Département de formation doctorale en informatique **École doctorale IAEM Lorraine**
UFR STMIA

Multi-View Oriented 3D Data Processing

Multi-View Orientée 3 D Traitement des Données

THÈSE

présentée et soutenue publiquement le 14 Decembre 2015
pour l'obtention du

Doctorat de l'université de Lorraine

(Spécialité informatique)

par

Kun Liu

Composition du Jury:
Rapporteurs: *Dr. Jean-Michel Dischler, Professeur, Université de Strasbourg, Strasbourg - France*
Dre. Raphaëlle Chaine, Professeur, Université de Lyon, Lyon - France
Examinateurs: *Dr. Rhaleb Zayer, Chargé de Recherche, INRIA, Nancy - France (maintenant en disponibilité chercheur Max Planck Institute fur Informatik Saarebruck)*
Dr. Bruno Lévy, Directeur de Recherche, INRIA, Nancy - France
Dr. Jean-Sébastien Franco, Maitre de conférence Informatique, ENSIMAG Grenoble, Grenoble - France
Dr. Sylvain Contassot-Vivier, Professeur, LORIA, Nancy - France

INRIA Nancy Grand Est
Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA)
UMR 7503 - Campus Scientifique - BP 239 - 54506 Vandoeuvre-lès-Nancy Cedex

Dedicated to my parents.

RÉSUMÉ

La richesse visuelle des données en trois dimensions (3D) et la possibilité de l'interaction et de l'exploration que ces données proposent en font un choix primal dans un large éventail de domaines couvrant la fabrication, la médecine, et le divertissement numérique. Cette popularité continue sa croissance grâce à l'avancement dans la technologie d'acquisition de données, à l'augmentation énorme de stockage modernes, et à l'amélioration des capacités de traitement de dispositifs informatiques modernes.

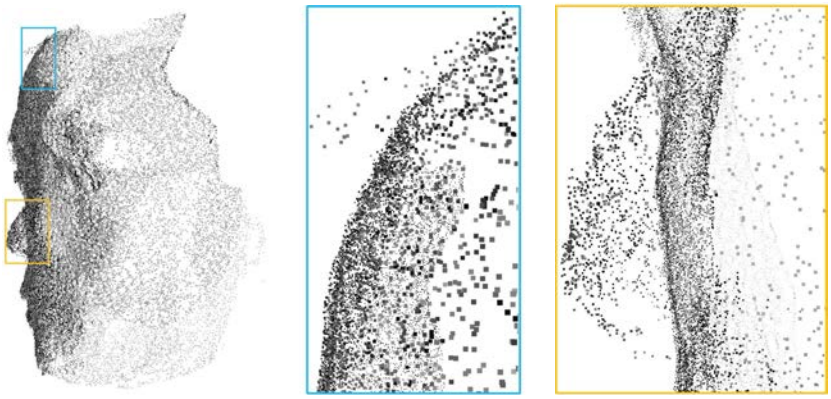
Traditionnellement, les données 3D peuvent être acquises en utilisant de nombreuses techniques différentes. Par exemple, les données 3D peuvent être modélisées dans un logiciel de conception assistée par ordinateur (CAO) ou être directement collectées en utilisant la mesure lidar dans un systéme d'information géographique (SIG). Plus récemment, la reconstruction stéréoscopique multi-vue a gagné en popularité grâce à sa facilité d'utilisation et à son faible coût. La méthode nécessite seulement un ensemble d'images de différents points de vue pour extraire l'information de profondeur manquante. En raison du développement important de caméras modernes, une acquisition de haute résolution et de haute précision peut être effectuée. Des projets populaires et interdisciplinaires entre l'infographie et la vision par ordinateur comme le projet "Building Rome in a Day" [3] dans lequel des milliers d'images recueillies à partir du Web sont utilisées pour créer des modèles de villes en 3D, démontrent davantage la flexibilité et l'évolutivité de cette approche.

Malgré de grands progrès dans la reconstruction multi-vue, la nature des données acquises pose encore de grands défis. En général, les données 3D brutes sont représentées sous forme d'un nuage de points et nécessitent des opérations de traitement spécifiques avant l'utilisation pratique. Alors que de nombreuses techniques de traitement de la géométrie existantes peuvent être directement appliquées à ces données, normalement elles ne permettent pas de produire de bons résultats car ces méthodes sont souvent inconscientes de la nature intrinsèque de ces données. En effet, historiquement la plupart de ces méthodes ont été conçues pour les données de scanner laser. Les données multi-vues ont ses particularités comme l'échantillonnage très irrégulier, le bruit multi-couche et les valeurs aberrantes; plus important, il est souhaitable de maintenir les correspondances entre la scène et les images tout au long des opérations de traitement.

Le travail effectué dans cette thèse vise à concevoir de nouvelles méthodes de traitement de données 3D qui tiennent compte de la nature spécifique de données multi-vues. L'accent sera mis sur deux

problèmes majeurs: i) le raffinement de nuage de points pour améliorer la qualité des données; et ii) la reconstruction de surface pour obtenir une représentation de maillage viable.

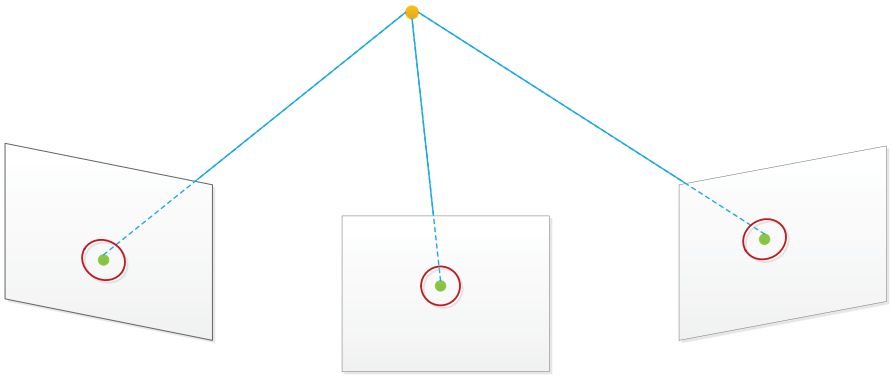
Le raffinement de nuage de points est l'un des problèmes les plus importants dans le traitement de données 3D. Il améliore la qualité des données en éliminant les artefacts tels que les valeurs aberrantes et le bruit. Comme le montrent les figures ci-dessous, celle de gauche affiche un nuage de points brute généré par la reconstruction multi-vue; celle du milieu et celle de droite sont deux vues rapprochées qui mettent en évidence respectivement les valeurs aberrantes et le bruit.



En général, les valeurs aberrantes sont définies comme des observations avec de grandes erreurs aléatoires, et le bruit est la différence entre les données acquises et les données réelles. Dans la pratique, les valeurs aberrantes ont des erreurs plus grandes et sont plus faciles à identifier que le bruit. Par conséquent, le filtrage des valeurs aberrantes se présente généralement moins de difficultés que le filtrage de bruit. Dans cette thése, l'accent sera mis principalement sur le filtrage de bruit.

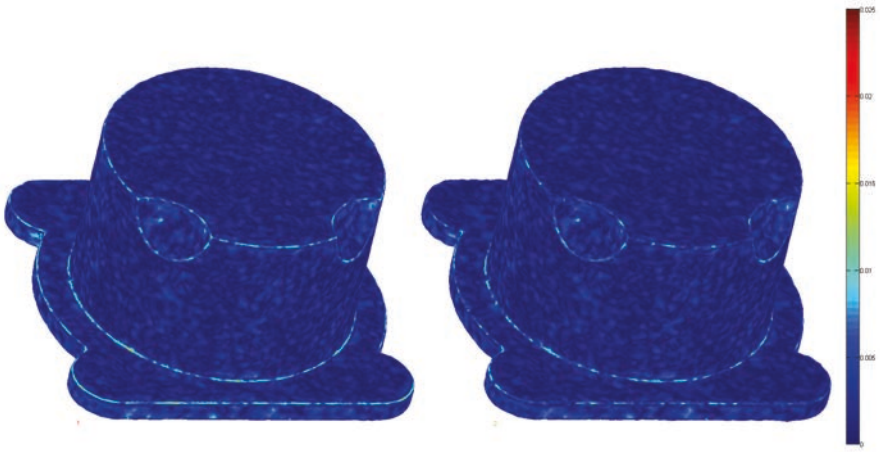
Bien qu'une grande quantité de méthodes sur le filtrage de bruit ont été développées pour les données de points 3D, par exemple [7, 141, 158], la plupart de ces méthodes ne fonctionnent pas bien sur les données provenant de la reconstruction multi-vue. Ces méthodes sont principalement destinées aux données acquises par des capteurs de distance et fonctionnent principalement dans un espace tridimensionnel. Par conséquent, utiliser directement ce type de méthodes sur des données multi-vues peut nuire gravement la correspondance entre la scène et les images. Cela peut être un défaut important dans les applications qui nécessitent des positions de suivi précises, par exemple, la photogrammétrie.

Dans cette thèse, une nouvelle méthode de filtrage de bruit sans endommager la correspondance entre la scène et les images est proposée. Comme illustré dans la figure ci-dessous, à partir d'un ajuste-



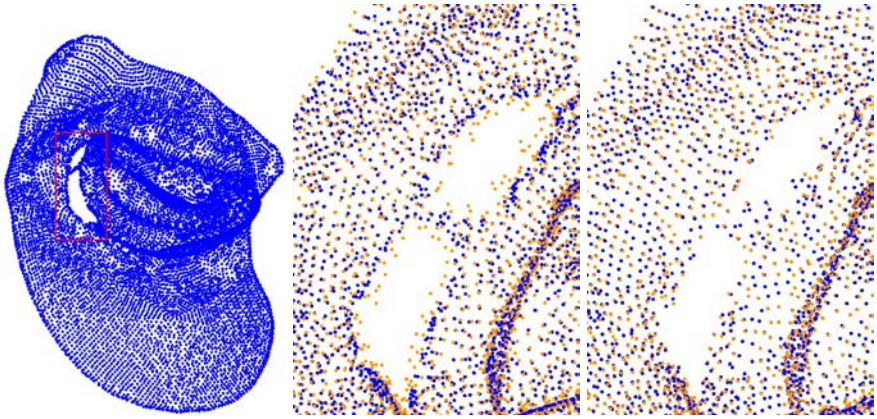
ment de faisceaux convergent, l'approche cherche la nouvelle position spatiale du point 3D tout en garantissant que l'erreur de reprojection [78] soit bornée, c'est-à-dire, les points en correspondance soient maintenus dans un disque autour des points initiaux. Mathématiquement, le problème est formulé comme une optimisation sous contrainte, c'est-à-dire, une minimisation d'une mesure de douceur avec une contrainte de borne sur l'erreur de correspondance dans les données multi-vues. Pour la mesure de douceur, une mesure qui favorise la planéité locale des données de points 3D est adoptée ; pour la contrainte, l'erreur de reprojection présente dans l'ajustement de faisceaux [78] est utilisée.

Afin de résoudre le problème de l'optimisation non-linéaire avec contrainte que nous formulons, nous nous reposons sur la méthode de barrière qui permet d'approximer un problème d'optimisation avec contrainte par une série de problèmes sans contrainte. La so-



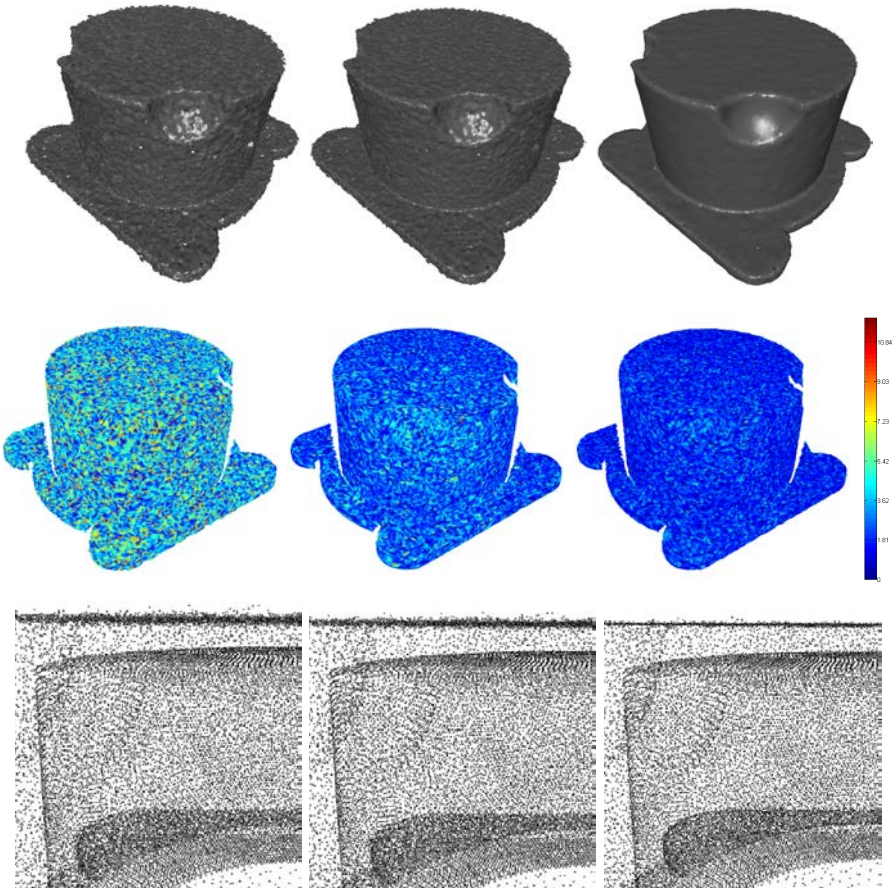
lution optimale de notre formulation est un nuage de points lisse

où l'erreur de reprojection est délimitée par un seuil prescrit. Comme les matrices creuses résultant de cette formulation ont tendance à être densément peuplées, nous profitons de la formule Sherman-Morrison à aborder le problème numérique sous-jacent. Cela permet la méthode de fonctionner sur un grand ensemble de données tout en conservant une empreinte mémoire raisonnable.



Un aspect clé de filtrage du bruit est la discrimination entre le bruit et les caractéristiques géométriques détaillées. L'absence d'une telle discrimination conduit souvent au problème de sur-lissage. Afin d'éviter ce problème, une norme solide est incorporée dans la définition de la mesure de douceur. Comme représenté dans les figures, celle de gauche affiche le résultat de la méthode proposée sur des données d'entrée bruitées en utilisant la norme L^2 et celle de droite indique l'effet de l'utilisation de la norme solide. Le code couleur indique la distance de Hausdorff par rapport au nuage de points de vérité terrain. En outre, compte tenu de la mesure de douceur adoptée, la méthode proposée peut surmonter les effets de "rétrécissement" qui est en effet un défaut important de certaines méthodes telles que le lissage Laplacien. Cela nous permet de produire des résultats corrects en particulier autour des trous et des zones où les données sont manquantes, ce qui sont des cas couramment rencontrés dans les données multi-vues. Comme nous le voyons sur les figures, un zoom sur le modèle de l'oreille (à gauche) illustre l'effet de rétrécissement de la régularisation Laplacienne (voir le nuage de points en bleu dans la figure du milieu). Le lissage avec contrainte que nous proposons est plus robuste à ces artefacts (voir le nuage de points en bleu dans la figure de droite). Dans les deux figures (au milieu et à droite), les données d'origine sont affichées en orange.

La méthode proposée est testée sur un ensemble de données synthétiques et du monde réel. Pour les données synthétiques, toutes les données de vérité terrain nécessaires sont disponibles pour l'évaluation. L'ensemble des neuf figures montre une comparaison entre l'ajustement de faisceaux fondé sur la régularisation Laplacienne et l'ajustement de faisceaux fondé sur le lissage avec contrainte que nous proposons,



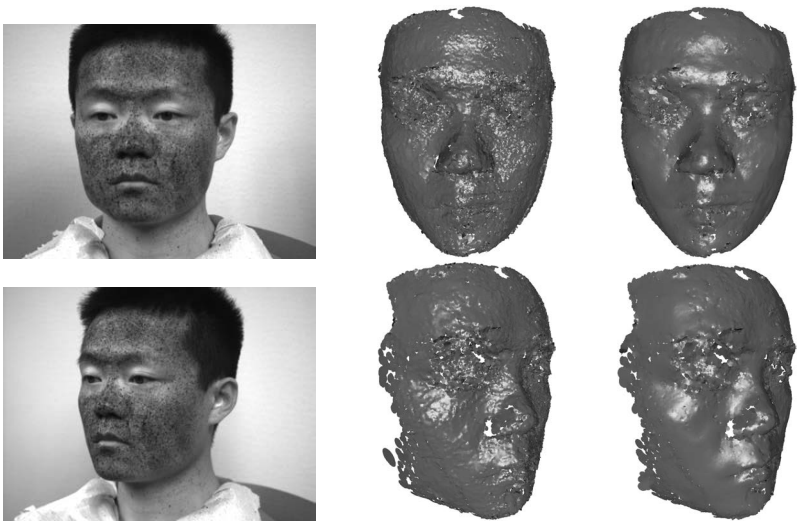
en utilisant des données synthétiques. Un nuage de points bruités (en haut à gauche) est traité en utilisant la première méthode (en haut au milieu) et la deuxième méthode (en haut à droite), tous les points de vue sont présentés en mode “splatting”. Les figures dans la deuxiéme ligne montrent les cartes de l’erreur de reprojection pour le même point de vue dans lesquelles les couleurs plus froides indiquent les erreurs plus petites. Les figures dans la ligne en bas montrent une vue rapprochée sur le nuage de points correspondant. Les deux exemples d’images (la première colonne ci-dessous) sur un ensemble de six grandes images de base ont été utilisés pour générer un nuage de points quasi-dense (la deuxième colonne), en utilisant l’approche de propagation dans [104]. Le résultat de la méthode proposée (troisième colonne) montre une amélioration de la qualité globale du nuage de points. Les nuages de points sont également affichés en mode “splatting”.

