effective 3D modeling pipelines \[78,79\]. As depth recovery is not the primary focus of this work, we simply modify existing techniques to compute the scene depth information from video.

Given an input video sequence \{I^t\} captured at street level, we first employ the
SFM software released by Zhang et.al. \cite{80} to estimate camera parameters. Then, our stereo matching module takes as input camera poses and the corresponding frames from the video and produces a dense depth map $D_t$ for each frame. The stereo matching pipeline used in our paper consists of a depth initialization step followed by a depth refinement process.

For depth initialization, we model stereo matching as an energy minimization problem. The global energy function contains three terms, i.e., a data term, a smoothness term and a segmentation term. The data term measures how well the depth map agrees with the input images under the color consistency assumption. In this paper, we use the standard plane-sweep approach as described in \cite{81, 82} to compute data costs. In our implementation, the plane-sweep stereo is solved on 17 consecutive images where the middle one is the reference view. The smoothness term incorporates the assumption that the scene is piecewise smooth and penalizes assigning different depth values to neighboring pixels. We use the truncated linear model \cite{83} to define our smoothness cost. In order to better handle textureless areas, we incorporate the segmentation information into our MRF stereo framework as a soft constraint. We first segment each reference frame using mean-shift algorithm \cite{84}. Each color segment is treated as a nonfronto-parallel plane in 3D and a robust plane fitting method \cite{85} is applied to estimate the plane parameters. Similar to \cite{86}, the segmentation term is modeled to penalize depth assignment that departs from that given by the pixel’s corresponding plane parameter. For each frame $I_t$, we use belief propagation (BP) \cite{83} to estimate an initial depth map $\widetilde{D}_t$ by minimizing our energy function.
In depth initialization step, we compute the disparity map $\tilde{D}_t$ for each frame without considering the temporal consistency among depth maps. To address this issue, during the depth refinement step a multi-view fusion algorithm \cite{87} is applied to refine the depth maps returned by BP. Depth maps after refinement are smooth both spatially and temporally and contain less visual artifacts.

4.2 Semantic Segmentation from Dense Depth Maps

After recovering dense depth maps from videos, our algorithm starts by over-segment each image into homogeneous pixel clusters, i.e. superpixels, then extract feature vectors. These vectors are used for training and classifications. Raw classified output are combined via a pairwise Markov Random Field (MRF) for final segmentation. Details are presented below.

4.2.1 Image over-segmentation

Over-segmentation of image into superpixels is a common preprocessing step for image parsing algorithms. We choose to use over-segmentation as one of the preprocessing steps in semantic segmentation due to the following reasons: First, each superpixel in 2D images can be approximately viewed as a patch in 3D. Some features we employed in this work can be well defined over a patch, e.g., surface normal, surface planarity etc. Second, over-segmentation can increase the chances that the boundaries of different object classes are extracted. In this regard, pixel-wise classification may result in less consistent boundaries. Finally, using over-segmentation can reduce the computational complexity of the system, since by counting each superpixel as one
sample, the number of total samples are largely reduced as compared to pixel-wise training and testing.

We applied a geometric-flow based algorithm named “TurboPixels”\cite{turbo} to achieve dense image over-segmentation. This recent technique can produce superpixels with uniform size and shape, maintain connectivity and compactness, and preserve original image edges. The choice for superpixel size is a key issue. On one hand, using large superpixels may bring in the risk of a superpixel spanning across multiple semantic objects. On the other hand, a small superpixel may contain insufficient points to precisely define a good feature. In our experiment, we set the initial number of superpixel as 6800 for image of size 960 × 720, roughly 100 pixels per superpixel.

4.2.2 Features from dense depth map

For each superpixel, we extract five features in 3D space to train our classifier. These features are computed based on the 3D points within each superpixel. All five features are invariant of appearance changes. The five features are surface normal, height above ground, surface local planarity, surface neighboring planarity, and distance to camera path, which are denoted as $F_n^i, F_h^i, F_l^i, F_g^i, F_d^i$ for each superpixel $i$. Brostow et.al.\cite{brostow} defined some similar features based on sparse 3D points and projected the features from 3D point clouds to 2D image plane. The way we compute our 3D features is different from theirs, and dense depth maps allow us to estimate these features in a more principled way.

**Surface normal** ($F_n^i$): We compute surface normal $F_n^i$ of a 3D patch by fitting a least square plane to the set of 3D points within a superpixel. Similar method has
Figure 4.2: Visualization of different steps and features in our algorithm.

Height above ground ($F_h^i$): An object’s height information is normally fixed and invariant to change of driving direction, thus can be used as a good feature for classification. [1] used height above camera center as one of the structure features. However, this feature is not invariant to the car on which the camera is mounted, or, large camera motion. We instead use height above ground as our feature. Our algorithm requires a process to estimate ground plane parameters from depth map. In our implementation we use an iterative RANSAC method to estimate ground plane. At the beginning only 3D points whose normal are close to the up direction in camera coordinate system are used for samples. To avoid including points from sidewalk into plane fitting, after each RANSAC procedure we decrease the error threshold $T_h$ used by RANSAC by half and use inliers from previous iteration to fit a new plane. Here $T_h$ is simply a value that controls the point to plane distance. RANSAC will treat a point as an inlier if the distance from the point to plane is smaller than $T_h$. After
a few iterations (6-8 in our implementation) the algorithm terminates and the final plane parameters are treated as ground plane parameters. We find this method works fairly well for our data, where nearly half of the pixels in images are dominated by ground scene. For a superpixel $i$, height above ground $F^i_h$ is computed as the average distance to ground within the corresponding 3D patch.

**Surface local and neighboring planarities:** We in work define two types of surface planarity. One is the local planarity of a 3D patch ($F^i_l$), which corresponds to a superpixel in the image. The other is the neighboring planarity ($F^i_g$), that is, measuring the variance of a 3D patch orientation with respect to its neighboring patches. The local planarity is computed by using RANSAC based least square plane fitting and calculating the sum of square distances from points to the plane. The neighboring planarity is defined as the average difference of a 3D patch’s surface normal with respect to its neighbors’ surface normals. These features are useful for splitting planar and non planar objects, for example, building facades and plants.

**Distance to camera path ($F^i_d$):** Inspired by [1], we can take advantage of the distance to camera path to separate objects which are horizontally distanced from the camera. We compute the minimum distance from the centroid of the 3D patch to the camera path. The camera path is estimated by fitting a quadratic curve to the camera trajectory. This approach is more accurate and robust for computing objects’ distances to camera path, and works well on various scenarios.
4.2.3 Randomized Decision Forest

Randomized decision forest is a well-known machine learning technique that has been employed in many computer vision tasks [89]. We use this technique to train our classifier based on features derived from the depth maps. In our implementation, we choose the proportion of training data used at each split node to be 0.66. A total number of 80 random decision trees are built for training. We experimentally find these parameters work well for achieving satisfactory recognition rate. We also apply the idea in [27] to balance the number of classes used for training, thus achieve a better class average performance.

4.2.4 Graph-cut based optimization

We construct a pairwise Markov Random Field (MRF) for each image $I$ by building a graph $G = \langle V, E \rangle$, where each node $v_i \in V$ in the graph represents a superpixel and each edge $e_{ij} \in E$ denotes the neighboring relationship between superpixels. The labeling problem is equal to assign a label $l_i \in L$ to each node $v_i \in V$. The optimal assignment $L_{assign}$ can be achieved by minimizing the energy:

$$E(L_{assign}) = \sum_{v_i \in V} \psi_i(l_i) + \lambda \sum_{e_{ij} \in E} \phi_{ij}(l_i, l_j) \quad (4.1)$$

Data term $\psi_i(l_i)$ and smoothness term $\phi_{ij}(l_i, l_j)$ are defined in the following paragraphs. They are computed from the feature responses and the randomized decision forest based classifier. After the costs are computed, a graph-cut optimization [90] is applied to obtain the global optimal labeling configuration.
Data term

The feature responses $F_n^i, F_h^i, F_l^i, F_g^i, F_d^i$ of each superpixel $i$ are collected and applied rank normalization before passed to randomized decision forest for training. Specifically, given the samples for a feature $F$ for all the superpixels as $F^1, F^2, ..., F^n$, where $n$ is the number of superpixels, we first find the low-to-high order statistics $F^{(1)}, F^{(2)}, ..., F^{(n)}$ and then replace each image’s feature value by its corresponding normalized rank, as:

$$\tilde{F}_i = \frac{\text{rank}(x_i) - 1}{n - 1}$$ (4.2)

where $F_i$ is the feature value for the $i$’th sample. The procedure uniformly maps all the features to the range of $[0, 1]$. When there are multiple samples with the same feature value, they are assigned the average rank of that value. We applied this data processing approach based on the fact that there are some scale factors between the features’ measurements of different scenes. In addition, normalized rank is effective to compensate some inaccuracies induced by depth map generation and 3D features computation. After rank normalization, the feature responses of each superpixel are passed to randomized decision forest with corresponding ground-truth labels to build the classifier. When testing, the feature responses of each testing sample are passed to the classifier and a posterior probability distribution $P_i(l_i|F_n^i, F_h^i, F_l^i, F_g^i, F_d^i)$ which represents the probabilities the testing sample belongs to each category $l_i \in L$ is returned. We define the data term in MRF as:

$$\psi_i(l_i) = -\log P_i(l_i|F_n^i, F_h^i, F_l^i, F_g^i, F_d^i)$$ (4.3)
**Smoothness term**

For a superpixel $v_i$ and each of its neighbor superpixel $v_j$, the smoothness cost is defined as:

$$
\psi_{ij}(l_i, l_j) = \left[ l_i \neq l_j \right] \cdot \frac{1}{\delta \| c_i - c_j \|_2 + 1}
$$

(4.4)

where $\| c_i - c_j \|_2$ is the L2-Norm of RGB color difference between neighbor superpixels. The penalty is inversely proportional to the color difference between neighboring superpixels. The more similar the colors of two neighbor superpixels are, the less likely they belong to different categories. In our experiment, the value of $\lambda$ is set to be 1.9 and $\delta$ to be 0.1.

**4.2.5 Temporal multi-view fusion**

The temporally redundant information in video can be used to enhance the accuracy of scene parsing. In Xiao et al.’s work [30], they utilized multi-view information by defining a Markov Random Field for the entire sequence and imposing smoothness terms on superpixels in different images. The way we take advantage of multi-view segmentation consistency is different from theirs. The output of randomized decision forest classifier is a posterior probability distribution which represents the probabilities that a certain testing superpixel belongs to each class. Although we prefer to deem each superpixel belonging to one class, there are some thin structures, such as a column pole, which are far from filling the whole superpixel. In this case, a pixel-wise refinement is needed to achieve more accurate results. Moreover, for those superpixels which are misclassified, temporal fusion is able to increase the chance of making
refinement based on neighbor frames’ classification results. The fused probability distribution of each pixel can be represented as: 

$$p_r(c) = \sum_{j \in N} w_j p_j(c), c = 1, 2, ..., k$$

where $p_r(c)$ represents the probability of a pixel in reference view belonging to class $c$, $p_j(c)$ is probability of the correspondence pixel in neighbor view $j$ belonging to class $c$. $N$ represents the set of neighbor frames and $k$ is the number of classes. The fused probability distribution is computed as weighted average of neighbor frames’ probability distributions, where weights $w_j$ are determined by how far the neighbor frame $j$ is from reference view $r$. We define $w_j$ as a Gaussian function:

$$w_j = \exp\left(-\frac{(j - r)^2}{\sigma}\right)$$

where $\sigma$ is set to 10 in our implementation.

After pixel-wise temporal fusion, we applied a superpixel level refinement by aggregating the refined probability distributions of all pixels within a superpixel. In this way we incorporate segmentation consistency across adjacent views.

### 4.2.6 Cross training and testing

We propose the idea of cross training and testing based on the fact that using per-pixel depth information is independent on camera pose and appearance. In our five 3D features, only surface normal is dependent on camera pose because we compute each 3D patch (a superpixel in 2D)’s normal in camera coordinate system. For cross view training and testing, surface normals in training and testing datasets should be transformed to a common coordinate system. We define the common coordinate system to be as the following: taking the surface normal of ground and camera
moving direction (vertical to ground surface normal) as \( y \) axis and \( z \) axis, \( x \) axis can be obtained by the cross product of them. Note that since we only need to transform surface normals, the origin of the coordinate system does not matter. Any normal calculated in the camera coordinate can be rotated to the common coordinate frame, enabling cross training and testing.

4.3 Experiments

We use the challenging CamVid database \cite{91} to evaluate our algorithm’s performance. The database includes four high quality video sequences at 30 fps with total duration about 10 minutes. The labeled ground truth images are extracted from the four original video sequences at a rate of 1 fps, with a part of one of the sequences at 15 fps. The image resolution is \( 960 \times 720 \). Camera extrinsic and intrinsic parameters are also provided in the database. 32 semantic object classes are defined which include fixed objects, types of road surface, moving objects (such as vehicles and people) and ceilings (such as sky, tunnel, archway). Same as in \cite{1}, we use 11 dominant categories: Building, Tree, Sky, Car, Sign-Symbol, Road, Pedestrian, Fence, Column-Pole, Sidewalk and Bicyclist. Labeled colors for each object class are shown in Figure 4.3(a). A quantitative comparison to the state of art is provided. In addition, in order to show our algorithm’s compatibility of view-independent training and testing, we carried out another test on images captured by a side-view camera provided by Google. Even using our classifier trained by the CamVid database which is mostly composed of front-view video sequences, we still get decent classification results.
4.3.1 Evaluation using the CamVid database

For comparison with the results from [1], we split the training labeled frames into two groups in the same way as in [1], shown in Figure 4.3(b). Two groups of the labeled training data, 0016E5 and 0006R0, are used for day sequence training data, and another group 0005VD are used for day sequence testing. The first half of the dusk sequence (0001TP) are used for training, and the second half are used for testing.

We train our classifier using randomized decision forest based on the five dense depth based 3D features and carry out the same set of experiments as in [1]. Table 4.1 shows the quantitative testing result. In terms of global accuracy (i.e. pixel-wise percentage accuracy), we achieve 82.1% in comparison to 69.1% of combined structure-from-motion and appearance based approach and 61.9% of solely structure-from-motion based approach in [1]. Our algorithm also performs well on most of per-class accuracies and outperforms combined motion and appearance based approach in 7 classes out of 11 classes, which are Building, Sky, Car, Sign-symbol, Road, Column-Pole and Bicyclist. There are two categories (Sidewalk, Pedestrian) that our ap-
Table 4.1: Comparison of Pixel-wise percentage accuracy with [1]. Our dense depth map based approach gives best result on 7 classes. ‘Global’ is the percentage of pixels correctly classified. ‘Average’ is the average value of per-class accuracies.

<table>
<thead>
<tr>
<th>Alg</th>
<th>Building</th>
<th>Tree</th>
<th>Sky</th>
<th>Car</th>
<th>Sign-Symbol</th>
<th>Road</th>
<th>Pedestrian</th>
<th>Fence</th>
<th>Column-Pole</th>
<th>Sidewalk</th>
<th>Bicyclist</th>
<th>Average</th>
<th>Global</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motion</td>
<td>43.9</td>
<td>46.2</td>
<td>79.5</td>
<td>44.6</td>
<td>19.5</td>
<td>82.5</td>
<td>24.4</td>
<td>58.5</td>
<td>0.1</td>
<td>61.8</td>
<td>18</td>
<td>43.6</td>
<td>61.8</td>
</tr>
<tr>
<td>Appearance</td>
<td>38.7</td>
<td>60.7</td>
<td>90.1</td>
<td>71.1</td>
<td>51.4</td>
<td>88.6</td>
<td>54.6</td>
<td>40.1</td>
<td>1.1</td>
<td>55.5</td>
<td>23.6</td>
<td>52.3</td>
<td>66.5</td>
</tr>
<tr>
<td>Combined</td>
<td>46.2</td>
<td>61.9</td>
<td>89.7</td>
<td>68.6</td>
<td>42.9</td>
<td>89.5</td>
<td>53.6</td>
<td>46.6</td>
<td>0.7</td>
<td>60.5</td>
<td>22.5</td>
<td>53.0</td>
<td>69.1</td>
</tr>
<tr>
<td>Depth</td>
<td>85.3</td>
<td>57.3</td>
<td>95.4</td>
<td>69.2</td>
<td>46.5</td>
<td>98.5</td>
<td>23.8</td>
<td>44.3</td>
<td>22.0</td>
<td>38.1</td>
<td>28.7</td>
<td>55.4</td>
<td>82.1</td>
</tr>
</tbody>
</table>

Approach’s performances are not as good as in [1]. Both can be attributed to the lack of high-quality depth maps, in which the depth difference between the ground and sidewalk is small. Pedestrian is moving so its depth map is usually wrong. In addition, there are not sufficient samples for the Pedestrian class. Applying superpixel based training on such small classes may face the problem of insufficient samples compared with pixel-wise training approach. Figure 4.4 shows the qualitative results achieved by our approach.

**Testing on illumination variation**

One expected advantage of our approach is its independence of appearance or illumination changes. We carried out a test training in one lighting condition (day/dusk) and testing on the other (dusk/day). [1] also carried out the same test using sparse 3D features and appearance features. We compare our results with theirs in Table 4.2 and Figure 4.5. As expected, our method has significantly improved the global classification accuracy. The improvement from dusk-training-day-testing is less than that from day-training-dusk-testing, probably due to the poor depth maps generated at dusk. Using active sensors, such as LiDAR scanners can alleviate this problem.
Figure 4.4: Scene Parsing result samples. From top to bottom: test image, ground truth, dense depth map inferred segmentation. Note that our algorithm solely using dense depth map is able to achieve accurate segmentation and recognition of street scene.

Table 4.2: Comparison of pixel-wise percentage accuracy with [1] in illumination variation test.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Day Train - Dusk Test</th>
<th>Dusk Train - Day Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mot&amp;Struct</td>
<td>45.5%</td>
<td>59.4%</td>
</tr>
<tr>
<td>Appearance</td>
<td>21.7%</td>
<td>50.5%</td>
</tr>
<tr>
<td>Depth</td>
<td>63.4%</td>
<td>69.2%</td>
</tr>
</tbody>
</table>

**Multi-View Temporal Fusion Test**

We test our multi-view temporal fusion algorithm the test sequence Seq05VD. A neighborhood of 40 frames of the reference frame are used for fusion. Some results are shown in Figure 4.6. It can be seen that fusion improves the overall consistency of the segmentation, which is particularly pronounced when viewing the sequence as a video. Quantitatively, however, we found that the overall accuracy just improved by 1%. Further investigation in this topic is needed.
Figure 4.5: Robustness to lighting condition changes. In the left figure, from top to bottom: test image, ground truth, segmentation results. The left two columns are examples of training by a dusk sequence and testing on day sequence. The right two columns show results when the training and the test sequence are swapped. Table 4.2 shows the overall accuracy and comparison to numbers reported in [1].

4.3.2 Cross Training Test

As mentioned above, we use per-pixel depth information which is independent of camera pose. It is reasonable to carry out a crossing training and testing experiment between two types of camera configuration: front-view (used in [1]) and side-view cameras (used in [30]). We use CamVid (front-view) database as training images and Google Street View images (side-view) for testing. All the 3D features in both training and testing database are transferred to the unified coordinate system. In practice, it is necessary to apply a scale change to the testing data due to the scale ambiguity in structure from motion. We use a person’s height as the reference. Our results are shown in Figure 4.7. It is visually comparable to the results from [30] in
which both training and testing are performed with Google Street side-view images.

We currently do not have access to ground truth labeled dataset, so a quantitative comparison is not available.

### 4.4 Summary

In this chapter, we presented a novel framework for semantic segmentation and recognition based solely on dense depth maps. Our main contribution is that we have shown that dense depth maps generated by stereo contain plenty 3D information for scene parsing and can outperform segmentation combining both sparse 3D features and color appearances. Our method does not rely on any appearance information, making it robust against lighting changes. In addition our full 3D metric representation
Figure 4.7: Preliminary result of changed-view training and testing. From the side-view video sequence, we can see that our algorithm trained based on front-view sequences still obtained acceptable results.

is independent of camera configurations, therefore we can use one set of front-view or side-view video for training and the other set for testing. This is not possible using appearance information dependent on camera poses and illuminations.

The accuracy of our approach depends on the quality of depth map. While we have applied state-of-the-art algorithms to calculate depth map, the quality of depth map is still quite fragile. Our next step is to apply laser range scan data that have significantly better accuracy and consistency. We expect to see large performance improvement with better input data. In addition, one future area of work is the bias against less frequently appearing objects, such as thin column poles. This is mainly due to lack of sufficient training examples, which naturally lead to a less statistically significant labeling for objects in these classes. One possibility to preserve the classification of less frequent object classes could be to include context information that may boost the significance of objects in certain cases.
Chapter 5 Depth-based Personal Photograph Enhancement Using Internet Photo Collections

In last chapter, we introduce a novel depth-based semantic segmentation framework for video sequence. This chapter will explore another problem which intelligently utilizes 3D depth information for various image enhancements. Given the growth of Internet photo collections we now have a visual index of all major cities and tourist sites in the world. However, it is still a difficult task to capture that perfect shot with your own camera when visiting these places, especially when your camera itself has limitations, such as a limited field of view. Therefore, we propose a framework to overcome the imperfections of personal photos of tourist sites using the rich information provided by large scale Internet photo collections.