Reconstruction de surface vise à construire une représentation de maillage à partir des données de nuages de points. C’est un domaine de recherche important dans le traitement de la géométrie en raison de l’importance des maillages dans de nombreuses applications. Dans la visualisation scientifique par exemple, les maillages sont généralement utilisés pour visualiser de grands ensembles de

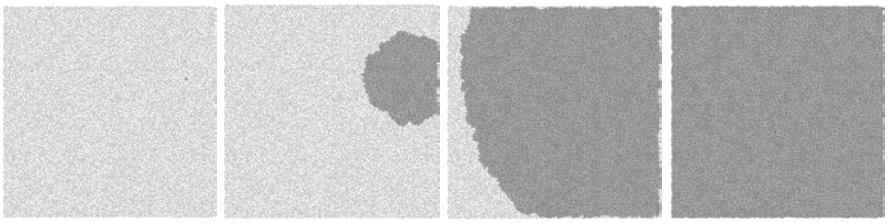
données 3D, puisque la représentation de maillage est beaucoup plus compacte qu'un nuage de points, et donc plus efficace en terme d'utilisation de la mémoire. En outre, la représentation de maillage peut également se intégrer harmonieusement à l'interface de programmation de rendu existants, comme OpenGL [92] et Direct3D [40]. De plus, les maillages peuvent faciliter le calcul des attributs géométriques tels que les courbures et les normales qui sont essentielles pour d'autres problèmes de traitement des données tels que la segmentation et la classification.

La majorité des méthodes classiques existantes de reconstruction de surface ont été conçues pour des données acquises par capteurs de distance. Cependant, le développement rapide de la reconstruction multi-vue pose de nouveaux défis. Les nuages de points provenant de la reconstruction multi-vue présentent certaines particularités en ce qui concerne des aspects tels que la densité, la distribution et la précision comme illustrés dans la figure ci-dessous. Un nuage de points est généré par la reconstruction stéréoscopique multi-vue [155]. Les rectangles bleu, vert et jaune soulignent respectivement la nature multi-couche des données, l'échantillonnage irrégulier et les valeurs aberrantes. Les méthodes classiques sont souvent mal adaptées à l'échantillonnage très irrégulier et la nature multi-couche des données multi-vues. D'autres opérations de poste-traitement, par exemple, lissage de maillages, simplification de maillages et remaillage, sont souvent nécessaires pour affiner les maillages obtenus afin de les rendre utilisables dans des applications pratiques. Par conséquent, un procédé de reconstruction de surface à usage général reste un objectif très difficile.

Afin de résoudre les problèmes mentionnés ci-dessus, une nouvelles méthode de reconstruction de maillage est développée dans cette thèse. En général, une reconstruction de surface satisfaisante se caractérise par de bonnes propriétés telles que la qualité de maillages

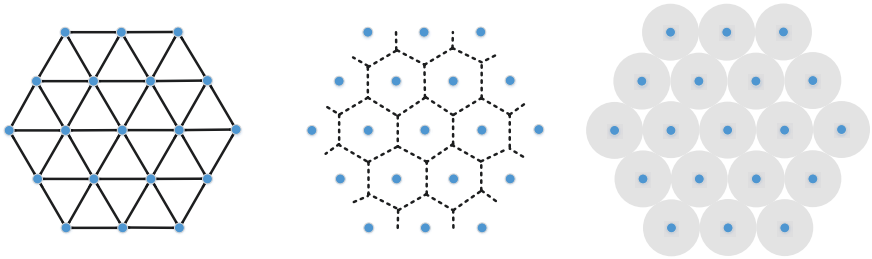


généré, un faible coût de calcul, et la robustesse au bruit. Cependant, ces propriétés souhaitées sont difficiles à obtenir en même temps dans une seule méthode. Comme observé dans les études de la littérature de reconstruction de surface [140, 5], la plupart des méthodes existantes souffrent de certaines limitations. Par exemple, en général les méthodes classiques à base de géométrie algorithmique ne peuvent pas gérer directement le bruit. D'autre part, les méthodes fondées sur la représentation de surface implicite créent inévitablement un grand nombre de triangles maigres.



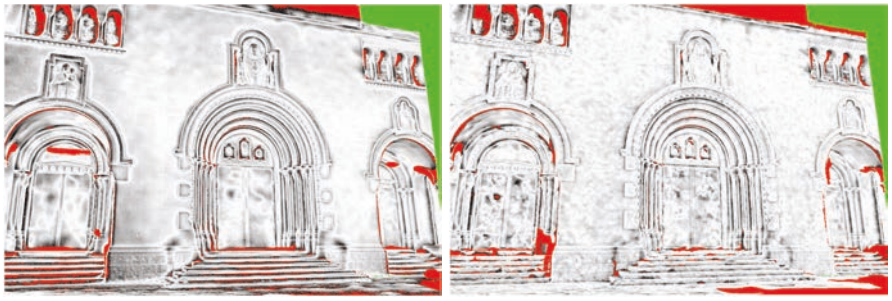
Dans cette thèse, la méthode proposée fonctionne localement en utilisant une stratégie de l'avancement du front pilotée par le principe global de l'empilement compact de sphères. La procédure de la reconstruction est illustrée dans la figure ci-dessus. Un nuage de points (points gris) représentant un carré est triangulé en utilisant la méthode proposée. Les figures de gauche à droite respectivement illustrent l'initialisation du triangle germe, les résultats intermédiaires reconstruits, et le maillage résultant final.

L'empilement compact de sphères est adopté sur la base de l'observation que le modèle de sphères empilées imite un pavage de Voronoï qui facilite la production de maillages de haute qualité. Comme nous le voyons dans les figures suivantes, celle de gauche montre une triangulation de Delaunay d'un ensemble de points (bleu). Les lignes pointillées dans la figure du milieu représentent un diagramme de



Voronoï qui est dual de la triangulation de Delaunay. La figure de droite illustre les sphères empilées (gris) dont la configuration imite le diagramme de Voronoï. La nature de la stratégie de l'avancement du front assure l'efficacité requise de la méthode proposée, tout en gardant que les algorithmes sous-jacents sont simples et élégants. En outre, tout au long de la procédure de l'avancement du front, une méthode modifiée de moindres carrés mobiles [7] est appliquée pour éliminer l'impact du bruit dans les données originales. En raison de la combinaison de l'avancement du front, l'empilement compact de sphères et la méthode de moindres carrés mobiles, l'algorithme proposé dispose de plusieurs propriétés désirées. En outre, étant donné que l'algorithme est basé sur l'avancement du front, il ne fonctionne que localement et donc il n'est pas nécessaire de charger toutes les données de points dans la mémoire simultanément. Cela facilite la triangulation d'un énorme nuage de points avec l'aide de la technique de "streaming" [24]. Pour la même raison, la méthode proposée peut être facilement parallélisée, puisque les triangles lointains sont créés indépendamment lors de la reconstruction de maillage.

(a) BP (b) PR



(c) YO (d) Ours

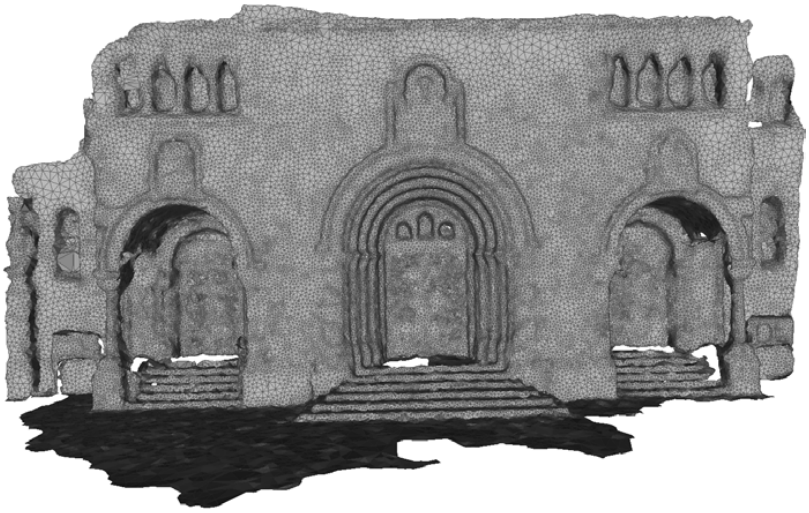
Visiblement, la méthode proposée utilise seulement des sommets du nuage de points d'entrée comme les sommets du maillage reconstruit (dans la limite d'une correction de position bornée). Cette caractéristique peut éviter implicitement d'introduire des erreurs de reprojection supplémentaires aux images dans les données multi-vues, et donc la méthode peut bénéficier à quelques applications telles que la photogrammétrie et le suivi des objets en vision par ordinateur.

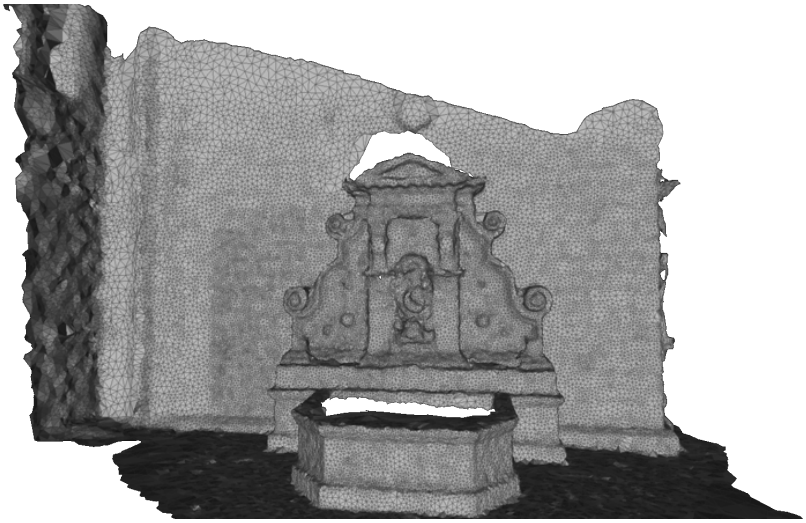
La méthode proposée est évaluée sur des données de nuages de points acquis par la reconstruction stéréoscopique multi-vue ainsi que sur autres données de points, afin de montrer la généralité de notre méthode. Pour les données multi-vues, nous effectuons des tests sur le banc d'essai proposé dans [155]. Les résultats de reconstruction de BP, PR, et YO respectivement dans [20, 128, 91], ainsi que le résultat de la méthode proposée, sont évalués selon la méthode décrite dans [155]. Les images ci-dessus montrent la carte de différence de profondeur pondérée par la variance. Les pixels rouges représentent les erreurs qui sont plus grandes que le seuil. Les pixels verts représentent les données de numérisation manquantes de la vérité terrain. Les erreurs relatives sont affichées en utilisant niveaux de gris, et des couleurs plus claires indiquent des erreurs plus petites. Les résultats dans les figures ci-dessous sont présentés pour les données Herz-Jesu et les données fontaine-P11.

Le travail présenté dans cette thèse se traduit dans les contributions publiées comme suit:

- Kun Liu, Rhaleb Zayer, Bundle Adjustment Constrained Smoothing for Multi-view Point Cloud Data, ISVC 2012, 8th International Symposium on Visual Computing, LNCS Volume 7431, pp 126-137, 2012.

- Kun Liu, Patricio A. Galindo and Rhaleb Zayer, Sphere Packing Aided Surface Reconstruction for Multi-View Data, ISVC 2014, 10th International Symposium on Visual Computing, LNCS Volume 8888, pp 173-184, 2014.





ABSTRACT

Point cloud refinement and surface reconstruction have been well studied and are widely applied in various applications, e.g., computer-aided design (CAD), geographic information science (GIS), and scientific visualization. However, most of the existing methods are originally designed for range sensor data. They have turned out be ill-adapted to the data generated by multi-view reconstruction, which has been one of the most popular acquisition methods, due to new challenges in such data, e.g., multilayer and high irregular sampling. Moreover, these methods are often oblivious to the essential nature in multi-view data such as the correspondence between the scene and images.

In this thesis, on account of the gap left by the previous methods, the problems of point cloud refinement and surface reconstruction are studied with special attention to multi-view data. Two novel methods are proposed and contribute to the interdisciplinary research of geometry processing and multi-view reconstruction. The first contribution is a novel smoothing method which improves the quality of point clouds originating from multi-view reconstruction without impairing the data. In the method, the smoothing problem is formulated as a nonlinear constrained optimization, concretely, minimizing a smoothness measure with a bound on the reprojection error. The optimization is addressed taking advantage of a barrier method which addresses the constrained optimization problem by solving a series of unconstrained optimization problems. In the numerical solving procedure, the arising challenge relating to densely populated matrix is resolved by means of the Sherman-Morrison formula. In contrast to the previous purely geometry-based methods, the proposed method combines the geometry assumption and the reprojection error, therefore, it is able to produce smooth results while still preserving the correspondence between the scene and images in multi-view data. The second major contribution is a surface reconstruction technique tailored for multi-view data. This technique triangulates a point cloud into a mesh using an advancing front strategy steered by a sphere packing criterion. Such strategy results in reconstructing meshes locally, which makes the method fairly suitable for parallel computing as distant triangles are always created independently. The adopted criterion prevents tedious optimization usually required by many previous approaches but still can give rise to high-quality meshes, because tightly packed spheres mimics a Voronoi tessellation which facilitates generating well-shaped triangles. Sphere packing also helps to address the issues of multilayer and irregular sampling in multi-

view data. Furthermore, a modified moving least square procedure is integrated into the advancing front framework to make the method robust to noise.

The two developed methods in this thesis are implemented completely and evaluated empirically on both synthetic and real-world data. The experimental results demonstrate the robustness and the efficiency of the methods. Taking advantage of the synthetic data, the experiment with respect to the smoothing method shows that noise is eliminated and even the reprojection error decreases surprisingly, namely, the method enhances the visual appearance and also improves the matches in multi-view data. Hence, the method can be regarded as a post-processing tool for multi-view reconstruction and can benefit applications which require accurate and consistent position information such as tracking in computer vision. Moreover, the method successfully resolves the over-smoothing problem and the shrinkage around where data is missing such as holes, which outperforms Laplacian regularization. The second contribution – the sphere packing aided surface reconstruction method – has also been evaluated and compared with several other popular approaches in the area. The comparison indicates the resulting meshes by the method have more triangles close to equilateral and fewer skinny ones, which suggests the method outperforms the others in such scenario. Most notably, the proposed method creates triangles only using points originating from the input point cloud. By using the benchmark, it demonstrates that such property causes a more accurate approximation of the original geometry. This suggests the method can be an ideal solution for applications which require high measurement accuracy such as photogrammetry.

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my research advisors, Dr. Rhaleb Zayer and Dr. Bruno Lévy, for their patient guidance, valuable enlightenment, and continuous encouragement during the past years. The thesis would not have happened without their insightful comments and constructive critiques.

I would like to thank Patricio Alejandro Galindo, Anass Lasram, Dongming Yan, Faqing Yang, Kai Wang, Kanle Shi, Lian Shi, Sai Qian, Samuel Hornus, and Shizhe Zhou. I am honored and lucky to meet you in such an extraordinary research institute to study and work together. The time in INRIA is a remarkable period which will permanently remain in my memory.

I wish to extend my thanks to Ana Carola Vacaflores, Barbara Menin, Boyi Ni, Chongyang Ma, Fengwei Chen, Jiaojiao Han, Jingjiao Liu, Jingwen Yang, Marianela Parodi, Shuguo Zhuo , Xiaoyun You, Xiyan He, Yan Lv, Yong Xu, Yongxin Liao, and Yushu Chen. I am grateful to spend a lot of beautiful and unforgettable moments with you in Nancy.

The sincere thanks to my old friends in Hangzhou – Binbin Chen, Congjian Jin, Guo Li, Haibin Huang, Hanlin Zheng, Lei Zhang, Li Bao, Ligang Liu, Lubin Fan, Min Meng, Qi Lou, Renjie Chen, Xiaokun Wu, Xiao Wang, Xiaoqun Wu, and Zhonggui Chen. I deeply appreciate the cooperative and supportive discussions for the research work.

I am also indebted to the colleagues and friends in London – Annabel Han, Charles Thomson, Congcong Peng, David Wong, Dietmar Backes, Hui Yang, Jan Boehm, Jungian Chen, Lei Wang, Mona Hess, Rui Sha, Sarah Wise, Shuai Li, Wei Shangguan, Ying Li, and Ziyi Jiang. The time in London is great because of your company.