Figure 5.1 shows an overview of several challenging photo enhancements made easier by prior information gained from IPCs. Our approach assumes that IPCs of the relevant sites have been processed in advance so that geo-located 3D models of the relevant landmarks are available.

The proposed method starts by finding a set of 3D landmark models potentially associated to the personal image. We use an iconic scene graph based search over the landmark models to identify a few potentially corresponding landmarks and their 3D information. Next, we identify the corresponding landmark through geometric verification by registering the personal photo with respect to the 3D model using SIFT feature matching and a RANSAC based robust pose estimation. A bun-
Figure 5.1: Overview of our photo enhancements. From left to right: original image, foreground segmentation, photometric enhancement, stereoscopic image conversion, field of view expansion, and geo-tagging. The saturation (over-exposure) in right corner of the original image is removed by photometric enhancement, which is used as input for all the following enhancement applications. The field of view expansion result is intentionally resized for better figure layout. All photo enhancements are realized using Internet photo collections of the same landmark, Notre-Dame.

dle adjustment refines the obtained registration of the personal photo. Then, a novel automatic foreground segmentation technique for separating occluding foreground objects from visible parts of the 3D model is used. After these pre-processing stages, we proceed to demonstrate four types of challenging enhancements to the personal photo: photometric enhancements (saturation and glare artifacts removal), stereoscopic image synthesis, field of view expansion, and geo-tagging, on seven different landmarks from man-made architecture to natural scene.

5.1 Modeling from IPCs

In order to verify the applicability of our proposed method, we apply two commonly used image-based reconstruction pipelines for testing. For IPCs that span a unique landmark, we reconstruct the camera locations using Bundler [32], which is an incremental structure-from-motion pipeline. We increase the density of the obtained point cloud by using PMVS [34]. For large collections that span several landmarks across a city, we use an iconic scene graph approach [35]. We refer to the resulting 3D point cloud as our 3D model in future tasks. In order to offer a scalable solution,
each image is represented by a binarized GIST descriptor \cite{92} and the dataset is then clustered using K-medoids. Each of the obtained clusters is geometrically verified in a parallel fashion. Finally clusters are combined using hierarchical structure from motion.

We found that both of these solutions proved to be very successful on many IPCs. Non-rigid objects such as people in front landmarks are automatically ignored, and only rigid structures (such as the landmark itself) are reconstructed.

To fully exploit the reconstructed 3D model, a ground model is often necessary. Unfortunately, there are often too few reliably matched features on the ground to reconstruct an accurate 3D ground model from the photo collections. As a result, we design a simple interactive fitting tool that is used when no ground points are available in the 3D model. A RANSAC based automatic facade plane fitting is firstly applied on landmark, and then we specify the intersection line of this facade plane with the ground by selecting two points in the 2D image. The ground plane is derived by assuming that the facade plane and the ground plane are perpendicular in the real world. This assumption is valid in most man-made structures, and provides sufficiently accurate ground planes for our methods with little user interaction (only two clicks).

5.2 Personal Image Registration

Given a new image $Q$ from one’s personal photo collection (PPC), in order to perform image enhancement we first need to register $Q$ to the reconstructed 3D landmark model $M$. This can be considered as performing an incremental update of the recon-
In order to offer a scalable registration process for photos to an IPC, we propose a hierarchical matching approach, which first identifies a small set of potentially corresponding landmarks and then verifies registration to these landmarks. We use global image descriptors as shown in [35] to search for the $k$-nearest neighbors of image $Q$ in the binarized GIST space [92]. This identifies a set of potential matching landmarks.

Then, a SIFT matching [93] between the candidate image $Q$ and the collective SIFT descriptors of the 3D points of images registered within corresponding 3D model of each of the landmarks $M$ is performed. To improve the robustness and the efficiency we used a mean-shift clustering [94] of the SIFT descriptors of the IPC model along the lines of Irschara et al. [95]. This camera pose is further refined non-linearly to obtain the optimized camera pose of $Q$.

While this method is efficient, it does not offer a satisfactory registration rate due to the fact that it only registers to views that contributed to the 3D models, and the surrounding areas of the landmark are often not represented in the 3D models. The key insight we use is that contrary to the un-ordered IPCs, personal photos are often a stream of images acquired during a tour of a city based on the referenced clock of a single camera. Hence the registration can be improved by jointly registering images from the PPC taken around the same time, as they are likely taken in close spatial proximity. Accordingly, for images that do not directly register with any model, we register them through robust fundamental matrix estimation, if they have overlap with registered images captured within a time range of up to an hour. Please note
Figure 5.2: For images that initially failed to register to a 3D model (first image in second row), we search for a match in the temporally neighboring images (second and third images in second row). After constraining their locations nearby, we are able to successfully register the input image to one of the reconstructed models (Among the three reconstructed models in first row, the input image is registered to the right one).

the choice of this range is not sensitive. If an overlap to a registered image in the PPC is found we use transitivity to propagate the registration. Examples of images registered using the time constraint are shown in Figure 5.2 for images taken over a time frame of 10 min.

5.3 Image Foreground and Background Segmentation

One important prerequisite for many photo editing operations is the segmentation of the foreground. As opposed to interactive segmentation methods that rely on user interaction to learn the foreground and background appearance models, our method can acquire training data automatically based on the IPC. The fundamental assumption we make is that a pixel belonging to the background landmark is likely to
be photometrically consistent across other views, whereas a foreground pixel usually is not.

Our method first projects the 3D model $\mathcal{M}$ onto the image $Q$ denoted by $m$. The next step finds a set of images from the database that are captured at nearby locations under similar camera poses and image conditions of $Q$, denoted by $S$. Suppose a visible 2D point $p \in m$ is projected from the 3D point $P$ and we denote its neighboring 3D point set as $N(P) = \{P' : \|P - P'\|_2 \leq 3 \cdot l\}$, where $l$ is the average spacing between two closest 3D points. We then compute $\text{NCC}(p, I_i)$, the normalized cross correlation of the color values of the projection of $N(P)$ on the image $Q$ and the projection on an image $I_i \in S$. We consider $p$ is consistent between the image $Q$ and the image $I_i$ if $\text{NCC}(p, I_i) \geq 0.6$ or $p$ is inconsistent if $\text{NCC}(p, I_i) \leq 0.2$. If $\text{NCC}(p, I_i)$ is low because the projection of $N(P)$ lies on occlusion boundary, we still treat $p$ as inconsistent between image $Q$ and $I_i$. If $p$ is consistent with the majority, i.e., over 80% of total number of images in $S$, $p$ is classified into the background seed set $B$; similarly, if $p$ is inconsistent with majority, $p$ is classified into the foreground seed set $F$.

We revise the initial setup of Grabcut [51] framework in two aspects: (1) we use automatically generated training data $F$ and $B$ to initially build the Gaussian Mixture Models for foreground and background instead of user-provided bounding box; (2) we add a constant penalty to the unary term of each pixel $p \in F$ (or $p \in B$) if $p$ is labeled as background (or foreground) at the first run. We then perform the iterative energy minimization from Grabcut [51] to compute the segmentation. Figure 5.3 and Figure 5.4 compare fully automatic segmentation results from our method with Grabcut. Due to our precise color seeds used to train the appearance models for
both foreground and background, our automatic approach achieves more accurate and meaningful segmentation than Grabcut.

Figure 5.3: Segmentation results: (top left) the original image superimposed with the bounding box prior for Grabcut, (top right) red dots for foreground color seeds and blue dots for background color seeds, (bottom left) the result of Grabcut, (bottom right) the result of our automatic approach.

5.4 Photo Enhancement

After the above pre-processing steps, we have a foreground-segmented image with a detailed 3D model. This rich representation allows us to perform a number of enhancements to the input image.

5.4.1 Stereoscopic Image Synthesis

Recently, stereoscopic 3D has seen a huge boom in popularity due to its ability of providing users with a more immersive viewing experience. Due to a combination of technological advances and the success of 3D movies in cinemas, home 3D displays
have become increasingly commonplace. However, the production of personal stereoscopic content is still far from prevalent. Though Fujifilm stereo camera has started guidance for personal stereoscopic content creation, options for personal stereoscopic content creation are still limited, and general-case automatic 2D to 3D conversion is an unsolved and highly under-constrained problem. In this section, we propose a framework for generating a convincing stereoscopic pair from a single 2D personal photograph using prior information derived from large scale IPCs.

In order to generate a stereo pair from a single image, we must first compute depth values for input image. We first apply the automatic foreground segmentation described in Section 5.2 and the depth values of background pixels are computed by projecting 3D model $M$ to image plane. Note that sparse depth values are enough for our stereoscopic view synthesis described later.

Assigning depth values to foreground pixels however, usually requires user interaction. For images with a computed ground plane, we can assign the depth value for the foreground layer by back-projecting the ground contact point, e.g., one’s feet, onto the 3D ground plane. However, for images without a visible ground plane, we allow the user to adjust its depth value interactively. In the most difficult case of multiple layers in the foreground, we developed a simple UI to allow further separation of the foreground. Our system supports strokes $\text{[54]}$ and bounding boxes $\text{[51]}$ to interactively separate different foreground objects. One such example is shown in Figure $\text{[5.4]}$. It should be emphasized that this interactive step is only needed for images with complex foreground. Please see the supplemental material for an additional example of this interaction.
After depth values for the scene are estimated, a virtual camera pose is computed such that the resulting stereo pair provides users with a natural and comfortable 3D viewing experience. It is indicated that for typical desktop displays with a viewing distance in the region of 700mm, the comfortable perceived depth range is 50mm in front and 60mm behind the display surface [96]. Similarly, we can use the same method to retarget the photo to any other display. We compute a virtual camera pair with an optimal baseline such that the disparity range is mapped into this comfort.
zone of human visual perception using the mathematical derivation from the same work. We choose to synthesize both left and right views centered around the original image, rather than creating one virtual camera at twice the distance, as the reduced size of disocclusion regions lead to less artifacts and more convincing stereo results. Given the left and right camera poses and the depth map, we project pixels from known points into both views, computing a sparse set of disparities, which is used to synthesize the stereo pair.

**Virtual view generation**  Stereoscopic view generation is a special case of the general virtual view synthesis problem, for which several classes of solutions exist. The most common of these is depth-image based rendering (DIBR), where a dense depth map is used to project each pixel into a novel view. However, these methods require per-pixel depth values, which can be difficult to compute, especially in untextured (sky) and unknown (disocclusion) regions.

Instead, we adopt a recent approach designed for 2D-3D conversion from scribble input [47] which we describe here for completeness. This method makes a piecewise continuous assumption that allows for discontinuities (determined automatically by our foreground segmentation) to appear at depth boundaries. To avoid disocclusions at these discontinuities, a two-step process is used. The first step computes a piecewise continuous image warp driven by our sparse disparity constraints. The second step stretches the background, using a content and disparity-aware retargeting method to fill in any disocclusions that have arisen. This method allows us to automatically generate high quality stereo pairs using our IPC computed depth prior.
Figure 5.5 shows some results of our stereo view generation. We also provide a comparison showing the naive approach of using a simple planar background after segmenting the foreground, as shown in Figure 5.6. We can see that our model $M$ provides a more realistic and convincing depth impression by giving shape to the background regions. In addition, in cases where no foreground exists, we can still achieve a compelling stereoscopic image.

![Figure 5.6: Comparison showing the effect of incorporating our 3D model $M$ into the stereoscopic conversion. a) Original image, b) result computed using a planar approximation to the background, c) result computed using $M$. In the latter case, the rocks on the right can be seen at their correct depth level. Please view zoomed-in on screen for best results.]

5.4.2 Field Of View Expansion

A common problem in photos is the limited field of view (FOV). This is particularly pronounced in self-portraits, such as the one shown in Figure 5.1. Here we discuss our approach to expand the original image’s FOV using the background model from an IPC. Compared to the stereoscopic synthesis, the expanded image can contain a significant amount of missing data. Therefore a different synthesis method is presented.
Figure 5.7: Results from FOV expansion, (top) original images (bottom) FOV expanded images. For the third set, the expanded background is blurred with a low-pass filter to match the input.

**Geometric registration** The first step for FOV expansion is to warp similar images from the IPC to the query image’s camera pose. The expanded region is then filled in mainly by the content from warped images, as well as texture synthesized using repetitive content.

Specifically, for pixels that exist in the 3D model $\mathcal{M}$, we use forward warping, projecting these points to the pose of the query image. While this accounts for a majority of registered pixels, projecting 3D point clouds can lead to holes in the image. In order to fix these small holes, we use a bilateral interpolation on the depth map prior to projection, which interpolates missing depth values weighted by color similarity and spatial distance measurements. For parts of the image that do not appear in the 3D model, mostly sky or ground in our scenes, we mainly use texture synthesis or homography warping from nearby images to fill in information. Because of the diversity of lighting conditions and foreground objects in IPCs, it is necessary to select images with high color similarity to use in FOV expansion. In our work, we use a SSD metric to measure color similarity of the reference image and the warped
source images. The top 20–50 images with highest color similarity are automatically selected for blending.

We choose a simple scheme for blending the selected images, using the median color of the top ranking matches, projected into the camera pose of the reference image. Figure 5.8 shows the quality of the median color images after forward warping, and also the regions which we are not able to model using the geometry. These areas are then filled either by texture synthesis or by warping nearby images with a homography.

**Using nearby images** Due to the power of large scale IPCs (or the similarity of people’s vacation photos), we found that often times we have numerous images from nearby camera poses. When selecting from these images, we favor images that have not only similar camera location and orientation, but also a wider field of view than image \( Q \), as these images will exhibit less distortion after warping, and will have a higher chance of containing the content needed to fill out our view. From these images, the top candidates are automatically chosen for blending the median image. Thanks to the large number of images in our database, we are often able to automatically find many images taken with very similar poses, which makes forward warping fairly accurate. However, for cases where there are no nearby images with the similar perspective, or when the query image has much higher resolution than source images, the forward warping naturally leads to blur in the filled image region (see Figure 5.18).

**Combining sources** Once we have geometrically and photometrically registered
the images, the next step is to fill the expanded areas. To achieve seamless blending of different registered images, we combine gradients from all the sources, forming a new gradient image with expanded FOV. The output gradients are automatically combined, and the priorities are: original image (1st), geometrically registered median image (2nd), homography warped nearby images (3rd) if we have to use(such as ground). Specifically, to compose gradients of expanded FOV, we keep gradients from original image, and then add in gradients from median image to expanded area. For regions that median image does not cover, we fill in gradients from homography warped images. After we compute our combined gradients map, we solve Poisson’s equation \([97]\) to seamlessly reconstruct the output image. Texture synthesis will be applied if sky region is missing, as will be described in next paragraph. When the background is out of focus, we allow users to specify a Gaussian blur for the expanded region, to match the reference image(column 3 in Figure 5.7). Figure 5.7 shows some results from our FOV expansion application.

![Figure 5.8: Median color images from selected geometrically registered images with high photometric consistency.](image)

**Content-aware scaling and texture synthesis** There are some extremely chal-
lenging cases where important occluders, such as people, are cropped by the image border. In this case there is no way to reconstruct the remaining portions of these occluders with our model $\mathcal{M}$ or with nearby images. We present a solution to this problem where we first apply automatic segmentation to extract the important foreground objects, and then use a content-aware image resizing technique [98] to stretch the remaining unimportant background regions (such as ground), while preserving the aspect ratio of foreground objects. This is used to expand the background, without creating artifacts due to foreground objects being cut off at the borders. As shown in the second image of Figure 5.7, by our novel usage of seam carving allows a photorealistic expansion of original image to be obtained for a difficult case.

Finally, as mentioned above, for sky regions (segmented using our geometric model) that remain unfilled after all prior compositing steps, we implement a texture synthesis approach [99] to fill the remaining holes. When there is no sky region at all in the original image, we use an infinite homography to warp the sky from the other nearby images, and then complete the region with texture synthesis. By combining a number of simple approaches, our system is able to achieve high quality results, filling in convincing information from a large set of registered images with minimal user interaction.

5.4.3 Photometric Enhancement

IPCs also provides an excellent sample set to fix up problematic areas in personal photos, in particular these areas on the background (e.g., the landmark). It should be noted that most cameras determine metering based on the central content, or
even on detected faces. Therefore it is more likely that the background part requires photometric correction.

**Flawed area identification** In this application we mainly deal with two types of enhancements, removing saturated regions and glare. We automatically detect saturated regions (over-exposed and under-exposed) by simply thresholding r,g,b values of pixels near 0 and 255. Automatic detection of glare is more difficult. Therefore we allow a user to identify the location of these artifacts, as shown in Figure 5.10.

**Image composition** We adopt the same scheme as in Sec 5.4.2 for computing a median color image over the registered images with high photometric consistency, using SSD to measure the color similarity of reference image and the warped source images (excluding areas that needs to be enhanced). Under this scheme, we find a median color image that contains important image detail within these saturated and glare regions, while maintaining color similarity to the reference image. To achieve seamless blending, we again perform Poisson blending [97], replacing gradients of the saturated regions in the personal photograph with those from median image.

**Tone mapping** Introducing detail into the saturated region can result in an HDR image whose dynamic range is beyond the 8-bits of the input image/display device. Therefore, as a post-processing step, we apply a standard tone mapping technique to obtain the final, viewable image. Figure 5.9 and Figure 5.10 shows some results of our photometric enhancement application and comparison with results from Photoshop.
Figure 5.9: Comparison showing the effect of our saturation removal using internet photo collections. a) Original image, b) Result from Photoshop color adjustment, c) Result from our method. Since saturated regions have no details, simple color adjustment cannot fix the problem. Please view zoomed-in on screen for best results.

5.4.4 Transfer of Information

IPCs are not limited to a set of images, they typically also incorporate meta-data such as geo-location and text tags. When a PPC is aligned to an IPC, these meta-data can be automatically transferred. This offers additional ways to browse your PPC either by displaying your images on a map or for example by grouping the images by label. In the following section, we describe how our system transfers geo-locations and text tags from IPCs to PPCs.

Geo-tagging While geo-location of the PPC photos is often desired for visualization effects (map of location of a collection, browsing, etc.) it is a tedious manual task to localize photos without GPS information. We find that in an IPC typically more than 99% of the photos do not contain GPS information. Our system removes the burden
Figure 5.10: Effect of our glare removal using internet photo collections. a) Original image with user selected glare region, b) Results using Photoshop Healing and Clone tools, c) Results from our method. Please view zoomed-in on screen for best results.

of manual localization from the user by automatically geo-localizing the photos of the PPC. We use the embedded tags (automatic and user clicked) of the images in the IPC. In order to support more accurate geo-location we also use Google StreetView panoramas, which have high accuracy geo-location and orientation\footnote{The panoramas used are automatically downloaded through the Google StreetView API http://code.google.com/apis/maps/index.html}.

First we translate the GPS latitude longitude information into an approximate local metric coordinate system the Universal Transverse Mercator (UTM) grid. This conversion allows us to consider Euclidean distances between geo-tags.

To obtain geo-localization for the PPC, our system uses a two-stage process.
In a first preprocessing stage executed once for each model in the IPC the system obtains an accurate geo-localization of the corresponding IPC model. This accurately localized model is then used to perform a geo-localization of the PPC images.

**IPC model localization:** Our algorithm first obtains an approximate geo-localization through a kernel voting. Each image that has a geo-tag votes for its location through a Gaussian kernel centered at its geo-tag location and a $3\sigma$ cut off distance of 25 meters for clicked geo-tags (approximate clicking accuracy) or a $3\sigma$ cut off distance of 10 meters for GPS based geo-tags. To suppress outliers we then select all geo-locations within the biggest mode as the set of valid geo-tags. The approximate location of each model is then obtained by the averaging of the inlier locations. Alternatively, when there are no geo-tags or no reliable geo-tags (no consistent votes) we use the text-tags of the images of the IPC for a location query on Google Maps to obtain an approximate geo-location.

The approximate geo-location of the IPC model is then used to obtain all nearby Google StreetView imagery (panoramas) available for the refined geo-localization, whose location is likewise transferred into the UTM coordinate system. Given that these images contain mostly road surfaces and cars below the horizon line we discard all information below the horizon for the further processing, (the horizon can be directly obtained from the image orientation). All panoramas are then registered into the model using our registration process from Section 5.2 but instead of using the standard three-point algorithm we use a three-point algorithm based on viewing rays given that the panorama directly provides viewing rays. Then using a RANSAC
approach we transform the IPC model coordinate system into the UTM coordinate system. We use the known positions of the panoramas in the IPC model and the UTM coordinate system of the same panorama as correspondences to estimate the transformation from the IPC model coordinate system to the UTM coordinate system. This step delivers an accurate transformation from the IPC model coordinate system to the geo-coordinate system. We apply this transformation to geo-localize the IPC model, which is then be used in the next step to geo-localize the images of the PPC.