Last but not least, I would like to give special thanks to my grandmother and my parents, for their love throughout my life. Words cannot fully express how thankful I am for their forever and unconditional support. Your support has been and will be always my greatest source of motivation, encouragement, and empowerment.

CONTENTS

1 INTRODUCTION

The visual richness of three-dimensional (3D) data and the wide interaction and exploration possibilities it offers make it a primal choice across a wide range of fields spanning manufacturing, medicine, and entertainment. This popularity continues to grow driven by the surge in data acquisition technology and the tremendous increase in modern storage, and processing capabilities of modern computing devices.

Traditionally, 3D data can be acquired using many different techniques, for instance, it can be modeled in software as in computer-aided design (CAD) or directly gathered using lidar measurement as in geographic information science (GIS). Most recently, multi-view stereo reconstruction has gained popularity thanks to its minimal requirements and low cost. The method requires only a set of photographs from different views to extract the missing depth information. Owing to the significant development of modern cameras, high resolution and accuracy acquisition can be performed. Popular projects on the crossroads between computer graphics and computer vision such as "Building Rome in a Day" project [3], where thousands of images collected from the Web are used to create 3D city models, further demonstrate the flexibility and scalability of the approach.

Despite great strides in multi-view reconstruction, the nature of the acquired data still poses great challenges. In general, the raw 3D data is represented as a point cloud and requires further processing operations before practical use. While many existing geometry processing techniques can be directly applied to such data, they fall short from producing compelling results as they are often oblivious to the intrinsic nature of such data. In fact, historically most of these methods have been tailored for laser scan data. Multi-view data on the other hand can exhibit highly irregular sampling, multilayered noise and outliers and more importantly, it is desirable to maintain the correspondences between the scene and images throughout the processing operations.

The work carried out throughout this thesis aims to tailor novel 3D data processing which take into account the specific nature of multi-view data. The focus will be put on two major problems: i) point cloud refinement for improving the quality of the data and ii) surface reconstruction for building a viable mesh representation.

Point cloud refinement

Point cloud refinement is one of the most important problems in 3D data processing. It improves data quality by eliminating artifacts such as *outlier* and *noise*. In general, outliers are defined as those observations with large random errors, and noise is the differences between acquired data and true data. In practice, outliers are more erroneous and easier to identify than noise. Thus outlier filtering usually presents fewer difficulties than noise filtering. In this thesis, the focus will be set mainly on noise filtering.

Although a large amount of methods on noise filtering have been developed for 3D point data, e.g., [7, 141, 158], most of them do not perform well on the data originating from multi-view reconstruction. These methods are primarily targeted at data acquired by range sensors and mostly operate in three-dimensional space. Therefore, using them directly on multi-view data can seriously impair correspondences between the scene and images. This can be a significant setback in applications which require accurate tracking positions, e.g., photogrammetry.

In this thesis, a novel method for filtering noise without forfeiting correspondences between the scene and images is proposed. The problem is formulated as a constrained optimization mathematically, i.e., a minimization of a smoothness measure constrained by a bound on the correspondences error in the multi-view data. For the former, a measure which favors local flatness of point data is used and for the latter, the *reprojection error* presented in bundle adjustment [78] is used. In order to address the arising nonlinear constrained optimization we rely on the barrier method, which is a procedure for approximating a constrained optimization problems by an unconstrained one. The optimum sought in our formulation is a smooth point cloud where the reprojection error is bounded by a prescribed threshold. Since this formulation results in a densely populated Hessian matrix (see Equation 63 in Chapter 4), we take advantage of the Sherman-Morrison formula to address the numerical issue. This allows the method to operate on large data sets while maintaining a reasonable memory footprint.

A key aspect during noise filtering is the discrimination between noise and detailed geometry features. The lack of such discrimination often leads to *over-smoothing*. In order to prevent this problem, a robust norm is incorporated in the definition of the smoothness measure. Moreover, in view of the adopted smoothness measure, the proposed method can overcome shrinkage effects which commonly burdens other methods such as Laplacian-based smoothing. This helps produce coherent results especially around holes and areas where data is missing, which are commonly encountered in multi-view data.

Surface reconstruction

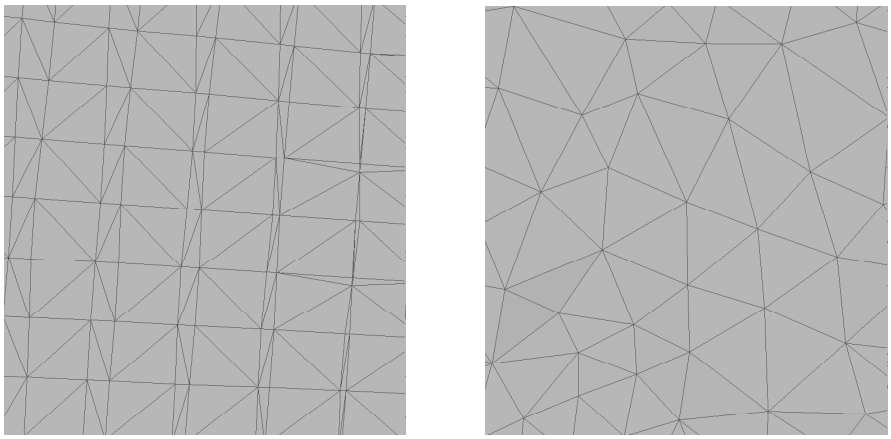
Surface reconstruction aims to build a mesh representation from a point cloud data. It is a major research area in geometry processing on account of the importance of meshes to many applications. In scientific visualization, for example, meshes are usually used to render large 3D data sets as the mesh representation is far more compact than a point cloud, and hence more memory efficient. In addition, the mesh representation can also seamlessly blend into the existing rendering APIs, such as OpenGL [92] and Direct3D [40]. Moreover, meshes can facilitate computing geometric attributes such as normals and curvatures which are essential for other data processing problems such as segmentation and classification.

The majority of existing classical surface reconstruction methods have been designed for data acquired by range sensors. However, the rapid development of multi-view reconstruction poses new challenges. Point clouds originating from multi-view reconstruction exhibit certain particularities with regards to aspects such as density, distribution, and accuracy as illustrated in Figure 8. Consequently, classical methods are often ill adapted to the highly irregular sampling and multilayered nature of multi-view data. Further post-processing operations, e.g., mesh smoothing, mesh simplification, and remeshing, are often required to refine the resulting meshes in order to make them usable in practical applications. Therefore, an all-purpose surface reconstruction method remains a very challenging goal.

Figure 8: A point cloud is generated by multi-view stereo reconstruction [155]. The blue, green and yellow rectangles highlight multilayers, irregular sampling and outliers respectively.

In order to deal with the aforementioned problems a novel mesh reconstruction methods is developed in this thesis. In general, a sat-

isfactory surface reconstruction is characterized by some properties such as high quality of resulting meshes, low computational cost, and robustness to noise. High quality of meshes often means that the mesh triangles are as equilateral as possible (see Figure 9b). However, these favorable properties are difficult to combine together into a single method. As observed in surveys on surface reconstruction literature [140, 5], many of the existing methods suffer from certain limitations. For instance, classical computational geometry based methods generally cannot handle noise directly. On the other hand, implicit surface methods inevitably create a large number of skinny triangles as shown in Figure 9a. In this thesis, the proposed method works locally using an advancing front strategy directed by the global principle of sphere packing. Sphere packing is adopted based on the observation that the pattern of tightly packed spheres mimics a Voronoi tessellation which facilitates producing high quality meshes. In addition, the nature of the advancing front strategy endows the method with the required efficiency while keeping the underlying algorithms simple and elegant. Moreover, throughout the advancing front procedure a modified *moving least square* [7] method is applied to curb down the impact of noise in the original data. As a result of the combined trio of advancing front, sphere packing, and moving least squares, the proposed method enjoys several desired properties.



(a) (b)

Figure 9: (a) displays a mesh with many skinny and flat triangles. (b) shows a mesh with triangles much closer to equilateral triangles.

Noticeably, the proposed method only uses points from the input point cloud as vertices of the reconstructed meshes (up to a bounded position correction). This feature can implicitly prevent introducing additional reprojection errors across images in multi-view data, and thus the method can benefit a few applications such as photogrammetry and tracking in computer vision. Furthermore, since the method is based on advancing front, it operates only locally and therefore it is not necessary to load the entire point data into the memory simul-

taneously. This facilitates triangulating a huge point cloud with the help of streaming technique [24]. For the same reason, the proposed method can be easily parallelized as distant triangles are created independently while reconstructing a mesh.

The rest of this thesis is organized as follows. First, the mainstream 3D acquisition techniques are reviewed in Chapter 2. In Chapter 3, the problem of point cloud refinement is studied, and a novel point cloud smoothing method tailored for multi-view data is presented to filter noise in Chapter 4. Existing surface reconstruction methods are investigated in Chapter 5, and both the advantages and the disadvantages of the methods are discussed. In Chapter 6, a novel mesh reconstruction method particularly suitable for multi-view data is proposed based on the combination of advancing front and sphere packing. Finally, the thesis is summarized and discussed in Chapter 7.

The work presented in this thesis results in the contributions published as follows:

- Kun Liu, Rhaleb Zayer, Bundle Adjustment Constrained Smoothing for Multi-view Point Cloud Data, ISVC 2012, 8th International Symposium on Visual Computing, LNCS Volume 7431, pp 126-137, 2012.

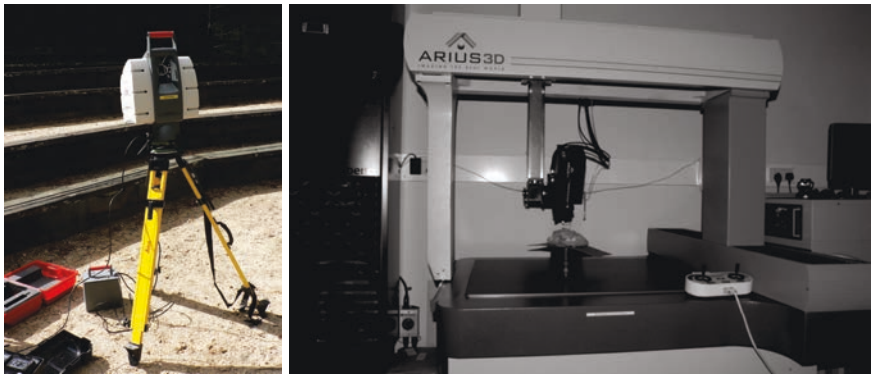
- Kun Liu, Patricio A. Galindo and Rhaleb Zayer, Sphere Packing Aided Surface Reconstruction for Multi-View Data, ISVC 2014, 10th International Symposium on Visual Computing, LNCS Volume 8888, pp 173-184, 2014.

2 3D DATA ACQUISITION

Nowadays 3D data is widely used in various areas. People in geomatics use airborne LiDAR to obtain 3D territorial data. Archaeologists digitize cultural heritages by means of laser scanners. With the help of motion capture techniques, artists record human actions for animated movies. In general, 3D data can be acquired in different ways.

3D data acquisition is a procedure to collect 3D scene geometry. An extensive amount of techniques have been applied for 3D acquisition. In this chapter, the techniques based on range sensors and multi-view stereo are reviewed, especially the fundamental theory in multi-view stereo. The former one is a series of classical methods, whereas the latter one undergoes a significant development over the last two decades. Moreover, both strengthens and weaknesses of the two categories of methods are discussed as well.

2.1 RANGE SENSOR BASED METHODS



(a) (b)

Figure 10: (a) a LiDAR scanner for buildings and rocks scanning; (b) a scanner for culture heritage scanning.

Among a large number of techniques in 3D acquisition, LiDAR [166] is a popular one extensively used in geomatics for surveying. To scan a 3D scene, a LiDAR scanner is used to emit lights towards targets. The distance from the scanner to the target is estimated by measuring the time of the light traverse. In this way, the geometry of the scene is obtained. As shown in Figure 10a, a LiDAR scanner, which is usually used for buildings and rocks scanning, is displayed. The direction of the emitted light is controlled by rotating the head horizontally and rotating the mirror in the head vertically. Hence, theoretically

the scanner can realize 360° scanning. Moreover, it can be used in both indoor and outdoor environments and produces high quality 3D models.

A laser scanner for culture heritage scanning is also shown in Figure 10b. This scanner has been used to digitize collections in museums [135]. It is manipulated by a control panel consisting two sticks (see right-bottom in the figure). The scanner can produce 3D models with high accuracy and resolution, which easily acquire fine geometry details. For instance, the scanner shown in Figure 10b can achieves the depth ability better than 25 micrometers and 100 micrometers of the minimum of spatial sampling interval.

LiDAR scanners can conduct high quality scanning. However, most of them are not portable. For example, the one in Figure 10a is mounted on a tripod and the one in Figure 10b is totally lack of mobility. Moreover, a few LiDAR such as airborne LiDAR requires navigation and positioning systems, e.g., the Global Positioning System (GPS). Furthermore, LiDAR scanners are generally fairly costly and thus they are only commonly applied in a few limited areas.

To acquire data more efficiently, several novel scanning systems have been also proposed recently. A mobile mapping system [162] is illustrated in Figure 11a. The system consists of a spherical camera [73] to capture panorama images and three laser scanners to acquire 3D scene geometry. The entire system is mounted in a trolley which can be moved freely and thus the system is suitable for indoor scanning. The scanner shown in Figure 11b can be even held by hands and operated conveniently [121]. The two new systems can efficiently accomplish scanning with decent quality. However, the high cost is still a crucial problem which obstructs more widespread use.

Noticeably, over last several years RGB-D sensors have become very popular due to the competitive prices and the acquisition qualities. In 2010, the motion sensing device Kinect [41] was first available for the video game console (see Figure 11c), and several similar sensors were also released afterwards, e.g., Asus Xtion Pro Live [15] and PrimeSense Carmine [134]. An RGB-D sensor, which consists of an RGB camera and a depth sensor, is able to simultaneously acquire colors and geometry efficiently. The depth sensor contains an infrared light projector and a monochrome CMOS used to detect the infrared light. As illustrated in Figure 12a, the light path can be determined by analyzing light patterns on objects. Since the distance between the projector and the CMOS is known as well as the directions of the outbound and the inbound lights, the depth can be computed simply using trigonometry. RGB-D sensors have led to a boost of 3D perception particularly for robot applications [138]. Various related applications have been also proposed in computer graphics and computer vision, e.g., surface reconstruction [123] and simultaneous localization and mapping (SLAM) [138]. However, as this technique is new and fast

(a) (b) (c)

Figure 11: (a) shows a handheld scanner; (b) displays a mobile mapping system; (c) is a motion sensing input device.

developing, and it still cannot achieve same performance as the traditional scanners.

2.2 MULTI-VIEW STEREO RECONSTRUCTION

The range sensors previously stated are widely applied. However, mostly the traditional range sensors are costly and difficult to operate. On the other hand, in the computer graphics and computer vision communities, inexpensive and easy set-up acquisitions are more favorable, e.g., multi-view stereo reconstruction which only requires cameras. The principle of multi-view stereo can be simply illustrated as Figure 12b. In contrast to Figure 12a, the light projector and the CMOS are substituted by CCD cameras. A depth map can be computed by finding the correspondences between images.

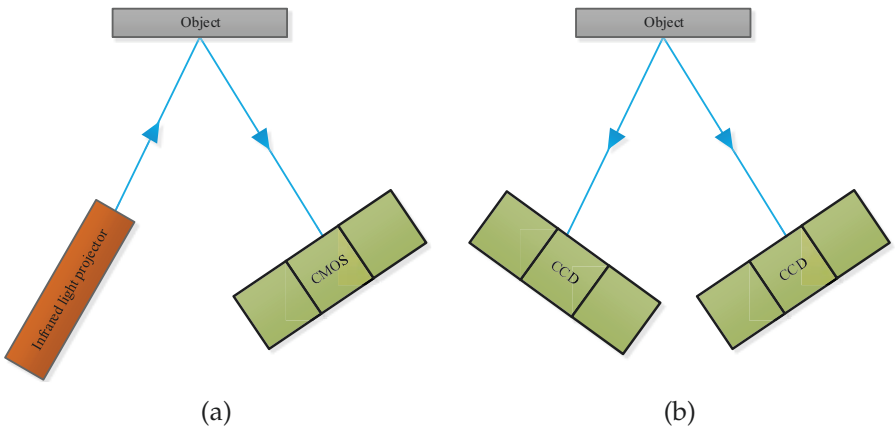


Figure 12: The working principle of Kinect is simply illustrated in (a) and multi-view stereo has a similar principle as shown in (b).

A large number of related work has been proposed [105, 66, 77, 67, 68], as well as industrial products [170, 116, 2]. The number of images used in reconstruction can vary from two such as binocular stereopsis [6] to thousands [152]. In addition, camera parameters can be unknown or estimated by camera calibrations [118]. In different methods, the results can be point clouds, meshes, patches and so forth. A taxonomy for multi-view methods is also proposed in [147] according to scene representation, photo consistency measure, visibility model, shape prior, reconstruction algorithm, and initialization requirement.