**PPC image geo-localization:** This is using the registration process discussed in Section 5.2 with respect to the geo-located model. This directly obtains geo-location in the UTM coordinate system for the photos of the PPC.

Typically there is a large fraction of the IPC images with geo-tags that are not registered with our model. These images provide valuable geographic information about less popular scenes in an IPC. Therefore for the images that failed to register to a geo-localized model, we search for matching images in the set of geo-tagged images using the same method as in Section 5.2 including trying to match images of the PPC taken at approximately the same time. The obtained geo-tags are then filtered using the same kernel voting as the IPC model geo-location. Matching to the complete set of geo-tagged images from the IPC drastically increases the number of geo-localizable images in the PPC.

We tested this geo-tagging approach on several PPCs that were taken in Berlin by three different users. We used an IPC consisting of 2.8 millions images retrieved from Flickr (including 353,584 geo-tagged images) and 467 geo-localized 3D models.
Figure 5.11: Geo- and textual tagging of personal photos using IPCs of Berlin and Notre Dame. Locations depicted with green pins correspond to the query images that are matched a 3D model, geo-tagged and geo-localized. Blue pins represent images from IPC.

In this case we were able to geo-tag more than 55% of the input images with an estimated accuracy of approximately 50 meters, which is related to the quality of input geo-tagged information. Visual geo-tagging results are shown in Figure 5.11.

**Image labeling** Labeling images offers additional information to ones PPC. It not only allows the user to search images in his or her PPC based on keywords, but also to retrieve additional information about a photo. For example, the tags found for an image of a monument are usually precise enough for Google to retrieve the corresponding Wikipedia article.

In order to offer precise tag candidates to the user, we propose a hierarchical automatic annotation algorithm. First we select the most popular tags from the complete IPC from which the model was computed. Then for each image we select the most popular tags from the particular landmark it was registered to, excluding the previously attached tags from the IPC. If an image is registered to several landmarks, we select the tags that have the highest combined mean occurrence across the landmarks.
(number of times a given tag is represented in the dataset divided by the total number
of tags). Finally we add tags coming directly from the images the candidate image
registered to, adding a few more localized tags.

In addition to these effects, we could also easily perform other enhancements based
on the scene depth, such as refocus, depth-of-field control, etc. These effects have
been demonstrated with depth obtained by other means, therefore we will not show
examples in this paper.

5.5 Experiments

To further evaluate the performance of our system, two groups of evaluations are
conducted. The first one evaluates the operating range of our methods, such as how
the results depend on the number of images in database, or the resolution/quality
of the query image. The second evaluates how convincing users find the enhance-
ment results, as well as our system’s processing speed and interaction requirements,
especially by comparing with state-of-the-art image editing software.

5.5.1 Operating Range Evaluation

The more images the database contains, the richer information we can obtain. In this
evaluation, we evaluate how our enhancement results depend on the number of images
in database. We choose the Mount Rushmore dataset to demonstrate it. We carried
out the same operations, varying only the number of images in database - 500 images,
250 images, 80 images. As shown in Figure 5.12, the first row demonstrate that with
more images in database, the 3D effects are more realistic. In the result using a
500 image database, we can see more 3D geometry variance in the far rock, which disappears when only using 250 images. When only 80 images are used, the depth effect becomes unnatural in the background. Those differences are due to the density of the reconstructed 3D point clouds. The second and third rows show that more images induce more realistic FOV expansion and photometric enhancement, due to a larger number of photometrically-consistent images that can be drawn from. This can be seen especially in the area between the two rightmost faces in FOV expansion case, and the leftmost face in photometric enhancement case. Similar comparison can be seen in another dataset (Row 4-6 in Figure 5.12).

Another factor that affects our system’s performance is the resolution/quality of the query image, especially on FOV expansion application. If the query image is of high resolution compared to other images in database, the geometric warping from database images to reference image will be more blurry, therefore producing a more blurry median image and final FOV expanded image. Conversely, if the query image is lower resolution than other images in database, the warped images can appear sharper than the query image. Generally, the more consistent resolution/quality a query image has with images in database, the better performance that our system can achieve. As shown in Figure 5.13, an image is expanded at two different resolutions, using the same IPC. Although the overall resolution of (b) is much higher than (d), the expanded region of (d) is more consistent to the quality of original image(c), therefore looks more natural, while in (b) the expanded region is more blurry relative to the quality of original image(a), which looks a little bit unnatural.
Figure 5.12: Comparison showing how the results depend on the number of images in database. Row 1-3: Mount Rushmore Dataset. Row 4-6: Notre Dame Dataset. From top to bottom: 2D to 3D conversion, FOV expansion, photometric enhancement. Please view zoomed-in on screen for best results.
Figure 5.13: Performance related to query image resolution. a) Original image, b) Result from original image, c) Downscaled original image (X5), d) Result from downscaled (X5) original image. The expanded region of (d) is more consistent to the quality of (c), therefore looks more natural, while the expanded region of (b) is more blurry relative to the quality of (a), which looks a little bit unnatural. Please view zoomed-in on screen for best results.
5.5.2 Performance Evaluation

We conducted a user study to evaluate how realistic our photo enhancement results are compared to the processing results from state-of-the-art image editing tools, e.g., Photoshop. Four Photoshop experts were asked to process 20 query images from 6 different scenes, among which 4 images are used for segmentation, 6 images for 2D to 3D conversion, 4 images for FOV expansion, 6 images for photometric enhancement.

Based on the design goal of our system, we hypothesize that using our system will be able to complete the tasks significantly quicker than using Photoshop, the state-of-the-art image editing tool. The average time required to complete the four tasks with our system and Photoshop are illustrated in Figure 5.14. We can see that for all the four tasks, our system takes much less time than using state-of-the-art image editing tools. ANOVA tests confirm that the time differences are statistical significant for all the four tasks($F = 10.87, p-value = 0.008 < 0.01$ for “Foreground Segmentation”; $F = 30.42, p-value = 0.0002 < 0.01$ for “2D to 3D Conversion”; $F = 17.21, p-value = 0.006 < 0.01$ for “FOV Expansion”; $F = 47.04, p-value = 0.002 < 0.01$ for “Photometric Enhancement”), which further validates our hypothesis.

25 users are asked to compare the Photoshop results with our enhancements on the same images. To avoid bias, images from two methods are randomly ordered and users do not know which image comes from which method. Users were required to select one of five preference choices: 1) Image 1 is much better than Image 2; 2) Image 1 is slightly better than Image 2; 3) Image 1 is equal to Image 2; 4) Image 1 is slightly worse than Image 2; 5) Image 1 is much worse than Image 2.
### Figure 5.14: Average time used for processing under our system and Photoshop.

- **Foreground Segmentation:** Ours >> PS (much better)
- **2D to 3D Conversion:** Ours > PS (slightly better)
- **FOV Expansion:** Ours = PS (equal)
- **Photometric Enhancement:** Ours < PS (slightly worse)

![Bar chart showing processing times](image)

### Figure 5.15: User study result: Percentage of preference choices between our results and Photoshop(PS) results. The four rows represent four different applications. The expert-driven PS segmentation can be deemed as ground truth. Users' visual evaluations show that our segmentation results are comparable to ground truth. For the other three applications, our results receive much higher evaluation scores.

<table>
<thead>
<tr>
<th>Application</th>
<th>Ours &gt;&gt; PS (much better)</th>
<th>Ours &gt; PS (slightly better)</th>
<th>Ours = PS (equal)</th>
<th>Ours &lt; PS (slightly worse)</th>
<th>Ours &lt;&lt; PS (much worse)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreground Segmentation</td>
<td>6.00%</td>
<td>22.00%</td>
<td>25.00%</td>
<td>24.00%</td>
<td>23.00%</td>
</tr>
<tr>
<td>2D to 3D Conversion</td>
<td>30.56%</td>
<td>30.56%</td>
<td>19.44%</td>
<td>13.89%</td>
<td>5.56%</td>
</tr>
<tr>
<td>FOV Expansion</td>
<td>61.00%</td>
<td>28.00%</td>
<td>4.00%</td>
<td>5.00%</td>
<td>2.00%</td>
</tr>
<tr>
<td>Photometric Enhancement</td>
<td>26.67%</td>
<td>38.67%</td>
<td>25.33%</td>
<td>4.00%</td>
<td>5.33%</td>
</tr>
</tbody>
</table>
As shown in Figure 5.15, the user study clearly shows that our results are either favored over those from Photoshop, or comparable (for the difficult task of automatic foreground segmentation). It is worthwhile to point out that taking Photoshop segmentation as ground truth, our automatic foreground segmentation achieves an average error rate of 3%, with minimum error rate of 0.7% and maximum error rate of 8.7% (multiple foreground layers case). This indicates that our approach is capable of automatically producing comparable foreground segmentations to hand-tuned foreground segmentation maps in Photoshop. The error rate is computed as the percentage of mislabeled pixels.

Figure 5.16 shows a more intuitive diagram of user preference. We normalize the five statements in Figure 5.15 to numerical scores ranging from +2 to -2. The mean value of these scores lies on the boundary between red bar (ours) and blue bar.
The larger the normalized evaluation score, the more preferable our approach was to Photoshop. Figure 5.17 shows some visual comparisons between our approach and the Photoshop images that we used in the study. This user study validates our conclusions again that leveraging IPCs and simple user interaction allows us to create more convincing image enhancements than what is possible with state-of-the-art image editing tools used by experts.

In terms of geo-tagging and locating evaluation, unfortunately we are not aware of a system that tries to geo-localize PPC to IPCs. Most systems only register images to a geo-registered model. In that our approach is similar when we can register the input image to a model, but when this fail we offer a backup to transfer geo-tag which, as far as we know, was not proposed before. Therefore we believe that these results are relevant. As for the geo-registering of model we offer a robust system based on commodity panoramas (e.g. Google street view images). Current system uses accurate GPS or a combination geo-tags and maps. We currently do not compare to this system, but our attempts to geo-register our model using only the geo-tags have failed because of the amount of noise in the input.

5.6 Discussions

Like many IPC-based approaches, the success of our approach depends on the availability of large sets of photos taken at the same site. Therefore its application is currently mainly limited to images of touristic sites. However, there are on-going efforts for large-scale image acquisition through competitive games [100]. So we are optimistic that the applicability of our approach will be greatly expanded in the
Figure 5.17: Comparison showing the effect of our enhancements using internet photo collections. a) Original image, b) Result from Photoshop, c) Result from our method. From top to bottom: foreground segmentation, 2D to 3D conversion, FOV expansion and photometric enhancement. Please view zoomed-in on screen for best results. To see more photometric enhancement comparisons with Photoshop, please refer to Figure 5.9 and Figure 5.10. More comparison results are provided in supplemental materials.
Fully automatic image segmentation is a challenging task. As shown in Figure 5.4, our automatic approach sometime is not sufficient to further separate foreground layers for 2D-3D conversion. Therefore we still require user interaction to separate the foreground layers. This is the most time-consuming part of our entire system (for which we have prepared a video in the supplementary materials).