2.2.1 *Camera model*

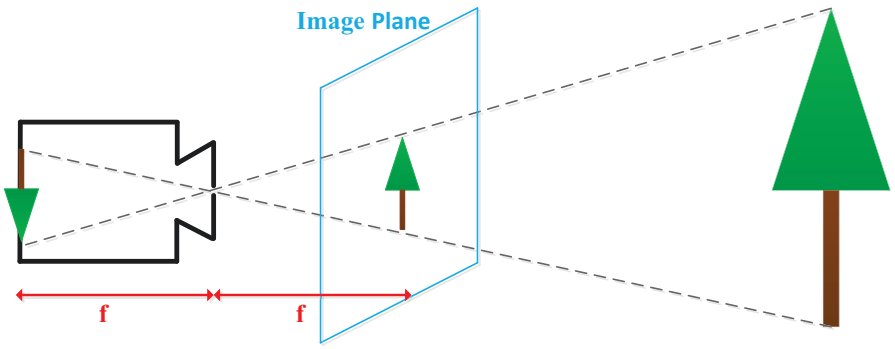


Figure 13: An illustration of the pinhole camera: The rays of light go through the small hole on the black box and project onto a photosensitive film on the opposite side. Iy this way, an image of a tree is generated. Noticeably, the tree in the image is upsidedown and thus the image is named as a photo-negative image. The corresponding photo-positive image can be considered on the virtual image plane (blue). In addition, the photo-negative and the virtual photo-positive images are equidistant from the hole.

In the literature survey [147], various methods on multi-view stereo reconstruction are reviewed, discussed and evaluated. Although these different methods use distinct techniques, the camera model is the shared base. As illustrated in Figure 13, the simplest camera model is the *pinhole camera* which is a box punctured to yield a small hole. On the opposite side of the box, a photosensitive film is used to capture light that goes through the small hole from outside. Theoretically the hole can be infinitely small, and a photo-negative image is obtained on the film. The distance between the photo-negative image and the hole is the *focal length* usually denoted by f. A virtual plane is created between the tree and the camera, and its distance from the hole is f as well. A photo-positive image can be obtained by projecting the tree on the plane. Furthermore, the virtual image plane is introduced to facilitate the computation of the relation between the 2D image and the 3D scene. As shown in Figure 14, a Cartesian coordinate system is

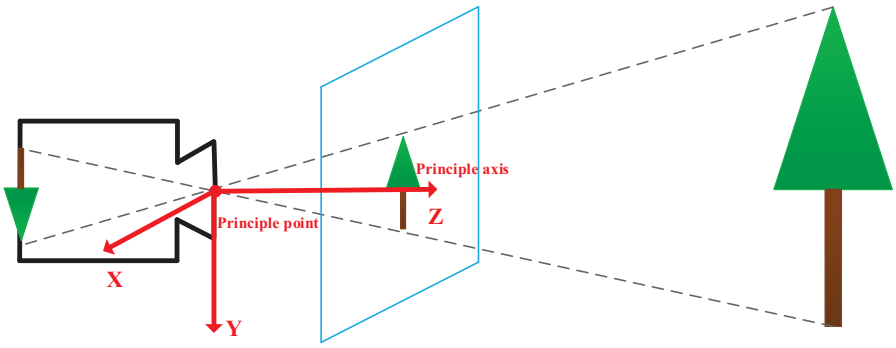


Figure 14: A Cartesian coordinate system is introduced in the pinhole camera model. The hole is regarded as the origin and it is named the principle point. The three axes are colored in red and the z axis is called the principle axis.

built by choosing the hole as the origin and the z axis along the view direction. In the pinhole camera, the origin is named the principle point and the z axis is called the principle axis. The relation between the image and the 3D scene can be characterized mathematically as

$$u = f\,\frac{x}{z} \qquad \text{and} \qquad v = f\,\frac{y}{z}, \tag{1}$$

where (x, y, z) is the 3D coordinates, (u, v) is the image coordinates and f is the focal length.

In digital image, the unit of coordinates is pixel and the origin is usually the left-top corner as illustrated in Figure 15. Hence, using image reference frame Equation 2 can be rewritten as

$$u = m_x f\,\frac{x}{z} + \frac{w}{2} \qquad \text{and} \qquad v = m_y f\,\frac{y}{z} + \frac{h}{2}, \tag{2}$$

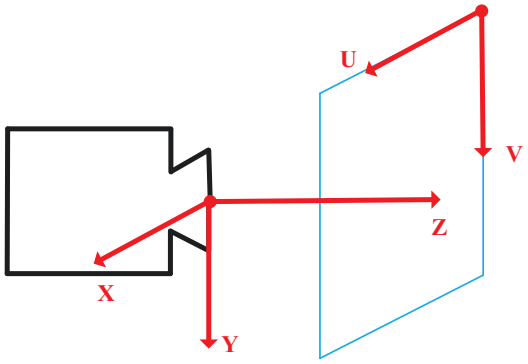


Figure 15: The reference frame of digital image with axes U and V.

or equivalently as a matrix form

$$z\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} m_x f & 0 & \frac{w}{2} \\ 0 & m_y f & \frac{h}{2} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{3}$$

where m_x and m_y are the inverse of the pixel width and height respectively, w and h are the image width and height, z also represents the 3D *depth* behind the camera. The matrix

$$\mathbf{K} = \begin{pmatrix} m_x f & 0 & \frac{w}{2} \\ 0 & m_y f & \frac{h}{2} \\ 0 & 0 & 1 \end{pmatrix} \tag{4}$$

is called the *calibration matrix*.

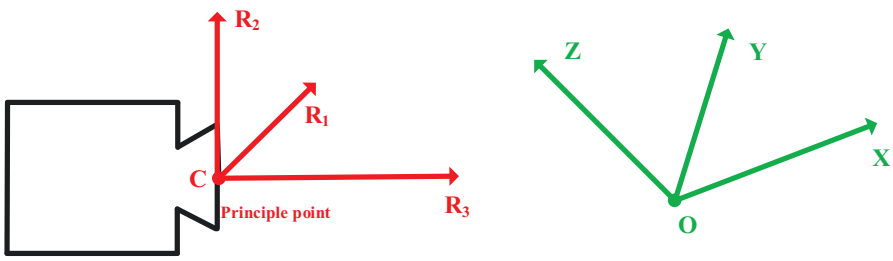


Figure 16: The camera (red) and the world (green) reference frames

However, in practice the camera reference frame used in Equation 1 is unnecessarily aligned with the world reference frame as show in Figure 16. Using the world reference frame, Equation 3 is rewritten as

$$\rho\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K}\mathbf{R}^T \left(\begin{pmatrix} x \\ y \\ z \end{pmatrix} - C \right), \tag{5}$$

where the matrix $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 & \mathbf{R}_2 & \mathbf{R}_3 \end{bmatrix}$, $\mathbf{R}_i$ $(i = 1, 2, 3)$ is the unit column vector along of corresponding axis of the camera reference frame under the world reference frame. $\mathbf{C}$ is the coordinate of the principle point under the world reference frame, and ρ is a positive value encoding the depth. Similar to Equation 3, Equation 5 can also be represented as the matrix form

$$\rho \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{K}\mathbf{R}^\top & -\mathbf{K}\mathbf{R}^\top\mathbf{C} \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, \tag{6}$$

The matrix $\mathbf{P} = \begin{bmatrix} \mathbf{K}\mathbf{R}^\top & -\mathbf{K}\mathbf{R}^\top\mathbf{C} \end{bmatrix}$ is named the *projection matrix*. The camera model defined in Equation 6 is called the *general linear camera model*.

If the matrix $\mathbf{P}$ is known, it is possible to compute $\mathbf{K}$, $\mathbf{R}$ and $\mathbf{C}$ reversely. $\mathbf{K}\mathbf{R}^\top$ is a full rank matrix as well as its inverse $\mathbf{R}\mathbf{K}^{-1}$. QR-decomposition [133] demonstrates a full rank matrix can be uniquely factorized into an orthogonal matrix multiplied by an upper triangular matrix with positive diagonal. Since $\mathbf{R}$ is an orthogonal matrix and $\mathbf{K}^{-1}$ is an upper triangular matrix with positive diagonal, $\mathbf{R}$ and $\mathbf{K}^{-1}$ can be computed by using QR-decomposition. Given $\mathbf{R}$ and $\mathbf{K}^{-1}$, $\mathbf{C}$ is also easily obtained. This process is the *camera calibration* to determine the parameters in the projection matrix. $\mathbf{K}$ encodes the *internal* parameters, while $\mathbf{R}$ and $\mathbf{C}$ encode the *external* parameters.

2.2.2 *Principles of 3D reconstruction*

3D reconstruction seeks to recover 3D geometry from one or multiple images, i.e., given a pixel of an image the corresponding 3D point is computed. As shown in Figure 17a, c_1 is the principle point and m_1 is a pixel of an image, the dashed line named the *projection ray* illustrates the affect of the projection matrix transforming 3D points to image pixels. However, besides the 3D point M all the points of the projection ray are projected on m_1 by using the general linear camera model stated in Section 2.2.1. It is impossible to compute the depth in Equation 6 without further knowledge. Therefore, 3D reconstruction from single image is an underdetermined problem.

3D reconstruction can be performed when two images captured from different views are given. As shown in Figure 17b, the 3D point M can be recovered by computing the intersection of the two projection rays passing the projections of M, i.e, m_1 and $m2$. This principle is similar to the one shown in Figure 12a, which is discussed in Sec-

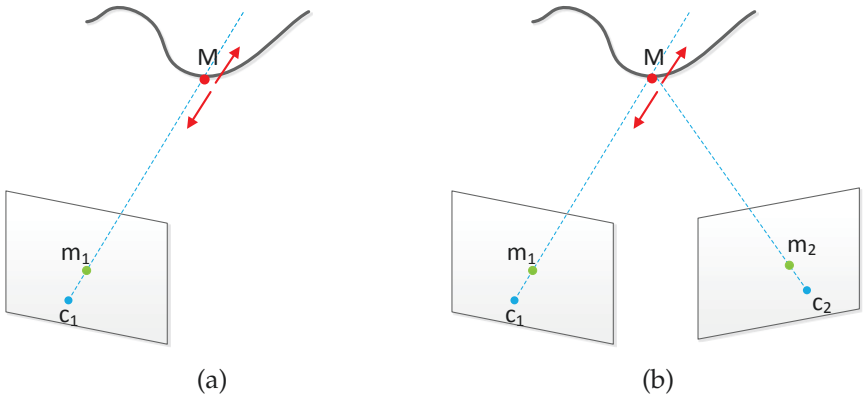


Figure 17: (a): 3D reconstruction from single image is an underdetermined problem, because m_1 is the projection of all points along the projection ray. (b): 3D reconstruction from two images is possible, and a 3D point can be constructed by finding the intersection of the two projection rays.

tion 2.1. Mathematically, using Equation 6 on m_i $(i = 1, 2)$, the linear system is formulated as

$$\begin{cases} \rho_1 m_1 = \begin{bmatrix} \mathbf{K}_1\mathbf{R}_1^\top & -\mathbf{K}_1\mathbf{R}_1^\top\mathbf{C}_1 \end{bmatrix} M \\ \rho_2 m_2 = \begin{bmatrix} \mathbf{K}_2\mathbf{R}_2^\top & -\mathbf{K}_2\mathbf{R}_2^\top\mathbf{C}_2 \end{bmatrix} M, \end{cases} \tag{7}$$

where ρ_i is the depth, $\mathbf{K}_i$, $\mathbf{R}_i$ and $\mathbf{C}_i$ are camera parameters defined in Section 2.2.1. The linear system consisting of six equations has five unknown variables in ρ_i and M. Therefore, if cameras are fully calibrated, 3D reconstruction of two images can be addressed by solving the linear system Equation 7. On the other hand, if camera parameters are unknown, the problem need be further examined, namely, the relation between the projections m_i $(i = 1, 2)$ and the 3D point M in Figure 17b. For example, as shown in Figure 18, m_1, $\mathbf{C}_1$ and $\mathbf{C}_2$ determine a plane intersecting the other image as the yellow line displayed. The corresponding pixel m_2 of the pixel m_1 is on the *epipolar line*, otherwise the projection rays cannot intersect. The epipolar line is the intersection of the image plane and the plane determined by $\mathbf{C}_1$, $\mathbf{C}_2$ and M.

In summary, the principles stated previously serve as the common base of different 3D reconstruction problems, e.g., Euclidean 3D reconstruction, affine 3D reconstruction and projective 3D reconstruction [78]. However, in general, more than two images are required to obtain a 3D reconstruction without knowing camera parameters. Therefore, the relations between multiple views are also explored in the following section.

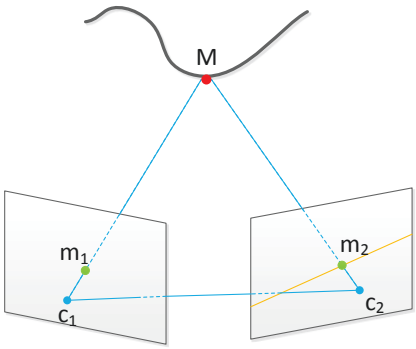


Figure 18: The corresponding pixel m_2 of the pixel m_1 is on the epipolar line by the right camera.

2.2.3 *Relations between multiple views*

Similar to the 3D reconstruction using two images as stated in Section 2.2.2, the key of using multiple images is establishment of point correspondences between different images. In this section, the case of using three images is presented.

To obtain the point correspondences between three views, a trivial solution is dealing the three images as two image pairs. As shown in Figure 19, m_1, m_2 and m_3 are three points in image 1, 2 and 3 respectively and correspond to the same 3D point M. Given image 1 and image 2, an epipolar line L_2 in image 1 can be determined as stated in Section 2.2.2. Similarly, an epipolar line L_3 in image 1 can also be determined by image 3. As discussed previously in Section 2.2.2, m_1 is along L_2 as well as L_3, and thus m_1 is situated in the intersection of the two epipolar lines as illustrated in Figure 24a. However, as shown

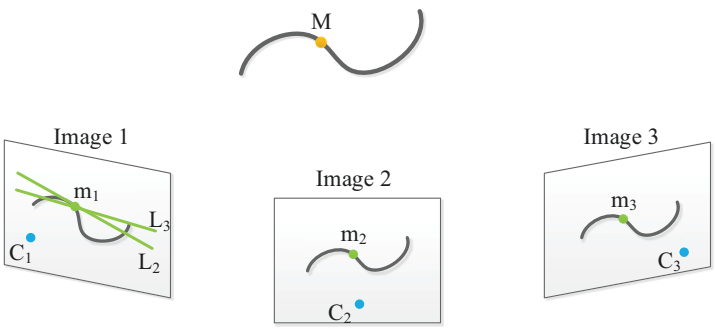


Figure 19: The point m_3 corresponding to m_1 and m_2 is situated in the intersection of the two epipolar lines by the middle camera and the right camera respectively.

in Figure 24b, if m_1, m_2 and m_3 are all located in the plane determined by the three camera centers, namely, C_1 C_2 and C_3, the two epipolar lines L_2 and L_3 coincide with each other, which is a degenerated case. The plane passing through C_1 C_2 and C_3 is called the *trifocal plane*.

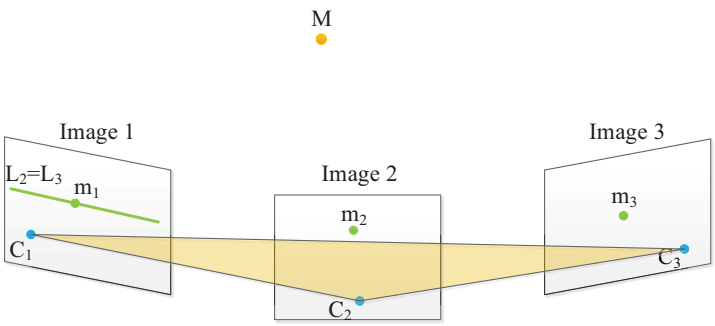
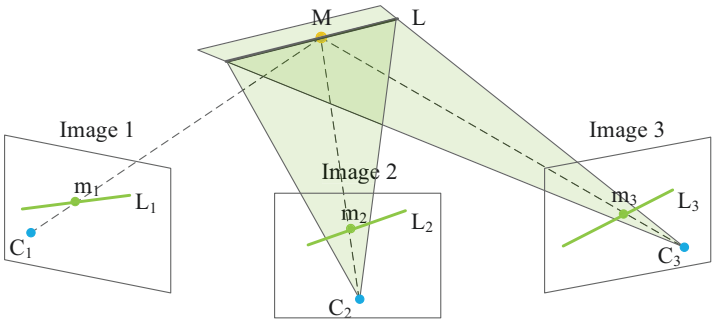
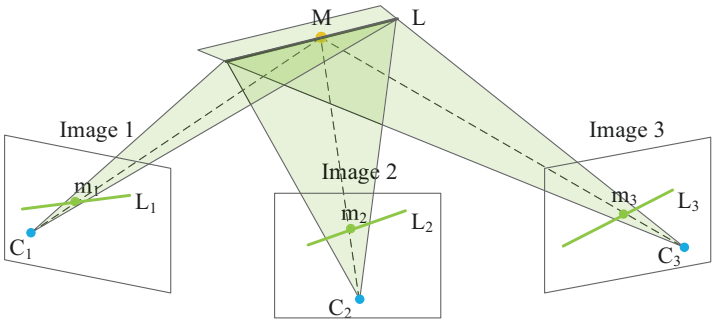


Figure 20: The epipolar lines L_2 and L_3 coincide if m_1, m_2 and m_3 are situated in the trifocal plane.