FOV expansion is also a difficult case, especially for areas that have no 3D information (Figure 5.18(a)) or inaccurate 3D information (Figure 5.18(b)). One example of such areas, are locations that are non-static, such as a fountain, which cannot be accurately reconstructed by our 3D model. This causes the corresponding location to become blurry when reconstructed from median color values of nearby IPC images (Figure 5.18(b)). One solution to this problem could be by including manual interaction to select patches where the result is sampled from a single image.

Generally, our proposed enhancement methods work on both daytime and nighttime images, however, personal image registration is more challenging for nighttime image. As long as there exist photometrically-consistent photographs to input image in database, the gradient field of created median image is good to use for our proposed enhancements. If the input image is taken under too extreme condition (e.g. sunset or glossy scenes), there is high possibility that very few photos under similar condition can be retrieved from database, the proposed method might not work quite well. Another limitation is that our median image based method might result in inconsistent lighting, since we didn’t take into account the lighting direction when creating median image, especially when there is strong directional lighting in the input image.
Figure 5.18: Limitations of our approach: (a) Missing data behind the columns caused bad interpolated depth values, and consequently a low-fidelity synthesized view in the column areas. (b) Another example of field-of-view expansion. The green box is the original photo. Note that the flowing water from the fountain on both the left and right sides, and the fountain rocks are blurry.

5.7 Summary

In this chapter, we approach personal photo enhancement from a novel direction - using IPCs. Our work leverages the 3D background models reconstructed from IPCs of the same landmark. With the rich information from large scale IPCs, we believe that by augmenting one’s personal photo with depth information, as well as the surrounding appearance information, a number of interesting photo enhancements can be achieved. Applications that we have explored vary from automatic image segmentation, stereoscopic view synthesis, to field of view expansion, photometric enhancement, geo-tagging etc, all of which show promising results and validate the potential of our approach.

Although the 3D model is reconstructed from internet photos using the state-of-the-art techniques, the sparsity and inaccuracy of the 3D model still contribute to some failure cases in photo enhancements. Newer approaches like Frahm et al. [35]
provide dense geometry which could help overcome the sparsity limitations. The 3D depth augmentation of 2D images also enables additional enhancements such as super-resolution or 3D re-targeting by using the full variety in resolution and illumination conditions captured by the IPC.

Finally, in this chapter we only focus on single image enhancement. However one future direction could be to extend our work to multiple images or even video sequences for enhancement.
Chapter 6 Depth-based Flower Petal Modeling with Botany Priors

In last two chapters, we use 3D depth information to solve semantic segmentation and image enhancement problems. The third problem we demonstrate that using depth can benefit is 3D parametric modeling. We focus on the 3D parametric modeling of flower, in particular the petals. The complex structure, severe occlusions, and wide variations make the reconstruction of their 3D models a challenging task. Therefore, even though the flower is the most distinctive part of a plant, there has been little modeling study devoted to it. We overcome these challenges by combining data driven modeling techniques with domain knowledge from botany. It should be emphasized that our reconstruction pipeline generates a parametric model, which is particularly suited for measurement, editing, and animation. For example, one could easily apply a geometric morphing between two models, or make global changes to the shapes by varying shape parameters. While these are not explored in the context of this paper, we believe our approach will enable more research in the modeling and animation of an intrinsic class of objects, flowers, with applications in botany, entertainment, and visual simulations.

Figure 6.1: From left to right: 1) Petal database for Lily species; (2) Input image; (3) Petal segmentation; (4) Scanned 3D data; (5) Reconstructed 3D model.
To capture the geometric details, we choose to use structured light scanner to acquire high quality 3D data of flowers. Our method starts from scanning individual flower petals with variations but from the same species, and building up a morphable model \[101\] for petal shape of a certain species. A level-set based active contour model is used for accurately segmenting the 2D image and 3D scanned points of a whole flower into different components (petals). Both 2D appearance and 3D depth information are used to guide the segmentation. Each segmented petal point clouds is subsequently fitted using the morphable model. We propose a joint multiple petal fitting algorithm using prior knowledge from Botany about flower spatial layout. Finally, the registered color image is used to generate texture maps for the 3D model. We will illustrate each part in details in next several sections. An overview of our approach is shown in Figure 6.1.

### 6.1 Scale-invariant Morphable Model

We choose to use a learned morphable shape model to reconstruct flower petals because of the parametric nature of the model representation. The benefit of using morphable model is that the optimization affects the entire petal, as opposed to per-vertex based deformation method, therefore can robustly handle occlusions. The uniform parametric form of reconstructed shapes will benefit many future work, such as modeling the development of flowers.

To build the morphable model, we scan a collection of exemplar petals from the same species, but with noticeable variations. Figure 6.1 left shows one database of Lily with 108 exemplars while Figure 6.2 shows a Pansy database with 60 exemplars.
Figure 6.2: Petal database for Pansies (60 exemplars).

The shape and size variations are not negligible even within the same species. To build the morphable model, we firstly align each exemplar shape to a reference shape via two principal axes. Then a non-rigid alignment is performed using CPD \[102\] to deform each exemplar shape to best fit the reference shape, in which way we obtain the correspondences. The correspondences are used to estimate a similarity transformation to transform each shape to reference shape coordinate system. Then we sample approximately 3500 vertices on the reference shape, and represent the shape of a petal by a shape-vector \( S = (v_1, v_2, ..., v_n) \in \mathbb{R}^{3n} \), where each \( v_k \) is a three-dimensional vector representing 3D coordinate. The correspondences in transformed exemplar shapes are used to build a morphable shape model using Principle Components Analysis (PCA), defined as follows:

\[
S_{model} = \bar{S} + \sum_{j=1}^{m-1} \alpha_j s_j = \bar{S} + B\alpha
\]

(6.1)

where \( \bar{S} \) is the average of \( m \) exemplar shapes and \( B = (s_1, ..., s_{m-1}) \) are the eigenvectors of the covariance matrices defining petal shape space. \( \alpha = (\alpha_1, ..., \alpha_{m-1}) \) are the coefficients of basis shapes.
Different from traditional method for building the morphable model, we compute a scale factor when transforming exemplar shapes to reference shape. This scheme is designed to eliminate the size variation among exemplar shapes, but focus more on the statistics of shape variations for reconstructing the details of a petal. The mean shape in morphable model is always firstly scaled to match the size of the source shape in our petal fitting process (Sec. 6.3). In this way, we build up a scale-invariant morphable model that can be applied to reconstruct petals of tremendously different sizes.

6.2 Flower Petal Segmentation

There has been some work on segmenting whole flowers from a scene [103], but few has been done on segmenting each individual component (petal) of a flower. The main challenges are from the high appearance similarity and noticeable self-occlusions, which makes the segmentation very challenging. Therefore, we manually specify a central position on each petal as an initialization to guide the segmentation.

We apply distance regularized level set evolution [104] formulation to an active contour model [105] to solve the petal segmentation problem. Both 2D and 3D gradient information are embedded in the active contour model as guidance for segmentation boundary evolvements. We extend the two-region level set method to multiple regions by defining $p$ level set functions (LSF) $\phi_i$, $i \in (1, ..., p)$, where $p$ is the number of petals in a flower. Each LSF $\phi_i$ represents a region $\Omega_i$, by setting $\Omega_i(x) < 0$ when $x \in \Omega_i$; $\Omega_i(x) > 0$ when $x \notin \Omega_i$; $\Omega_i(x) = 0$ when $x$ is on the contour of $\Omega_i$.

Let $I$ be the color image and $D$ be the depth map projected from 3D scanned data.
We define a 2D gradient indicator function $g_c$ and a 3D gradient indicator function $g_d$ as
\[
g_c = \frac{1}{1 + |\nabla G_{\sigma} * I|^2}; \quad g_d = \frac{1}{1 + |\nabla G_{\sigma} * D|^2}
\] (6.2)

Our final gradient indicator $g$ is computed as a linear combination of $g_c$ and $g_d$
\[
g = \beta g_c + (1 - \beta)g_d
\] (6.3)

where $G_{\sigma}$ is a Gaussian kernel for smoothing the color and depth image to reduce noise. For each LSF $\phi_i$, we define an energy function $E(\phi_i)$ by
\[
E(\phi_i) = \lambda L_g(\phi_i) + \alpha A_g(\phi_i) + \mu R_{\hat{p}}(\phi_i)
\] (6.4)

By finding the minimum of $E(\phi_i)$, we can obtain the segmentation as the region $\Omega_i$ that $\phi_i < 0$ represents. $\lambda$, $\alpha$ are the coefficients of the energy functions $L_g(\phi_i)$ and $A_g(\phi_i)$. $\mu$ is the coefficient of distance regularization term $R_{\hat{p}}(\phi_i)$. They are defined as:
\[
L_g(\phi_i) = \int_{\Omega_i} g\delta(\phi_i)|\nabla \phi_i| dx
\] (6.5)
\[
A_g(\phi_i) = \int_{\Omega_i} gH(-\phi_i) dx
\] (6.6)
\[
R_{\hat{p}}(\phi_i) = \int_{\Omega_i} \hat{p}(|\nabla \phi_i|) dx
\] (6.7)

where $\delta$ and $H$ are Dirac delta function and Heaviside function, $\hat{p}$ is a potential function for distance regularization.

The energy in $L_g(\phi_i)$ computes the line integral of the function $g$ along the zero level contour of $\phi_i$, which is minimized when the zero level contour of $\phi_i$ is located
at the petal boundary indicated by $g$. The energy $\mathcal{A}_g(\phi_i)$ computes the weighted area of region where $\phi_i(x) < 0$. It is used to accelerate the movement of zero level contour in the level set evolution process, while slowing down when it arrives at petal boundaries where $g$ takes smaller values. The distance regularization term $\mathcal{R}_p(\phi_i)$ is defined for maintaining the signed distance property of LSF.