When the three points m_1, m_2 and m_3 are all close to the trifocal plane, L_2 and L3 almost coincide. In the scenario, it is difficult to robustly compute the intersection of the two epipolar lines, and thus the problem is ill-conditioned. Fortunately, besides the epipolar relation, more algebraic relations can be derived.



(a)



(b)

Figure 21

The relation, which will be presented immediately, can be illustrated intuitively in Figure 21. m_1, m_2 and m_3 are three image points which all correspond to the 3D point M. L_2 in image 2 is an arbitrary line passing m_2, while L_3 in image 3 is also an an arbitrary line passing m_3. As shown in Figure 21a. L_2 and the camera center C_2 can determined a plane, as well as L_3 and C_3. In general, the two gen-

erated planes intersect in a line denoted as L. The projection L_1 of L in image 1 must pass through the point m_1 as shown in Figure 21b. This relation is expressed mathematically as follows.

Using the Equation 5, three projection equations are obtained

$$\begin{cases} \rho_1 m_1 = K_1 R_1^T (M - C_1) \\ \rho_2 m_2 = K_2 R_2^T (M - C_2) \\ \rho_3 m_3 = K_3 R_3^T (M - C_3). \end{cases} \tag{8}$$

The first equation in Equation 8 is rewritten as $M = \rho_1 R_1 K_1^{-1} m_1 + C_1$ and is substituted into the other two equations

$$\begin{cases} \rho_2 m_2 = \rho_1 K_2 R_2^T R_1 K_1^{-1} m_1 + K_2 R_2^T (C_1 - C_2) \\ \rho_3 m_3 = \rho_1 K_3 R_3^T R_1 K_1^{-1} m_1 + K_3 R_3^T (C_1 - C_3). \end{cases} \tag{9}$$

Noticeably, $K_2 R_2^T (C_1 - C_2)$ and $K_3 R_3^T (C_1 - C_3)$ are the projections of C_1 with respect to image 2 and image 3 respectively. For conciseness, $A_2 = K_2 R_2^T R_1 K_1^{-1}$, $A_3 = K_3 R_3^T R_1 K_1^{-1}$, $\rho_{e_2} e_2 = K_2 R_2^T (C_1 - C_2)$ and $\rho_{e_3} e_3 = K_3 R_3^T (C_1 - C_3)$ are introduced. Thus, Equation 9 is rewritten as

$$\begin{cases} \rho_2 m_2 = \rho_1 A_2 m_1 + \rho_{e_2} e_2 \\ \rho_3 m_3 = \rho_1 A_3 m_1 + \rho_{e_3} e_3. \end{cases} \tag{10}$$

Since the line L_2 passes m_2 and the line L_3 passes m_3, ${L_2}^T m_2 = 0$ and ${L_3}^T m_3 = 0$ hold. Hence, Equation 10 is transformed into

$$\begin{cases} 0 = \rho_1 {L_2}^T A_2 m_1 + \rho_{e_2} {L_2}^T e_2 \\ 0 = \rho_1 {L_3}^T A_3 m_1 + \rho_{e_3} {L_3}^T e_3. \end{cases} \tag{11}$$

A_2, A_3, e_2, e_3, ρ_{e_2} and ρ_{e_3} entirely depends on the configuration of the cameras. In addition, L_2 and L_3 are arbitrary lines as long as passing m_2 and m_3 respectively. Therefore, ρ_1 is the only unknown variable depending on the 3D point M. To eliminate ρ_1 in Equation 11, the following equation is obtained

$$({L_2}^T A_2 m_1)(\rho_{e_3} {L_3}^T e_3) - (\rho_{e_2} {L_2}^T e_2)({L_3}^T A_3 m_1) = 0. \tag{12}$$

By reordering the terms in Equation 12, the equation is rewritten as

$${L_2}^T \mathbf{T}(m_1) L_3 = 0, \tag{13}$$

where $\mathbf{T}(m_1) = (A_2 m_1)(\rho_{e_3} e_3) - (\rho_{e_2} e_2)(A_3 m_1)^T$, which describes the fundamental trifocal relation. Moreover, if $m_1 = (x_1, x_2, 1)^T$, $\mathbf{T}(m_1)$ can be expressed as

$$\mathbf{T}(m_1) = \mathbf{T}_1 x_1 + \mathbf{T}_2 x_2 + \mathbf{T}_3, \tag{14}$$

which indicates that $\mathbf{T}(m_1)$ is a linear combination of $\mathbf{T}_i$ with the coefficients m_1. The matrix $\mathbf{T}_k$ only depends on the configuration of the cameras, and the value of its (i, j)-position is

$$\mathbf{T}_k^{ij} = \rho_{e_3}(A_2)_{ik}(e_3)_j - \rho_{e_2}(A_3)_{jk}(e_2)_i \ . \tag{15}$$

In summary, if m_1, m_2 and m_3 are the corresponding points in different three images, for all line L_2 passing through m_2 and L_3 passing through m_3, the fundamental trifocal relation as Equation 13 holds. For further discussion, the reader is referred to the book on multiple view geometry [78].

2.2.4 *Multi-view setup*

Different from range sensor based methods, cameras in multi-view methods cost much less and can produce images with a good resolution such as more than ten million pixels. Therefore, by using multi-view reconstruction, it is also possible to obtain a high resolution and accurate acquisition. Furthermore, the acquisition can be performed for objects with different scales, which is a significant advantage as well. Therefore, multi-view stereo will be specially concerned and Figure 22 illustrates the multi-view setup used in this thesis. The setup

(a) (b) (c)

Figure 22: The multi-view setup used in this thesis: (a) Six synchronized video cameras are calibrated and two studio lighting kits are used to improve illumination; Six images (b) of a face are captured from different views; A point cloud (c) of the face is generated by multi-view reconstruction.

consists of synchronized six video cameras and two studio lighting kits. The camera can capture images at 17 FPS with a full resolution of 1388×1038, and the lighting kits are used to offer sufficient and uniform light. The camera calibrations [174, 156] are performed before the acquisition, and six images are captured from different views as shown in Figure 22b. The face is painted with black patterns to facil-

itate image matching by enriching textures. Afterwards point clouds can be computed by applying the algorithms [78, 157] as shown in Figure 22c.

2.3 SUMMARY

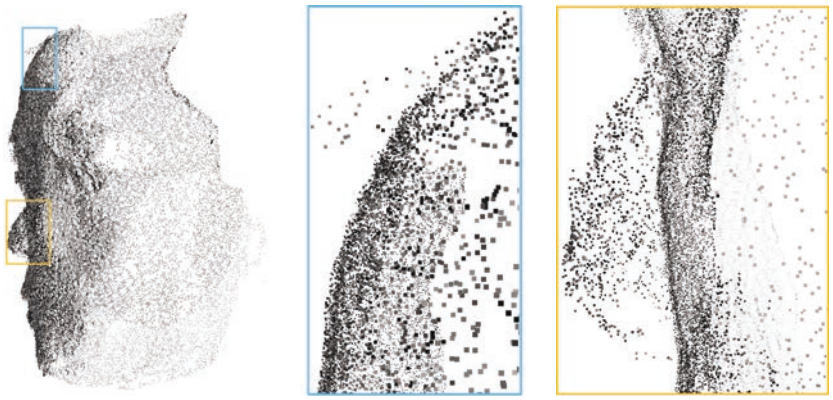
In this chapter, the mainstream methods for 3D data acquisition, especially multi-view stereo, are reviewed. Various acquisition methods attract the attention from both academia and industry. Generally the range sensors can perform a high-resolution and accurate 3D scanning. However, they are usually costly and are not easy to manipulate. These problems introduce an obstacle to the more widespread use. On the other hand, only cameras are required in multi-view stereo reconstruction. Comparing to the traditional range sensors, multi-view reconstruction is much cheaper, more simply operated, and thus it is mostly concerned in this thesis. Moreover, two algorithms tailored for multi-view data are proposed respectively for point cloud refinement in Chapter 3 and surface reconstruction in Chapter 6.

3 STATE OF THE ART IN POINT CLOUD REFINEMENT

Many different 3D acquisition techniques can be used to produce point clouds as stated in Chapter 2. As one of the most popular techniques, multi-view reconstruction generates quasi-dense or dense point clouds [105, 68]. However, varying degrees of geometric errors are generally in presence of these point clouds such as outliers and noise. These errors can severely affect applications using these point clouds, and hence, in this chapter, point cloud refinement is comprehensively studied to improve the quality of such data.

3.1 BACKGROUND

A point cloud acquired by multi-view reconstruction is displayed in Figure 23a. In Figure 23b and Figure 23c, the geometric errors outliers and noise are respectively highlighted in close-up view. Notably, the outliers also result in the multilayer issue as illustrated in Figure 23b. As previously discussed in Section 2.2.2, multi-view reconstruction



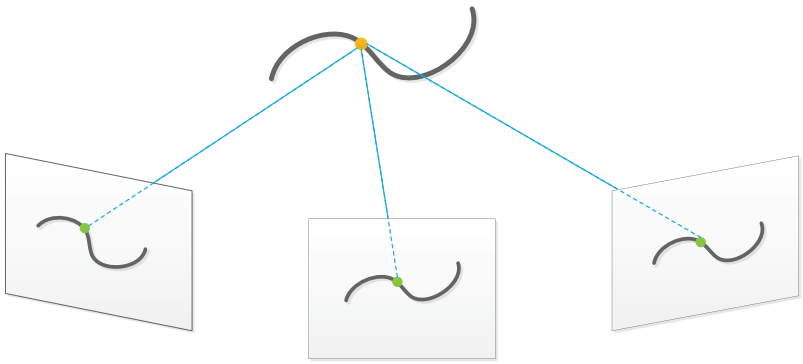
(a) (b) (c)

Figure 23: (a) displays an unrocessed point cloud generated by multi-view stereo reconstruction. (b) and (c) are two close-up views which highlight outliers and noise respectively.

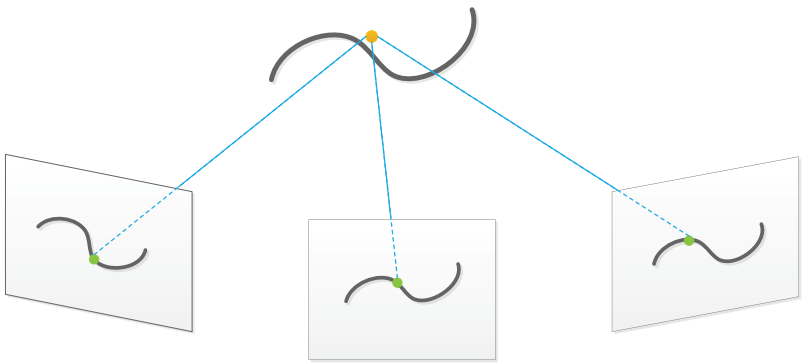
is based on image matching. Theoretically, exactly accurate matching leads to a perfect 3D reconstruction as shown in Figure 24a. On the other hand, inaccurate matching can cause unexpected errors as demonstrated Figure 24b. These errors can be outliers or noisy points, which are determined by the quality of image matching.

In order to enhance image matching, an extensive amount of work has been proposed [88, 89, 97]. These methods generally adopt a

quasi-dense matching framework. Sparse matching pixels are first computed as seeds. New matching candidates are examined and selected from the neighborhood of current seeds until matches propagate to cover the most pixels among multiple images. Such propagation is usually constrained by some criterion such as epipolar geometry. The occurrence of outliers can be significantly decreased by increasing matching accuracy. However, in practice, the matching errors cannot be completely eliminated in view of many factors such as ill textures, spatial discretization, and lighting conditions. In addition to outliers and noise, irregular sampling in multi-view data can be problematic as well. As shown in Figure 23c, points in the cheek are sparser than other regions. This issue can result in an ill-shaped neighbor graph whose quality is essential in many geometry processing operations. Spefifically, it can badly affect surface reconstruction methods discussed in Chapter 5 as certain required sampling conditions are violated. Therefore, point cloud refinement for multi-view data is challenging and important.



(a)



(b)

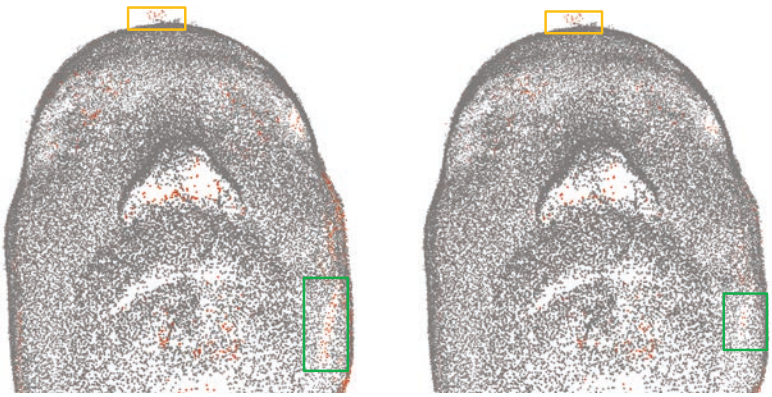
Figure 24: Illustration of multi-view stereo reconstruction with accurate and inaccurate matching respectively. The 3D dot (orange) is computed based on the corresponding pixels (green).

This chapter is organized as follows. The existing methods for outlier filtering are presented in Section 3.2. In Section 3.3, the techniques for noise filtering used in geometry processing and multi-view recon-

struction are reviewed. Bundle adjustment, which is the most important technique to refine multi-view data, is studied in Section 3.4.

3.2 OUTLIERS FILTERING

A large amount of research work has been proposed in the literature on outlier filtering for point data [136, 153, 32, 167]. In fact, there is no unified precise definition for outlier. In general, outliers are identified as points that seriously disagree with local geometry, otherwise points are inliers. To filter outliers in point clouds, an outlier detection method is usually applied and afterwards detected outliers are removed from the original data.



(a) (b)

Figure 25: The two point clouds are the results of outlier detection methods using distance based criterion (a) and density based criterion (b) respectively. The orange rectangles highlight outliers and the green rectangles highlight the points misclassified as outliers.

In the existing methods, different criteria are applied to detect outliers, and the methods are generally categorized into neighborhood based ones and geometry based ones. The former ones are performed by analyzing properties of point neighborhood such as shape and sampling. The latter ones are on the basis of reconstructed geometry. In addition, neighborhood based methods are further sub-grouped into distance based ones inspired by that outliers are isolated from other points and density based ones inspired by that outliers are in presence of low density areas. Figure 25 displays the results obtained by applying methods using the distance based criterion and the density based criterion, respectively. Quite a few points are misclassified as individual criterion is fairly not sufficient. In practice, multiple criteria are often combined to yield a satisfactory outlier classifier. Table 1 summaries the criteria discussed in this subsection.

In [136], an outlier detection method is proposed by means of analyzing statistical information of point neighborhoods. For each point p in a point cloud P, the mean value $\bar{d}_k(p)$ of distances between p

Detection criteria		Methods
Neighborhood based	Distance based	[136, 167]
	Density based	[153, 32, 167]
Geometry based		[167]

Table 1: The criteria used in the methods discussed in Section 3.2.

and its k-nearest neighbors [120] is computed. Given the computed set $\{\bar{d}_k(p) \mid p \in P\}$, the mean value μ_k and the standard derivation σ_k of the statistical variable $\bar{d}_k(p)$ are calculated. Intuitively, an outlier is distant from the original geometry and thus tends to have a larger $\bar{d}_k(p)$. Based on this observation, outliers can be detected as

$$P_{outliers} = \{p \in P \mid \bar{d}_k(p) \geqslant (\mu_k + \alpha\sigma)\}, \tag{16}$$

where α is a prescribed parameter. This approach is simple and efficient, however, it turns to be problematic as arising many misclassified results when point sampling becomes nonuniform as illustrated in Figure 26. The points in less dense areas have larger distances from their neighbors and thus $\bar{d}_k(p)$ as well. Therefore, these points are easily misclassified as outliers using the criterion stated in Equation 16.

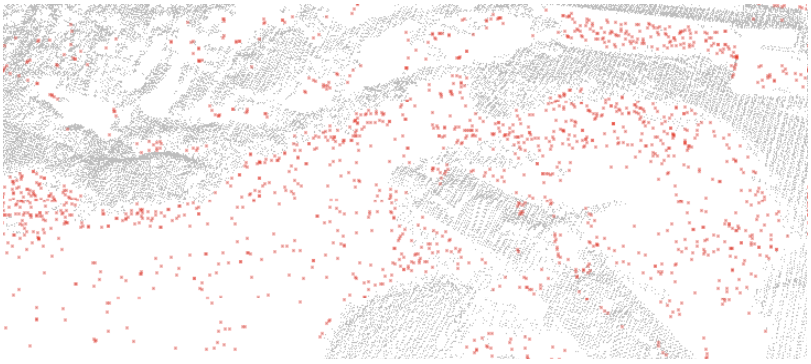


Figure 26: The method proposed in [136] cannot handle point clouds with nonuniform sampling. Many points in less dense areas are incorrectly classified as outliers (red points).