The energy function in Eq. 6.4 can be minimized by solving the following gradient flow:

$$
\frac{\partial \phi_i}{\partial t} = \delta(\phi_i) \left( \lambda \text{div} \left( g \nabla \phi_i \right) + \alpha g \right) + \mu \text{div} \left( d_p(\nabla \phi_i) \nabla \phi_i \right) + \mu \text{div} \left( \frac{\nabla \phi_i}{\sqrt{\nabla \phi_i}} \right)
$$

Minimization of Eq. 6.4 is under the constraint $\bigcup_i \Omega_i = \Omega$ and $\bigcap_i \Omega_i = \Phi$, namely, we want to prevent overlapped and vacuum regions. Therefore we employ the idea from [106] to enhance the evolution process of $\phi_i$ as:

$$
e_k := \lambda \text{div} \left( g \nabla \phi_i \right) + \alpha g
$$

$$
\frac{\partial \phi_i}{\partial t} = \delta(\phi_i) \left( e_i - \min_{\delta(\phi_j) > 0; j \neq i} (e_j, e_i - 1) \right)
$$

We initialize each LSF with a binary step function $\phi^0_i$ defined by

$$
\phi^0_i(x) = \begin{cases} 
-c & \text{if } x \in R \\
\frac{1}{c} & \text{otherwise}
\end{cases}
$$

where $c > 0$ is a constant, and $R$ is a square region centered at the initial position manually clicked on each petal. As shown in Figure 6.3, the initialized regions finally evolve to accurately match the boundary of each petal, with no overlap or vacuum.
6.3 Flower Petal Fitting

To handle the occlusions and maintain correct 3D spatial relations of multiple flower petals, we propose a joint petal fitting scheme, incorporating prior constraints from spatial layout information and segmentation results. It is worth noting that the input petal shape and the morphable shape model are in different coordinate systems. Therefore, we estimate a similarity transformation \((s, R, t)\) between the two coordinate systems that transforms reconstructed shape from morphable model space to input space for fitting. Specifically, suppose a flower has \(p\) petals and \(L\) different layers. Let \(L(i)\) be the layer where \(i^{th}\) petal lies in. We minimize the following energy over the set of model parameters \(\bar{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_i, ..., \alpha_p)\), where \(\alpha_i = (\alpha_i^1, \alpha_i^2, ..., \alpha_i^{m-1})\) is the shape coefficients of \(i^{th}\) petal. Suppose \((s_i, R_i, t_i)\) are the similarity transformation from morphable model space to the \(i^{th}\) petal.

\[
E(\bar{\alpha}) = \lambda_p E_P(\bar{\alpha}) + \lambda_c E_C(\bar{\alpha}) + \lambda_s E_S(\bar{\alpha}) \quad (6.12)
\]

There are three terms in this energy function, which are defined as:
\[ E_P(\vec{\alpha}) = \sum_{i=1}^{p} \| \mathcal{W}_i(s_i R_i(\bar{S} + B\vec{\alpha}_i) + t_i - C_i) \|^2 \] (6.13)

\[ E_C(\vec{\alpha}) = \sum_{i=1}^{p-1} \| (s_i R_i(\bar{S} + B\vec{\alpha}_i) + t_i)_{(k)} - (s_{i+1} R_{i+1}(\bar{S} + B\vec{\alpha}_{i+1}) + t_{i+1})_{(k)} \|^2 \] (6.14)

\[ E_S(\vec{\alpha}) = \sum_{L(r)=L(s)} \| \vec{\alpha}_r - \vec{\alpha}_s \|^2 \] (6.15)

The first term \( E_P(\vec{\alpha}) \) measures the distance between reconstructed model and the set of all correspondences \( C_i \) on \( i^{th} \) petal. \( \mathcal{W}_i = \text{diag}(w_{1i}^1, \ldots, w_{ni}^n) \otimes I_3 \) is the weight matrix for all vertices \( v_i = (v_1^i, \ldots, v_n^i) \) in the shape \( S_i \). \( I_3 \) is \( 3 \times 3 \) identity matrix.

When finding correspondences, we enforce boundary-to-boundary, and inner-to-inner matching between reconstructed shape and input. We also compute an occlusion map (occluded region) and occluded boundary (false boundary) for each input petal based on petal segmentation and layer information from Botany. Therefore, there are four cases that each vertex \( \vec{v}_i^k \) can be related to its correspondence \( C_i(\vec{v}_i^k) \) on input during reconstruction:

1) \( \vec{v}_i^k \) is on boundary of shape model and \( C_i(\vec{v}_i^k) \) is also on real boundary in input;
2) \( \vec{v}_i^k \) is on boundary of shape model but \( C_i(\vec{v}_i^k) \) is on false boundary in input;
3) \( \vec{v}_i^k \) is inside the petal model and \( C_i(\vec{v}_i^k) \) is not occluded in input;
4) \( \vec{v}_i^k \) is inside petal model but \( C_i(\vec{v}_i^k) \) is occluded in input.

For case 1, we set \( w_{ik}^k = w_{ib} \); for case 3, we set \( w_{ik}^k = w_{nb} \); for case 2 and 4, we set \( w_{ik}^k = 0 \). For a vertex that \( \| \vec{v}_i^k - C_i(\vec{v}_i^k) \| > \tau \), we set \( w_{ik}^k = 0 \). Figure 6.4 shows the four cases when finding correspondences on input shape.

The second term \( E_C(\vec{\alpha}) \) enforces the root of each reconstructed petals to converge to the same point in 3D space. The subscript \( (k) \) stands for a pre-defined root vertex.
Figure 6.4: Four cases of finding correspondences on input shape. Red points stand for true correspondences, and green points are false correspondences.

index in the morphable shape model. This is a reasonable semantic prior for many types of flowers in real world. Captured from top view, there is always missing data around the root region of each petal, due to the occlusion caused by pistil. Therefore, adding this prior can effectively assist the convergence of petal roots in 3D space, which also contributes to more realistic reconstructed flowers as a whole.

The last term $E_S(\vec{\alpha})$ encodes the similarity of different petals on the same flower. It enforces petal $r$ and $s$ on the same layer having similar shapes, modeled by the Euclidean difference of coefficient vectors $\alpha_r$ and $\alpha_s$. This term can effectively add strong shape priors to petals with severe occlusion, by assuming that it has similar shape with less-occluded petal in the same layer.

The optimization of our cost function $E(\vec{\alpha})$ is subject to two further constraints. In a reconstructed model, multiple petals in overlapped regions should maintain the same 3D spatial relations as in input scans. We therefore induce a constraint that the reconstructed depth of occluded regions should be larger than the depth of the part in another petal that occludes it. By projecting the reconstructed shape to image, we identify the vertices that lie in the occluded region of that petal. From the
segmentation result, we also know which petals are occluding these vertices, as well as their corresponding reconstructed depth values.

The other constraint restricts the reconstructed shape to lie in our training sample spaces. These two constraints are reasonably defined as:

\[(s_i R_i (\bar{S} + B \alpha_i) + t_i)_{(k)}^z \geq d_o \quad \forall k \in O_i \]

\[\mu - b\sigma \leq \alpha_i \leq \mu + b\sigma\]

where \(O_i\) represents the the set of vertex indices in occluded regions of the \(i^{th}\) petal. The superscript \(z\) stands for depth(z-coordinate) value of a vertex. \(d_o\) is the corresponding depth value in the occluding petal. \(\mu\) and \(\sigma\) are the means and standard deviations of the coefficients of training samples, and \(b > 0\) is a constant.

We optimize the cost function \(E(\bar{\alpha})\) iteratively in a coarse-to-fine fashion. In the first stage, we set the weights of inner vertices in \(E_P(\bar{\alpha})\) to 0, namely, \(w_{nb} = 0\), only align the boundary vertices of input and reconstructed shape, and only subject to constraint in Eq. 6.17 After boundary points converges, \(w_{nb}\) is restored in the following optimization process for better fitting inner regions. We find correspondences \(C_i\) on input shape using the 2D projections, given the boundaries are well aligned. Finally we incorporate constraint in Eq. 6.16 to refine the relative depth relations among different petals. Each stage is repeated until convergence, or reaching a maximum number of iterations \(N\). Intermediate results of each stage showing progressively improvement can be seen in Figure 6.5.

Simultaneous optimization of \(E(\bar{\alpha})\) over \(\bar{\alpha}\) and \((s_i, R_i, t_i)\) is non-linear. For sim-
Figure 6.5: Intermediate results of coarse-to-fine petal fitting. a) After initial alignment; b) After boundary alignment; c) After fitting with inner points; 4) After adding relative depth constraints.

For simplicity, we linearly optimize over $\vec{\alpha}$ and $(s_i, R_i, t_i)$ separately in each iteration. We initialize $\vec{\alpha} = 0$, namely, using the mean shape $\bar{S}$ as starting point for all petals. In order to estimate the initial similarity transformation $(s_i^0, R_i^0, t_i^0)$, we use petals with sufficient visibility based on segmentation information to estimate an average target size. Besides, a rough root position is estimated via the optimal convergence point of their principal axes. For each pedal, the mean shape is then attached to the root, then aligned with the corresponding principal axes, and finally scaled to the target size. In each iteration, $(s_i, R_i, t_i)$ are firstly re-estimated before optimizing $E(\vec{\alpha})$ over $\vec{\alpha}$. Algorithm 1 shows the details of our fitting algorithm.

Figure 6.6 shows a challenging case with severe occlusion. Our joint multiple petal fitting algorithm can still successfully reconstruct the complete shape.
Initialization: \( k = 1; \bar{\alpha}^k = 0; \)

\((s_i, R_i, t_i) = (s_i^0, R_i^0, t_i^0); \forall i \in 1, 2, ..., p\)

for \( l = 1:3 \) do

while \( k \leq N \) do

\[ S^k_i = \bar{S} + B\alpha^k_i; \quad \forall i \in 1, 2, ..., p \]

Compute \((s_i, R_i, t_i)\) from \( S^k_i \) to \( i^{th} \) input petal; \( \forall i \in 1, 2, ..., p \)

Find closest points \( C_i \);

Compute a least-square solution of \( \bar{\alpha}^{k+1} \) for minimizing \( E(\bar{\alpha}) \); s.t

condition \( l \)

if \( \|\bar{\alpha}^k - \bar{\alpha}^{k+1}\| \leq \epsilon \) then

break;

else

\[ \bar{\alpha}^k = \bar{\alpha}^{k+1}; \quad k = k + 1; \]

end

end

end

\( \bar{\alpha} = \bar{\alpha}^k; \)

condition 1: \( w_{nb} = 0 \) and Eq. 6.17

condition 2: Eq. 6.17

condition 3: Eq. 6.16 and Eq. 6.17

Algorithm 1: Joint flower petal fitting algorithm

---

Figure 6.6: Example of severely occluded petal reconstruction. From left to right: 1) Scanned petal data with occlusion; 2) Reconstructed petal without texture; 3) Reconstructed petal with texture.
6.4 Texture Mapping

After reconstructing the shape of each petal, we use standard texture mapping to add texture to our parametric flower model. Each petal is textured mapped individually, by projecting vertices in our parametric shape model on 2D images. For occluded regions, we fill in with content from non-occluded petals for synthesizing a complete texture for partially scanned input.