To resolve the sampling density issue aforementioned, a method is proposed in [153] by taking advantage of examining the local density of point neighborhood. Inspired by [32], *local outlier factor* (LOF) is used to predict outliers. In order to define local outlier factor, the terms of *k-distance*, *reachability distance* and *local reachability density* are introduced beforehand. The *k-distance* of a point $p \in P$, denoted as $\text{k-distance}(p)$, is the k-th largest distance from other points in P to p, and the k-nearest neighbors of p is denoted as $N_k(p)$. The *reachability distance* of q with respect to the point p is defined as

$$\text{reach-dist}_k(q, p) = \max\{\text{k-distance}(p), d(q, p)\}, \tag{17}$$

where $d(q,p)$ is the Euclidean distance between q and p. For example, $\text{reach-dist}_4(q_1,p) = \text{k-distance}(p)$ and $\text{reach-dist}_4(q_2,p) = d(q_2,p)$ as shown in Figure 27. In addition, *local reachability density* is defined

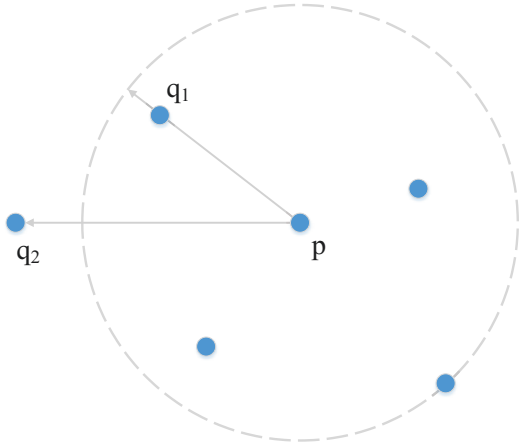


Figure 27: An illustration of the reachability distance with respect to the point p ($k = 4$).

as

$$\text{lrd}_k(p) = \frac{|N_k(p)|}{\sum\limits_{q \in N_k(p)} \text{reach-dist}_k(p,q)}, \tag{18}$$

where $|N_k(p)|$ is the cardinality of the set $N_k(p)$. Intuitively, the value of $\text{lrd}_k(p)$ is small if p is an outlier, because $p \notin N_k(q)$ and the value of $\text{reach-dist}_k(p,q)$ is large. Once all the preliminary ingredients are determined, *local outlier factor* is defined as

$$\text{LOF}_k(p) = \frac{\sum\limits_{q \in N_k(p)} \frac{\text{lrd}_k(q)}{\text{lrd}_k(p)}}{|N_k(p)|} \tag{19}$$

It can be proved that $\text{LOF}_k(p)$ approximately equals to 1 if p is an inlier [32]. Hence, outliers are identified in this way.

Although lots of automatic methods are proposed, none of them can be applied for general point clouds. Therefore, an interactive method [167] is developed to help detect and remove outliers. The interactive method integrates three criteria, namely, *plane fit*, *miniball* and *nearest-neighbor reciprocity*, to yield an outlier classifier. The plane fit criterion is illustrated in Figure 28. The least square plane H is computed to fit the k-nearest neighbors $N_k(p)$ of p. The distance from p to H and the average distance of points in $N_k(p)$ to H, denoted as d_p and $\bar{d}_k(p)$ respectively, are used to predict outliers as

$$\chi_{pl}(p) = \frac{d_p}{d_p + \bar{d}_k(p)}. \tag{20}$$

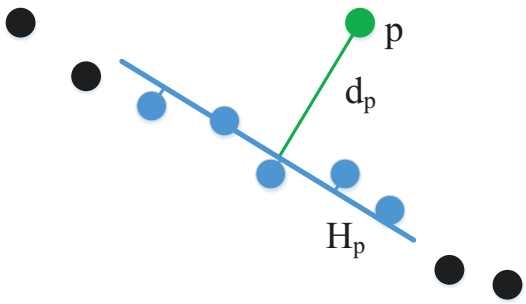


Figure 28: The plane fit criterion compares the distance from p to the plane H with the average distance of points in $N_k(p)$ to H. $N_k(p)$ is the k-nearest neighbors of p and $k = 5$ here. H is the least squares plane fitting the points (blue) in $N_k(p)$.

Thus, $\chi_{pl} \in [0, 1]$ is an estimator which assigns a likelihood of an outlier to each point. Similar to the plane fit criterion, as shown in

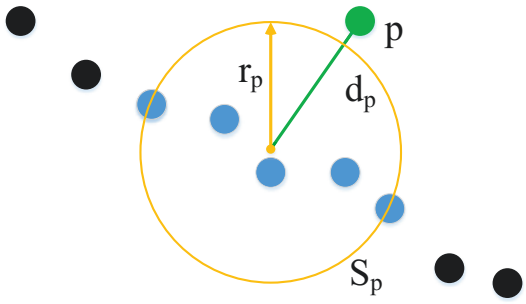


Figure 29: The miniball criterion compares the distance from p to the center of sphere S_p and the sphere radius r_p. S_p is the smallest sphere enclosing the k-nearest neighbors of p ($k = 5$ in this case).

Figure 29, the miniball criterion computes the smallest sphere S_p enclosing $N_k(p)$ and then compare the distance d_p from p to the sphere center and the sphere radius r_p. An estimator χ_{mb} is also defined as

$$\chi_{mb}(p) = \frac{d_p}{d_p + 2r_p/\sqrt{k}} \ , \tag{21}$$

where $\sqrt{k}$ compensates the increase of r_p when k grows bigger. The nearest-neighbor reciprocity criterion is based on the observation that an outlier p does not necessarily stay in $N_k(q)$ where $q \in N_k(p)$ as explained in Figure 30. Similarly, an estimator is defined as

$$\chi_{bi}(p) = \frac{|N_k^{uni}(p)|}{k}, \tag{22}$$

where $N_k^{uni}(p) = \{q \mid q \in N_k(p), p \notin N_k(q)\}$ and $|N_k^{uni}(p)|$ is the cardinality of $N_k^{uni}(p)$. Combing the three estimators associated with

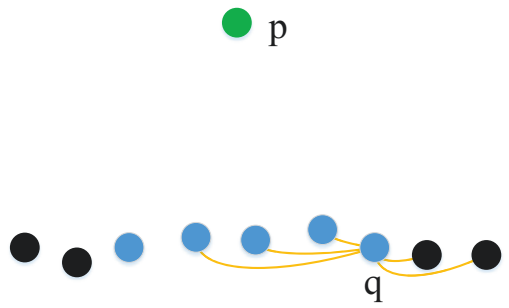


Figure 30: The 5-nearest neighbors of (green) point p is displayed in blue. The point q is one of the neighbors, but the 5-nearest neighbors of q doesn't contain p.

weights w_i which are prescribed interactively, a point p can be determined if it is an outlier by calculating

$$\chi(p) = w_1\chi_{pl}(p) + w_2\chi_{mb}(p) + w_3\chi_{bi}(p), \tag{23}$$

where $\sum w_i = 1$. Since the computation of the estimators only needs k-nearest neighbors which is possibly pre-computed, the result can be updated with low cost once the weights are adjusted.

Remarks: Although the methods aforementioned for outlier filtering are purely geometric approaches originally developed for range sensor data, however, generally they can be applied for multi-view point data without any modification. This is because the operation of outlier filtering just permanently eliminates outliers without affecting other point data. In addition, several other solutions are also proposed in multi-view reconstruction to prevent outliers such as using the segmentation masks [69, 68]. For each image, a segmentation mask is generated automatically or manually to specify the foreground (reconstructed object) and the background. During multi-view reconstruction, a computed 3D point can be reprojected (see Section 2.2.1) in the image domain to check if it is an outlier. Specifically, if a 3D point is reprojected in the region of the background, the point can be labeled as an outlier.

3.3 NOISE FILTERING

Many efforts have been denoted to obtain smooth geometry in various multi-view reconstruction methods. Most of the existing techniques are inspired by the corresponding approaches in the area of geometry processing. Therefore, these related approaches in geometry processing are first examined in this subsection.

3.3.1 *Techniques in geometry processing*

A considerable amount of work has been proposed for noise filtering in geometry processing. Most of these methods are purely geometry based and are originally designed for range sensor data.

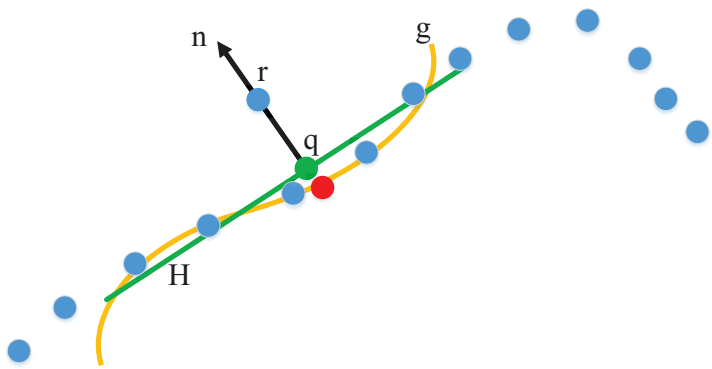


Figure 31: An illustration of the moving least squares projection: A local frame $\{n, H, q\}$ is determined first, and a local bivariate polynomial approximation g is computed. The projection of r is displayed as the red dots.

In [7], a method based on *moving least squares* is proposed to smooth point clouds efficiently. The basic idea is first constructing a smooth implicit surface with help of moving least squares and obtaining a new position afterwards for each point by projecting the point on the surface. Concretely, as shown in Figure 31, for the point r, a local reference frame $\{n, H, q\}$ is determined by minimizing the cost function

$$\sum_i \langle n, p_i - q\rangle^2 \theta(\|p_i - q\|), \tag{24}$$

with constraints $q = r + tn$ and $\|n\| = 1$. $\langle \cdot, \cdot \rangle$ is the inner product, p_i are points in the point cloud, and θ is a Gaussian function. In addition, H is the local plane, q is the origin of the local reference frame, and n is the unit vector orthogonal to H. Once the local reference frame is defined, a local bivariate polynomial approximation g can be computed. The projection (the red dot) of r defined as $g(0, 0)n + q$ is the new position of the point r. In this way, noisy points are moved towards to the correct positions and thus a smoother point cloud is obtained.

The technique of mean-shift has been applied to solve clustering problems in computer vision [65, 38], as well as nosie filtering in geometry processing [141]. The main idea of the approach presented in [141] is defining a smooth likelihood function with respect to the probability of that a point belongs to the real surface. In order to remove noise, the mean-shift based iteration schema is performed to move each point to the local maximum of the likelihood function. In detail, given the point cloud $\{p_i\}$, for each p_i, a local least squares

plane is estimated and denoted as $(x - c_i)^T n_i = 0$. By using the local plane, a local likelihood function is defined as

$$L_i(x) = \Phi_i(x - c_i)(h^2 - ((x - c_i)^T n_i)^2) \tag{25}$$

The parameter h is called the kernel size [21] and Φ is an anisotropic Gaussian function. Thus a point x closer to the least squares plane has a higher probability value $L_i(x)$. The global likelihood function is defined as the weighted sum of L_i

$$L(x) = \sum_i w_i L_i(x), \tag{26}$$

where $w_i \in [0, 1]$. Noticeably, w_i can be determined taking advantage of confidence measures from acquisition systems such as the number of return in LAS data [161]. To seek the local maximums of the function L, gradient descent is applied. Thus, the iteration is performed as

$$p_i^0 = p_i, \quad p_i^{k+1} = p_i^k - m_i^k, \tag{27}$$

where $m_i^k = \frac{\sum_j w_j \Phi_j (p_i^k - c_j)((p_i^k - c_j)^T n_j) n_j}{\tau}$, and τ is the step size. After a few iterations, a noisy point cloud gradually converges to a smooth one.

The noise filtering methods previously discussed are proposed for point data. In fact, many methods have been proposed for mesh data as well in geometry processing. These methods use vertex connectivity in mesh data, and a similar connectivity can be built in point data such as using k-nearest neighbors. Hence, these methods generally can be used for point data with slightly modifications.

In [158], a discrete Laplacian is proposd and is used to remove noise in meshes. The vertex set is denoted as $\{v_i\}$, and the neighbor set of v_i is denoted as N_{v_i}. In a mesh, $v_j \in N_{v_i}$ if v_i and v_j share the same face. The discrete Laplacian $\triangle$ is defined as

$$\triangle x_i = \sum_{v_j \in N_{v_i}} w_{ij}(v_j - v_i), \tag{28}$$

where weights $\{w_{ij}\}$ are positive, and $\sum_{v_j \in N_{v_i}} w_{ij} = 1$. The weights can be chosen in various ways, and a simple way is using the inverse of the cardinality of N_{v_i}, i.e., $w_j = 1/|N_{v_i}|$. A more general way is using a positive function $\phi(v_i, v_j) = \phi(v_j, v_i)$, and the weights are defined as

$$w_{ij} = \frac{\phi(v_i, v_j)}{\sum_{v_k \in N_{v_k}} \phi(v_i, v_k)}. \tag{29}$$

For example, the function ϕ can be defined by means of the inverse distance

$$\phi(v_i, v_j) = \frac{1}{\|v_i - v_j\|}. \tag{30}$$

In principle, the discrete Laplacian described in Equation 28 can be regarded as a type of measure for the difference between v_i and its neighbors $v_j \in N_{v_i}$. With the help of the Laplacian, a smoothing procedure can be conducted for each point v_i with the formulation

$$v_i' = v_i + \lambda \triangle v_i, \tag{31}$$

where λ is a scale factor (see Figure 32).

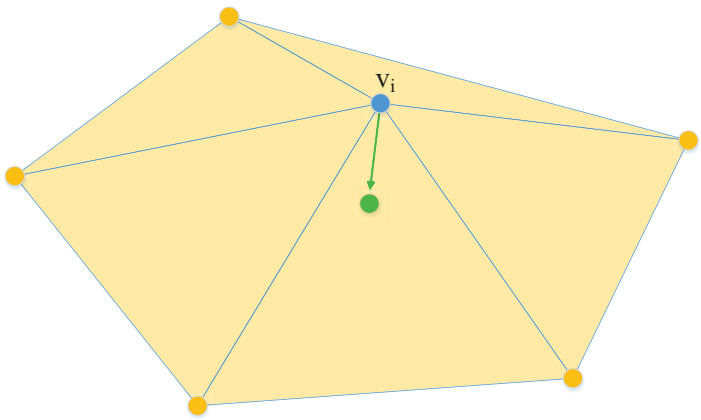


Figure 32: An illustration of the discrete Lapalcian $\triangle$: the yellow dots are the neighbors of v_i, and a new position (green) is obtained by translating v_i with the vector $\lambda \triangle v_i$.

Bilateral filter was a smooth filter which is able to remove noise while preserving edges. It was first proposed in [159] for image smoothing using the formula as

$$I_{fitered}(x) = \frac{\sum_{x_i \in \Omega_x} I(x_i) g(I(x) - I(x_i)) f(x - x_i)}{\sum_{x_i \in \Omega_x} g(I(x) - I(x_i)) f(x - x_i)}, \tag{32}$$

where $I(\cdot)$ is pixel intensity and $g(\cdot)$ and $f(\cdot)$ usually are Gaussian functions. The intensity difference $g(I(x) - I(x_i))$ is accounted in weights for interpolation in addition to distance difference $f(x - x_i)$, therefore, close pixels with very different intensities have small contributions in $I_{fitered}(x)$. Such property can effectively prevent image features from blurring through smoothing. Inspired by this and with the observation that mesh normal has a similar role in meshes as pixel intensity in images, a bilateral denoising method for meshes is proposed in [59]. The nonlinear anisotropic approach is summarized as Algorithm 1,. As you can see, the method has a same framework as the Laplacian method presented in [158]. In the method [59], a point position is

moved along its normal vector, and the scale factor is computed by using both Euclidean distance and orthogonal distance. Notably neighbors with large orthogonal distances can hardly effect the new point position, therefore, the geometry features such as sharp edges can be preserved after the denoising.

Algorithm 1 Bilateral mesh denoising [59]

Input: vertex $\mathbf{v}$, normal $\mathbf{n}$
Output: vertex $\hat{\mathbf{v}}$

```
{v_i} = neighborhood(v)
K = |{v_i}|
sum = 0
normalizer = 0
for i = 1 : K do
    t = ||v − v_i||
    h = ⟨n, v − v_i⟩
    w_c = exp(−t^2/(2σ_c^2))
    w_s = exp(−h^2/(2σ_s^2))
    sum += (w_c w_s)h
    normalizer += w_c w_s
end for
return v̂ = v + n · sum/normalizer
```

Remarks: The approaches of outlier filtering targeted at range sensor data can be directly applied for multi-view data as discussed previously. However, direct use of noise filtering algorithms on multi-view point data can cause severe problems. The existing algorithms such the ones stated in this sections are performed based on certain geometry assumptions, and they are totally oblivious to the natures of multi-view data. For example, these approaches can badly damage the relationship between 3D scene and images in multi-view data. Therefore, it can be a significant limitation for applications such as photogrammetry and tracking in computer vision. In order to address this issue, a novel smoothing method tailored for multi-view point data is developed in Chapter 4.

3.3.2 *Techniques in multi-view reconstruction*

Fairly rare individual methods are especially proposed to multi-view data smoothing. Most of the existing smoothing techniques, as integrated components, are blended with multi-view reconstruction procedures.

The work [68] presents a multi-view reconstruction method to yield dense point clouds. The procedure of the reconstruction consists of several steps, namely, initial feature matching, expansion, and filter-

ing to produce so-called *patches* which form an oriented point cloud. To generate a mesh using the resulting point cloud, a visual hull is gradually deformed to approximate the point cloud. For the smoothing consideration, a geometric smoothness term is used as a regularization term in the arising optimization problem, i.e.,

$$E_s(v_i) = |-\zeta_1 \triangle v_i + \zeta_2 \triangle^2 v_i|^2/\tau^2, \tag{33}$$

where $\triangle$ is the discrete Laplacian similar to the one defined in Equation 28. In practice, w_{ij} can be defined as the inverse of the cardinality of N_{v_i} or the inverse of the distance $\|v_i - v_j\|$ as discussed in Section 3.3.1. In [44], a weighting scheme is proposed based on curvatures

$$w_{ij} = \cot\alpha_j + \cot\beta_j, \tag{34}$$

where α_j and β_j are the angles opposite the edge e_{ij} as shown in Figure 33. As stated in [44], the curvature based laplacian is demon-

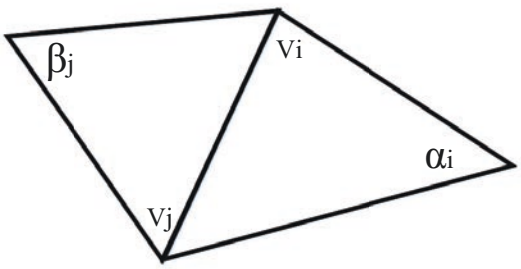


Figure 33: v_i and v_j are two vertices. The edge e_{ij} are shared by two triangles. α_i and β_j are two angles opposite to the edge e_{ij} in the two triangles.

strated to be superior than others for smoothing problem. [165] also suggests that no discrete Laplacian can satisfy all the desirable properties and thus explains the diversity of discrete Laplacian operators..

A method is developed in [80] to deal with the multi-view reconstruction problem especially for large scene. The technique of graph cut is employed in the method to compute a visibility consistent mesh. A variational procedure is performed afterwards to refine the mesh. In the refinement processing, a surface smoothness measure is combined with the global optimization as a regularlizer. To define the regularlizer, the thin plate energy [95] is used. Similar to the method [7], a local quadratic polynomial approximation $f = f(u, v)$ can be computed and used to approximate local geometry. In [80], the thin plate energy is integrated into the regularizer as the form

$$E_{TP}(f) = \int f_{uu}^2 + 2f_{uv}^2 + f_{vv}^2. \tag{35}$$

Thin plate energy, which is first introduced in the areas of geometric design [55], has been widely applied for data interpolation and smoothing. In the problem of data interpolation, for example, a function $f(x)$ is sought to fit pairs $\{(x_i, y_i)\}$. The problem can be typically addressed by minimizing the following objective function

$$E(f) = \|f(x_i) - y_i\|^2. \tag{36}$$

If $f(x)$ is required to be smooth, the thin plate term can be introduced and the Equation 36 is reformulated as

$$E(f) = \|f(x_i) - y_i\|^2 + \int f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2. \tag{37}$$

For the variational problem, [164] demonstrates that theoretically it has unique minimizer. The f is also called as thin plate spline which punishes strong blending in a physical view.

3.4 BUNDLE ADJUSTMENT

Bundle adjustment [78], which simutanneously fine-tunes the 3D structure and the viewing parameters, has become an important step in multi-view reconstruction. The input for bundle adjustment consists of a point cloud $x = (x_1^T, ..., x_n^T)$ originating from multi-view reconstruction and the camera parameters $p = (p_1, ..., p_m)$. Mathematically, bundle adjustment can be formulated as a nonlinear minimization problem with the cost function

$$h(x, p) = \sum_i^n \sum_j^m \delta_{ij} \|Q(p_j, x_i) - a_{ij}\|^2. \tag{38}$$

In the formula, $Q(p_j, x_i)$ and a_{ij} are two points on the image j and correspond to the point x_i, but $Q(p_j, x_i)$ is computed based on the camera model while a_{ij} is obtained using image matching. δ_{ij} encodes the visibility information of x_i with respect to the image j, namely, it equals 1 if the point x_i can be seen on image j, otherwise it equals 0.

3.4.1 *Levenberg-Marquardt (LM) algorithm*

In the arising minimization problem, the cost function in Equation 38 is defined as a sum of nonlinear squares. This suggests that the problem can be addressed taking advantage of nonlinear least squares methods. The best known method to solve this problem is the the Levenberg-Marquardt (LM) algorithm [126], which can be regards as a mix of the Gauss-Newton method [126] and gradient descent [126]. As a modified Gauss-Newton method, the Levenberg-Marquardt al-

gorithm is an iterative technique. In every iteration, a new solution is obtained by solving a linear approximation of the original nonlinear problem. Concretely, P and X are respectively rewritten as

$$P = (a_1^T, ..., a_m^T, b_1^T, ..., b_n^T)^T$$

and

$$X = (x_{11}^T, ..., x_{1m}^T, x_{21}^T, ..., x_{2m}^T, ..., x_{n1}^T, ..., x_{nm}^T)^T.$$

Without loss of generality, all points are assumed to be visible in the images, and thus Equation 38 can be simplified as

$$\underset{P}{\operatorname{argmin}} \|Q(P) - X\|^2. \tag{39}$$

Taking advantage of Taylor expansion, Q can be linearized as

$$Q(P + \delta P) \approx Q(P) + J\delta P, \tag{40}$$

where $J = \frac{\partial Q(P)}{\partial P}$ is the *Jacobian* matrix. Hence, once the current solution P is given, an updated solution can be obtained as $P + \delta P$. By means of Equation 40, Equation 39 can be rewritten as

$$\underset{P}{\operatorname{argmin}} \|J\delta P - (X - Q(P))\|^2. \tag{41}$$

Thus in each iteration the *normal equations* [133] as follows is solved

$$J^T J\delta P = J^T(X - Q(P)). \tag{42}$$

In addition, during the minimization a *damping term* λ is added to the main diagonal of $J^T J$ to guarantee a steady decrease with respect to the cost function. In fact, therefore, the following linear system is solved

$$(J^T J + \lambda I)\delta P = J^T(X - Q(P)), \tag{43}$$

where I is an identity matrix. There are two kinds of variables in P, namely, the camera parameters and the 3D positions. If they are treated separately, J can be written in the form $J = [J_a \ J_b]$, and the matrix of the linear system is represented as

$$J^T J + \lambda I = \begin{bmatrix} U_\lambda & W \\ W^T & V_\lambda \end{bmatrix}, \tag{44}$$

where $U_\lambda = J_a^T J_a + \lambda I$, $V_\lambda = J_b^T J_b + \lambda I$, and $W = J_a^T J_b$. Since U_λ and V_λ are both block diagonal matrices, Shur complement trick [110] can be applied to solve the system more efficiently.

3.4.2 *Newton's method and its variants*

Levenberg-Marquardt algorithm is developed based on *Gauss-Newton algorithm* [126], while Gauss-Newton algorithm is adapted from *Newton's method* [126]. The relations between these three relevant methods are discussed in this section.

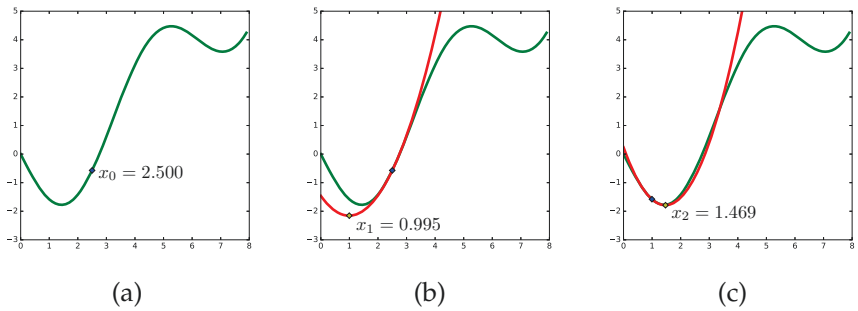


Figure 34: An illustration of solving the optimization problem in Equation 45 by using Newton's method: (a) The objective function is represented by the green curve. The solution is initialized as x_0. (b) The solution is updated by using the Taylor expansion around the current solution. The red curve depicts the quadratic approximation. (c). The solving procedure continues by computing the Taylor expansion around the new solution.

Newton's method [126] is originally used to find the approximated roots of a function. In optimization, it can be used to find an optimum of a problem. For example, Figure 34 illustrates the solving procedure using Newton's method for the problem

$$\text{minimize} \quad f(x) = \frac{x^2}{10} - 2\sin x. \tag{45}$$

Newton's method is an iterative method and the solution is initialized as shown in Figure 34a, i.e., $x_0 = 2.500$. The entire solving procedure is based on the Taylor expansion around the current solution

$$f(x) = f(x_n) + f'(x)(x - x_n) + \frac{1}{2}f''(x_n)(x - x_n)^2 + o(\|x - x_n\|^2). \tag{46}$$

Since the high order term $o(\|x - x_n\|^2)$ can be omitted, the objective function around the current solution can be approximated by the quadratic function

$$y = f(x_n) + f'(x)(x - x_n) + \frac{1}{2}f''(x_n)(x - x_n)^2. \tag{47}$$

The minimum of this function can be easily computed as the yellow dot in Figure 34b and thus the solution is updated, i.e., $x_1 = 0.995$. The solving procedure continues by computing the Taylor expansion around the new solution as shown in Figure 34c.

In Newton's method, the first derivatives and the second derivatives are both required. However, the second derivatives (Hessian matrices for multiple varibles) can be challenging to compute [126]. Gauss-Newton algorithm [126] is proposed to address this issue by using the Jacobian matrix to approximate the Hessian matrix. Gauss-Newton algorithm is used to solve a nonlinear least squares problem as

$$\text{minimize} \quad f(x) = \frac{1}{2}\sum_{j}^{m} r_j^2(x). \tag{48}$$

Taking advantage of matrix notation, the objective function can be written as

$$f(x) = \frac{1}{2} r(x) r(x)^T, \tag{49}$$

where $r(x) = \left[r_1(x), ..., r_m(x)\right]$. The gradient (first derivative) of $f(x)$ is

$$\bigtriangledown f(x) = J(x)^T r(x), \tag{50}$$

and the Hessian (second derivative) of $f(x)$ is

$$\bigtriangledown^2 f(x) = J(x)^T J(x) + \sum_{j}^{m} r_j(x) \bigtriangledown^2 r_j(x), \tag{51}$$

where $J(x)$ is the Jacobian matrix

$$J(x) = \left[\frac{\partial r_j}{\partial x_i}\right]_{j=1,...,m;\ i=1,...,n} = \begin{bmatrix} \bigtriangledown r_1(x)^T \\ \vdots \\ \bigtriangledown r_j(x)^T \end{bmatrix}. \tag{52}$$

In Equation 51, the term $\sum_j^m r_j(x) \bigtriangledown^2 r_j(x)$ can be omitted and the Hessian can be approximated as

$$\bigtriangledown^2 f(x) \approx J(x)^T J(x). \tag{53}$$

In order to avoid the singularity in Equation 53, Levenberg-Marquardt algorithm introduces a damping term and thus the approximated Hessian in Levenberg-Marquardt algorithm is written as

$$\bigtriangledown^2 f(x) \approx J(x)^T J(x) + \lambda I, \tag{54}$$

where I is an identity matrix.

3.4.3 *Extensions of bundle adjustment*

Traditional bundle adjustment are severely affected by two issues, namely, a large amount of required computational resources with respect to both time and memory and the error accumulation [103]. In order to resolve these problems, many extensions of traditional bundle adjustment have been proposed

In the problem of large scale multi-view reconstruction, thousands of images are often used. The arising large optimization procedure cannot be handled by traditional bundle adjustment efficiently with a reasonable memory use. In order to address the problem, an out-of-core bundle adjustment is proposed in [124]. The method utilizes the divide-and-conquer strategy to decouple the original problem into a few smaller one which can be solved in parallel. The key idea is illustrated in Figure 35. The graph representing the optimization graph is divided into two subgraphs. For each subgraph, a new node is introduced to build local coordinate systems in it. Therefore, the whole bundle adjustment problem is reformulated as a few individual small optimization problems and a global aligning. In addition, graph cut is applied in the method to seek an optimal partitioning solution to minimize edge spanning of different subgraphs. However, the proposed method requires obtaining an initial setting by bundle adjustment as the input, and the initialization problem is equally challenging.

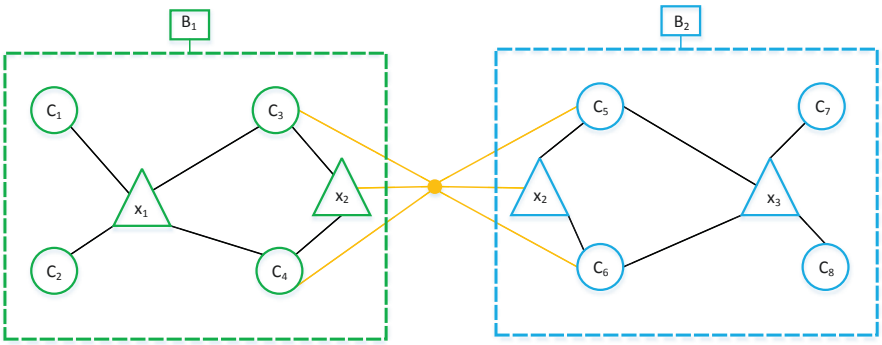


Figure 35: The optimization graph is separated to two ones. x_i and C_i represent 3D coordinate and camera parameter respectively. B_i is the introduced based node for each subgraph. The black lines and the orange lines indicate the relationships of variables in same subgraphs and across different subgraphs.

A local bundle adjustment technique is developed in [119] to greatly reduce the computational complexity comparing with traditional bundle adjustment, which makes processing long video sequences possible. The aim of the research work is targeted at simultaneously refining 3D coordinates and camera parameters for video sequence data. A triplet of images are first selected as the key frames to compute a initial 3D reconstruction. For each new frame, the image match with the last key frame is computed as well as the new camera pose. If in-

sufficient matches are obtained or the camera pose changes distinctly, the frame is added. By using the new key frame and the last two ones, a new 3D reconstruction can be computed, and afterwards the local bundle adjustment is applied to refine both the 3D reconstruction and the camera poses. Although the method can be performed in real time, it cannot produce a dense or quasi-dense reconstruction, and the resulting point clouds are fairly sparse.

In order to reduce the error accumulation in bundle adjustment, the Global Positioning System (GPS) sensor data has also been utilized [103]. By means of the GPS data, in [103] additional constraints are introduced into the original nonlinear minimization formulated by traditional bundle adjustment. In this way, the camera positions are bounded during the bundle adjustment procedure, which helps to increase the accuracy and the robustness of traditional bundle adjustment. Bundle adjustment has been also integrated and applied in other applications such as 3D reconstruction from videos. In [64], a regularized bundle adjustment is proposed to address the structure-from-motion problem. A generic model is used as a template which is morphed into the original geometry. In principle, this additional model as a sort of pre-knowledge is translated into the regularizer in bundle adjustment. In this way, the model based regularizer is integrated into the existing framework of bundle adjustment, which addresses the structure-from-motion problem efficiently without estimating camera parameters.

3.5 SUMMARY

In this chapter, the problem of point cloud refinement is studied in detail. The existing methods on outlier and noise filtering are reviewed, as well as bundle adjustment which is an important technique for multi-view data refinement. Most of these methods are purely geometric and designed for general point data or range sensor data. The possibilities of applying these methods on multi-view point data are also analyzed. For outlier filtering, these geometric approaches generally can be used for multi-view data. On the other hand, direct use of smoothing approaches on multi-view point data can be problematic. Therefore, a novel smoothing method tailored for multi-view data is developed in Chapter 4.

4 CONTRIBUTION: MULTI-VIEW POINT DATA SMOOTHING

The contribution in this chapter was published in

- Kun Liu, Rhaleb Zayer, Bundle Adjustment Constrained Smoothing for Multi-view Point Cloud Data, ISVC 2012, 8th International Symposium on Visual Computing, LNCS Volume 7431, pp 126-137, 2012.

Direct use of denoising and mesh reconstruction algorithms on point clouds originating from multi-view stereo is often oblivious to the reprojection error. This can be a severe limitation in applications which require accurate point tracking, e.g., photogrammetry.

In this chapter, a method is proposed for improving the quality of such data without forfeiting the original matches. The problem is formulated as a robust smoothness cost function constrained by a bounded reprojection error. The arising optimization problem is addressed as a sequence of unconstrained optimization problems by virtue of the barrier method. Substantiated experiments on synthetic and acquired data compare the proposed approach to alternative techniques.