6.5 Experiments

We demonstrate our flower modeling algorithms on two different flower species, Lily and Pansie. The first species has large shape and size variations and the second has severe occlusions. We also demonstrate the scalability of our method for cross-species modeling, by reconstructing a third species, Dasiy, using the morphable model built from Lily. These two species share similar petal shapes but in remarkably different sizes. We use the following parameter settings to generate all results:

\[\{\beta, \lambda, \alpha, \mu, c, \lambda_p, \lambda_c, \lambda_s, w_b, w_{ab}, \tau, b, N, \epsilon\} = \{0.3, 5, -3, 0.2, 2, 1, 1000, 2, 80, 10, 15, 3, 20, 1e^{-3}\}\]

The prior knowledge about Lily obtained from Botany is the two layer structure, each of which consists three petals. Each top layer petal occludes the bottom layer petals on its two sides. There are multiple variations in petal’s shape, size and color across Lily species. Even petals on the same flower have very different shapes across layers. To test the effectiveness of our method, we use a single morphable shape model from Lily species to reconstruct samples of different variations.
Figure 6.7: 3D modeling of Lily species. From left to right: (a) Petal segmentation; (b) Scanned 3D data; (c) Reconstructed model without texture; (d) Reconstructed model with texture; (e)-(f): Model from different views. The orange example (last row) is scaled up for visualization purpose. Zoom-in for better visualization.

Figure 6.7 shows the results of reconstructing four different variations of Lily. Despite noticeable occlusions and shape variations, our method still achieves high quality modeling of flower petals. Especially, we realistically recovers the shapes of occluded petals. It is worth mentioning that the orange Lily is about half the size of the other three variations. Our scale-invariant morphable model can robustly reconstruct the petal shapes by eliminating the size ambiguity and focus on shape variations.

The second species is Pansie which contains five petals on three layers, with top layer (one petal) occluding middle one (two petals), and middle layer occluding the bottom one (two petals). This type of flower is more challenging due to the severe occlusions in bottom layer. Only a tiny part of that layer is visible during scan. In the same way, we use one morphable shape model for this species to reconstruct samples of four different variations. As shown in Figure 6.8, our algorithm reconstructs
Figure 6.8: 3D modeling of Pansie species. From left to right: (a) Petal segmentation; (b) Scanned 3D data; (c) Reconstructed model without texture; (d) Reconstructed model with texture; (e)-(f): Model from different views. Zoom-in for better visualization.

the flower models with high quality. For the two petals in bottom layer with severe occlusions, our method can fit the visible part very well, while maintaining reasonable predictions on invisible parts. In Pansie species, the depth differences between neighboring layers are very small. Such large occlusion regions allow a large degree of freedom when fitting invisible regions, which is prone to violate 3D relative geometries among different parts. Our relative depth constraints for fitting can efficiently avoid this and obtain more realistic reconstructions. The similarity constraints on petals in same layer constrain each other within a reasonable shape range during fitting and finally reach a common good solution.

**Constraint evaluation** To highlight the importance of our proposed constraints, we conduct comparisons with removing one constraint each. Figure 6.9 shows corresponding results. We can see our joint petal fitting scheme with these prior constraints are crucial in obtaining high quality modeling. As marked out in red, without the
Figure 6.9: Constraint importance evaluation. From left to right: a) Scanned 3D data; b) Remove one constraint; c) With all constraints. From top to bottom: Comparisons with removing 1) root convergence constraint ($E_C$); 2) relative depth constraint (Eq. 6.16); 3) similarity constraint ($E_S$). Zoom-in for better visualization.

root convergence constraint $E_C$, roots of individually fitted petals usually cannot converge, making the reconstruction unnatural. This constraint also assists in handling occlusions by preventing petal from shrinking to only visible parts. The relative depth constraint (Eq. 6.16) maintains correct 3D geometry relations among different layers. The similarity constraint $E_S$ can ensure a reasonable reconstructed shape when a petal is under severe occlusion.

We also make a comparison with a leaf fitting method in a state-of-the-art foliage reconstruction work \cite{4}, which use the same morphable shape model for leaf. They use similar framework but without any prior constraints. Figure 6.10 shows
a comparison with their leaf fitting method. We can see that our method with the proposed constraints can better reconstruct occluded regions, recover correct 3D geometry relations of different components and more surface details. The reason that the leaf fitting method in [4] cannot be applied to our flower petal is that petals in our database have significantly larger variations in shape and size, compared to leaves they work on. Without additional constraints, the individually fitted shape has more freedom and is prone to grow or shrink to non-realistic shapes.

**Cross species test**  Last but not the least, to demonstrate the scalability of our method, we conduct a cross species test, using morphable shape model of Lily species to reconstruct flowers from a different species (Daisy). The two species share similar shape, but the size of Lily petal is approximately 20 times larger than Dasiy’s. Reconstructing Dasiy is even more challenging since there are a large number of occlusion regions and relative 3D geometry relations to handle (approximate 20 petals). The
cross species reconstruction result is shown in Figure 6.1. Despite substantial variations, our algorithm can still robustly obtain high quality reconstructions as long as the training and testing species share similar petal shapes.

6.6 Summary

In this chapter we present a framework for 3D modeling of flower petals. Our approach builds a scale-invariant morphable model of flower petal shape from different variations within a species. In our data-driven modeling approach, the key idea is to use domain knowledge from botany study in petal fitting to handle occluded shape and maintain correct 3D spatial relations. We demonstrate our modeling algorithm with high quality reconstructions for various flower species.

A limitation of our current approach is that there are still inaccuracies in finding correspondence among variations of a flower species, which can downgrade the quality of morphable shape model. And the performance of our fitting algorithm is dependent on the segmented boundary of petals. More constraints might need to be incorporated for robust fitting under inaccurate petal boundaries. In addition, an alternative scanning method is to use volumetric imaging techniques (e.g. micro-CT scanners) which can lead to more complete data acquisition and 3D modeling.

A future extension of the work is to model the development process of flowers. By capturing real-world data for multiple flowers in different life stages, a reconstruction framework taking the temporal domain into consideration can be developed to build a 4D spatio-temporal flower development model, which can be used for estimation of flower growth and visualization of flower development process.
Chapter 7 Conclusions and Future Work

In this dissertation, we have focused on how to use 3D depth information to solve some typical computer vision/graphics problems, including semantic segmentation and scene understanding, image enhancements, and 3D modeling. We have presented several novel algorithms for each component. In this chapter, we summarize our technical innovations and suggest areas for future work.

7.1 Innovations

This dissertation has introduced the following four innovations:

- **Improving Stereo Video Matching via User Interaction and Space-Time Propagation.**

In this work, we propose a stereo video matching system that allows user interaction to obtain high quality, dense disparity maps on key frames and then intelligently propagates the user input and key frame disparities to automatically produce high quality disparity maps on intermediate frames. The disparity maps on key frames are obtained using several novel, easy-to-use, and effective interactive tools. Our novel propagation algorithm estimates 3D transformations that map user corrected areas in key frames to intermediate frames. Experiments demonstrate the effectiveness and efficiency of our hybrid interactive/automatic approach.
• **Using dense 3D depth information for semantic segmentation.**

We investigate how well we can solve semantic segmentation and dense 3D information can perform. Our main innovation is that dense depth maps generated by stereo contain plenty 3D information for scene parsing and can outperform segmentation using sparse 3D features or appearance features. Our 3D metric representation is independent of lighting changes or camera configurations.

• **Using Internet photo collections for personal photo enhancement leveraging 3D depth information.**

We present a framework to approach personal photograph enhancement from a novel direction - using IPCs. The bridge for connecting PPCs and IPCs is the 3D depth information of background landmark reconstructed from IPCs. Augmenting personal 2D images with 3D information, the strong available scene prior allows us to address a number of traditionally challenging image enhancement techniques and achieve high-quality results using simple and robust algorithms.

• **Using 3D depth information with botany priors for high quality 3D flower modeling.**

We propose a framework for parametric modeling of flower petals. To our knowledge, our system is the first to focus on flower modeling, petals in particular, from 3D point cloud. The key contributions of our work can be summarized as: 1) a novel petal fitting algorithm that is robust to significant occlusions; 2) a robust scheme for flower petal segmentation, by extending a two-region level-set formulation to multiple regions; 3) a scale-invariant morphable petal shape
model which can handle wider shape variations within a species, or even across species. It should be emphasized that our reconstruction pipeline generates a parametric model, which is particularly suited for measurement, editing, and animation. We believe our approach will enable more research in the modeling and animation of an intrinsic class of objects, flowers, with applications in botany, entertainment, and visual simulations.

7.2 Future Work

At the end of each previous chapter (Chapters 3, 4, 5, and 6), we list summaries of our proposed methods, and discuss some limitations about each work. In this section, we propose a few more ambitious research topics and share our impressions of future trends in depth-assisted solutions in computer vision.

The flexible and accurate acquisition/generation of 3D depth information of real-world environment has been a long-term goal in computer vision community. Many elements in 3D content generation algorithms have, in many ways, matured. For instance, popular industrial solutions have emerged like Microsoft Kinect, TOF camera, Intel RealSense 3D Camera. While high-quality 3D depth information can be obtained with those systems, the ways computer use 3D visual cues to navigate and understand the world are still far from fully explored. One important future direction worth to explore is to advance the usage of 3D computer vision for augmented reality (AR). AR is an exponentially growing area, which allows to place virtual object or superimpose virtual information on real world. The potential application domains of AR are vast, including medical, education and training, virtual try-on for online
business, manufacturing and repair, gaming and entertainment and etc. The enabling
technology that effectively allows augmenting our reality is the computer vision, es-
pecially 3D related technology. This is because the main cue for current AR systems
is the artificial vision, therefore computer vision for AR has gained increasing im-
portance in the AR context. Although a number of important problems have been
faced in AR, there still remain many challenges that prevent the implementation of
mature AR applications. Specifically, as online merchandize has been spreading all
over the world, virtual try-on technique becomes a promising approach to bridge the
gap between online shopping and offline try-on. However, even the current state-of-
the-art techniques still struggle with either the lack of photorealism or tedious manual
work. Looking into the near future, how to use 3D depth information to automate
the process of virtual try-on and achieve photo-realistic results is, in our view, a very
promising direction.
Bibliography


[98] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. 


VITA

NAME
Chenxi Zhang

PLACE OF BIRTH
Shenyang, People’s Republic of China

EDUCATION

• June 2007: B.S. in Information Engineering, Zhejiang University, Hangzhou, China

SELECTED PUBLICATIONS


• Chenxi Zhang, Mao Ye, Bo Fu, Ruigang Yang, Data-driven Flower Petal Modeling with Botany Priors, In Proc. of the IEEE Intl. Conf. on Computer Vision and Pattern Recognition, June 2014.