4.1 INTRODUCTION

Over the last decade, *bundle adjustment* has become one of the key steps in multi-view reconstruction. It intervenes as a single nonlinear optimization which simultaneously fine-tunes the 3D structure and the viewing parameter estimates [160]. Bundle adjustment requires a set of feature correspondences which can be sparse, quasi-dense or dense in order to control the reprojection error and yields a refined visual reconstruction. In an ideal setting the resulting point cloud data would reflect the exact geometry of the original object. In practice however, several factors such as ill-textured objects, spatial discretization, structured noise, and lighting conditions contribute toward matching errors. These errors cannot be fully fixed by bundle adjustment and the point cloud generally exhibits noise to varying degrees. A commonly adopted solution is the construction of an approximating surface using existing meshing algorithms, e.g., [91, 101]. These geometric algorithms operate mainly in the three dimensional domain and do not necessarily maintain correspondences between the scene and image features. As a result, the cross-image correspondences are lost and can only be approximated by reprojection on

the surface. Although recent approaches in multiview reconstruction, e.g., [68], can improve the visual appearance tremendously, they cannot be readily used in applications such as a metrology or non-contact shape and deformation measurement where an accurate and consistent tracking of surface points over time is crucial for gathering information such as strain or parameter estimation. Furthermore, geometry processing methods which enforce bounds on the reprojection error are not concerned in the literature.

In this chapter, a point cloud smoothing approach tailored for multiview point cloud data is proposed. The problem is formulated as the minimization of a smoothness measure constrained by a bound on the reprojection error. For the former, a measure which favors local flatness of the point cloud data is proposed and for the latter, a formulation similar to standard sparse bundle adjustment is adopted. Both measures are combined into a constrained nonlinear optimization formulation. A barrier approach is used to drive the numerical optimization towards a smooth point cloud where the bounds on the reprojection error are enforced. In order to overcome numerical problems related to the densely populated nature of the arising matrix equations, the Sherman-Morisson formula is applied. This allows for addressing relatively large data sets while keeping reasonable memory requirements.

The quality of resulting point cloud data is evaluated by means of ground truth data generated synthetically. Tests on real data acquired and reconstructed using existing methods [104, 68] confirm the quality of the results. The robustness of the approach is demonstrated to irregular data sampling, to sharp features and to shrinkage. The proposed approach does not make any assumption on the nature of the noise in the data and does not require any additional input, e.g., visual hulls. The only assumption made is the geometric smoothness, which is often a property of the original model. Although, any further matching computations are not performed on the underlying images, experiments on synthetic data sets suggest that the approach moves existing matches closer towards to the exact matches.

In summary, this proposed algorithm has the following contributions:

- Formulate a constrained optimization for smoothing multi-view point clouds with bounded reprojection error
- Develop robust and efficient numerical solution procedure

The approach can be regarded as a post-processing tool and could be used in conjunction with existing reconstruction algorithms. The rest of this chapter is organized as follows: Section 4.2 covers the most related work, Section 4.3 lays out the general setup and the notation, and Section 4.4 discusses the solution of using the Laplacian operator as smoothing regularizer. In Section 4.5, the smoothing cost

function is introduced and the constrained optimization is set up. In the section, numerical aspects of the proposed approach are discussed as well. The results are summarized in section 4.6.

4.2 RELATED WORK

The prior art on multi-view model acquisition is extensive. In order to keep this exposition succinct, the most related work is reviewed and the reader is refered to [78, 147] for a general overview.

Despite its long history, most of research effort on bundle adjustment has been dedicated to numerical optimization strategies [160]. Subjects such as fusing it or enhancing it with additional input has been studied less. For instance, [63, 64] introduce model-based constraints as a regularizer within the bundle adjustment formulation in the context of head reconstruction. The authors of [98] propose using GPS (global positioning system) data as a penalty for the reprojection error and optimize the problem in the least squares sense. More closely related work is the approach proposed in [103], where GPS and Structure-from-Motion data are fused within a constrained optimization formulation. The approach is applied to a setting which combines monocular image sequence with GPS data. In this chapter, the geometric smoothness of the multi-view data is more concerned and available sensor data such as GPS data is not used.

Smoothing surface meshes [158, 44, 59, 87] and point clouds [167, 141], are well studied topics in geometry processing. Traditionally, these approaches are tailored for input data obtained from scanners. Unfortunately, many of these methods do not perform well on point clouds originating from multi-view stereo reconstruction. Furthermore they enforce the correspondences between 3D structure and image feature points.

4.3 PROBLEM SETTING AND NOTATION

In the following, the input data consists of a point cloud originating from standard multi-view acquisition [78] along with the camera parameters. The data can stem from dense [58] or quasi dense [104] matching approaches.

The point cloud will be represented as $x = (x_1^T, ..., x_n^T)$, where x_i represents the three dimensional coordinates of the i-th point. For a set of m views, the camera matrices can be conveniently assembled as $p = (p_1, ..., p_m)$ where p_j be the vector of parameter for camera j. The point corresponding to x_i on an image j will be denoted a_{ij}.

4.4 BUNDLE ADJUSTMENT WITH SMOOTHING REGULARIZATION

The overall goal of this work is to determine a smooth geometric model and the configuration of cameras that are maximally consistent with the observations. A good starting point is bundle adjustment with a regularizer.

As smooth point cloud data is sought while minimizing bundle adjustment, it seems natural to consider reformulating the problem as a minimization of the following objective function

$$h(x) + \alpha \|L(x)\|^2 \tag{55}$$

where $h(x) = \sum_i^n \sum_j^m \delta_{ij} \|Q(p_j, x_i) - a_{ij}\|^2$ is the bundle adjustment function from Equation 38 with fixed camera parameters, L is the Laplacian operator discretized locally using the k-nearest neighbors, and α is a weighting parameter. This formulation blends naturally within the standard Levenberg-Marquardt algorithm as the Laplacian operator can be simply considered as regularizer.

Unfortunately such an approach would suffer from over-smoothing as well as limitations known to Laplacian operator especially with respect to sharp features. Additionally due to the sparsity of the point cloud and the discretization based on nearest neighbors, shrinking effects can appear around holes or areas where data is missing as illustrated in Figure 40. In the following section, an alternative approach is proposed to remedy such artifacts as illustrated in the aforementioned figure.

4.5 BUNDLE ADJUSTMENT CONSTRAINED SMOOTHING

The problem is regarded as searching for a smooth surface such that reprojection error is minimal. In order to account for possible errors in the cross-image feature correspondences, the matches are allowed to evolve in a small disk around their initial location as illustrated in Figure 36. This would allow searching for a smooth surface while maintaining image feature correspondences in the vicinity of their initial positions.

In the context of an optimization for the whole point cloud data, the radius constraint are not required for the individual points. Instead, it is enforced as a global constraint. In the following, the smoothness measure is defined and the combination with the reprojection error is also shown.

It would be possible to tailor a smoothing approach which restricts the displacements within a small ball around the initial spatial point locations. This kind of smoothing however, does not take into consideration the reprojection into image space and can lead to large errors. This is illustrated in Figure 36b which shows the spatial search domain (ball around the initial point) and its counter part image space

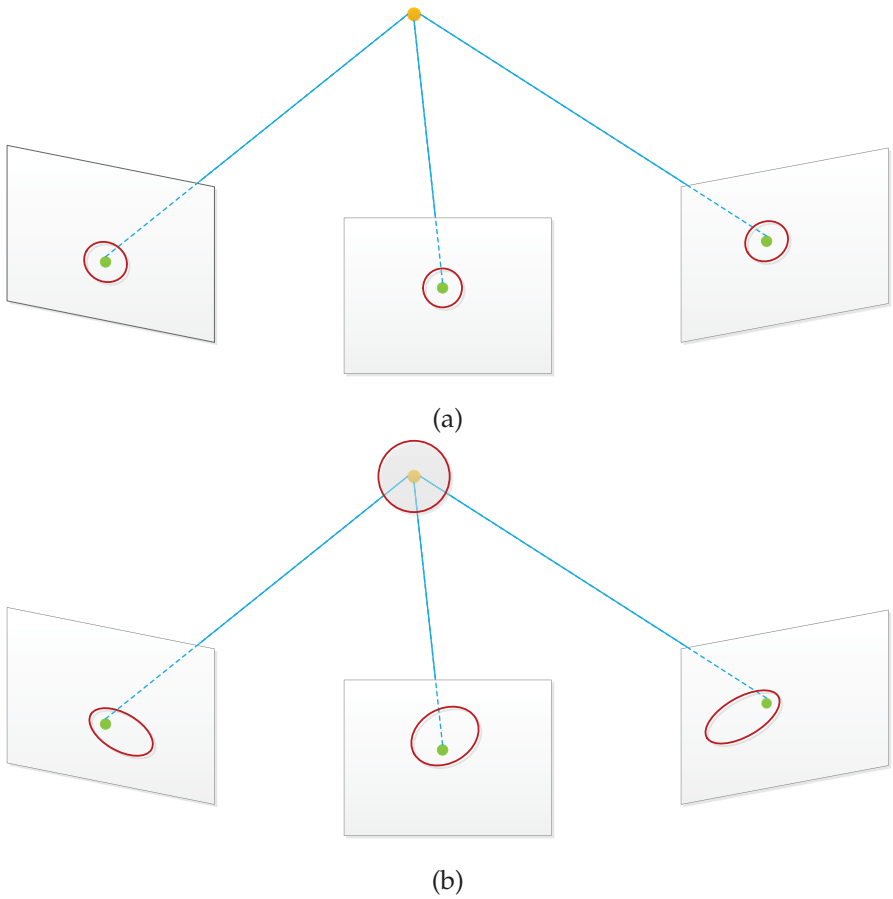


Figure 36: Starting from a converged bundle adjustment, the approach (a) searches for new spatial position of the 3D point while guaranteeing that the reprojection error is bounded i.e. the matches are maintained within a disk around the input matches. On the other hand, constraining the smoothing within a ball around the initial spatial position (b) can lead to larger reprojection errors as the shape of the corresponding projection (planar ellipses) is not taken into account (please refer to the text for details).

(planar ellipses). As the planar ellipses can be elongated, the reprojected point position can lay far from the initial match and hence such an approach would corrupt the initial matching results. This effect can be further amplified when dealing with wide base-line views. In contrast, the constraints are formulated in image space. The spatial position is then forced to lay at the intersection of the fat bundle-lines (small cylinders around the bundle lines) and thus a tight bound on reprojection error is guaranteed.

4.5.1 *Smoothness measure*

In order to define smoothness for point cloud data, the points are endowed with local adjacency relations. k-nearest neighbor algorithm

is used to construct a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the point set and an edge (x_i, x_j) is added in $\mathcal{E}$ if x_i is one of k-nearest neighbor of x_j. In all the experiments k is set to 10, using higher values also hardly changes the results. Additionally, each point is associated with a normal direction. The point normal n_i ($\|n_i\| = 1$) is estimated for every point i by the method proposed in [82], which uses principal component analysis on the local neighborhoods.

Given computed normal directions, the local planarity is defined for each edge (x_i, x_j) in $\mathcal{E}$ as

$$(n_{ij} \cdot (x_i - x_j))^2, \tag{56}$$

where n_{ij} is the average normal associated with the mid-point of edge(x_i, x_j). It penalizes the deviation of the points from the average plane defined by the midpoint and its normal. The contribution of all edges in $\mathcal{E}$ is summed up to define the global cost function

$$f(x) = \sum_{(x_i, x_j) \in \mathcal{E}} (n_{ij} \cdot (x_i - x_j))^2. \tag{57}$$

This function acts in two ways, it tends to improve the local flatness by minimizing the scalar product and second, since the edge vector is not normalized, it tends to pull neighboring points together. This local flatness measure is commonly used in the context of mesh simplification [79].

A similar formulation in the $\mathbf{L}^1$-norm has been employed in [18]. In order to avoid differentiability difficulties raised by lower order norms the $\mathbf{L}^2$ is used in view of coupling this measure with the reprojection error. In general, $\mathbf{L}^2$ responds strongly to outliers. This shortcoming is avoided by means of a robust norm

$$\Psi(s^2) = \sqrt{s^2 + \epsilon^2};\ \epsilon = 1e^{-6} \tag{58}$$

This function can be regarded as a differentiable norm of the absolute norm function and its impact is illustrated in Figure 43.

4.5.2 *Reprojection error constraint*

Bundle adjustment is assumed to be applied for the input point cloud data and let ϵ_0 be the residual reprojection. Then the our reprojection constraint is defined as

$$h(x) < \lambda \epsilon_0 \tag{59}$$

with relaxing parameter $\lambda > 1$. The constraint defined above caps the reprojection error. It also resolves cases where minima of the cost function $f(x)$ is not unique by restricting the search within a very close range.

4.5.3 *Constrained optimization*

At this stage, for the problem setup all the ingredients necessary are determined and the smoothing procedure is formulated as the following constrained optimization

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g(x) \leqslant 0 \end{aligned} \tag{60}$$

where $g(x) = h(p, x) - \lambda\epsilon_0$. The constraint function $g(x)$ depends only on the structure as the camera parameters are fixed in what follows.

This problem is more intricate than bundle adjustment alone as the cost function and the constraint are both nonlinear. It is addressed by using the barrier method [111] which is a procedure for approximating constrained optimization problems by unconstrained ones. The smoothing problem can be solved then as a series of nonlinear minimization problems of the form

$$f(x) + B(g(x), c) \tag{61}$$

where $B(g(x), c)$ is the barrier function and c is a positive constant. The barrier function operates by introducing a singularity along the constraint boundary. Probably, the popular choice is the logarithmic barrier function which tends to infinity at the constraint boundary. Furthermore, its simplicity w.r.t subsequent derivative computations and its satisfaction of the self-concordance criterion makes it an attractive choice [126]. In the current chapter, a logarithmic function is used

$$B(g(x), c) = -c \cdot \ln(-g(x)) \tag{62}$$

Concretely, solving the problem in Equation 60 amount to minimizing a series of function in the form of Equation 61 with different constant $c = c_k$, where $\{c_k\}$ is a decreasing sequence tending to 0, i.e. for each k, $c_k \geqslant 0$, $c_{k+1} < c_k$.

Each nonlinear minimization Equation 61 is solved iteratively using Newton's method. The associated Hessian matrix is

$$H = H_{f(x)} + B' H_{g(x)} + B'' \nabla g \nabla g^T \tag{63}$$

where $H_{f(x)}$ and $H_{g(x)}$ are the Hessian matrices of $f(x)$ and $g(x)$ respectively, B' and B'' are the the first and second derivatives, ∇g is the gradient of $g(x)$. In the implementation $H_{g(x)}$ is approximated by $J^T J$, where J is the Jacobian of g.

The last term in Equation 63 is a densely populated matrix and turns out to be problematic when solving the linear systems involved

at each Newton iteration. This issue is avoided by means of the Sherman-Morrison formula [133] which reads

$$(A + uv^T)^{-1} = (I - \frac{A^{-1}uv^T}{1 + v^T A^{-1} u})A^{-1} \tag{64}$$

and holds for arbitrary invertible square matrix A and vectors u and v such that $1 + v^T A^{-1} u)$ is non zero.

Writing $\hat{H} = H_{f(x)} + B' H_{g(x)}$ and $\hat{g} = \sqrt{B''}\nabla g$, by virtue of Equation 64 H^{-1} can be written as

$$H^{-1} = (I - \frac{\hat{H}^{-1}\hat{g}\hat{g}^T}{1 + \hat{g}^T \hat{H}^{-1}\hat{g}})\hat{H}^{-1} \tag{65}$$

Therefore, linear systems of the form $(\hat{H} + \hat{g}\hat{g}^T)x = b$ can be converted into

$$x = (I - \frac{\hat{H}^{-1}\hat{g}\hat{g}^T}{1 + \hat{g}^T \hat{H}^{-1}\hat{g}})\hat{H}^{-1} b \tag{66}$$

Since $\hat{H}$ is sparse and not densely populated, the system can be handled using standard linear solvers. It is imperative to note that the inverse of $\hat{H}$ need not be computed. Instead, Equation 66 is split into two subsystems $\hat{H}y = b$ and $\hat{H}z = \hat{g}$. The results are then plugged back into Equation 66.

4.5.4 *Numerical considerations*

In this section, numerical issues are discussed in details regarding constrained optimization and linear sparse solvers as a complement to Section 4.5.3.

4.5.4.1 *The Lagrange dual problem and the barrier method*

In Section 4.5.3, the smoothing problem as the constrained optimization

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g(x) \leqslant 0. \end{array} \tag{67}$$

In optimization theory [30], The *Lagrangian* associated with the problem in Equation 67 is defined as

$$L(x, \lambda) = f(x) + \lambda g(x). \tag{68}$$

In addition, the *Lagrange dual function* is defined as

$$g(\lambda) = \inf_{x \in \mathcal{D}} L(x, \lambda), \tag{69}$$

where $\inf$ is the *pointwise infimum* [30] and $\mathcal{D} = \{x \mid g(x) \leqslant 0\}$. If $\lambda \geqslant 0$ and the problem in Equation 67 has the optimal value p^*, for any $\lambda \geqslant 0$ we have

$$g(\lambda) = \inf_{x \in \mathcal{D}} (f(x) + \lambda g(x)) \leqslant \inf_{x \in \mathcal{D}} f(x) = p^*. \tag{70}$$

Therefore, the Lagrange dual function provides lower bounds for the optimal value p^*. This leads to another problem to obtain the biggest lower bound

$$\begin{aligned} \text{maximize} \quad & g(\lambda) \\ \text{subject to} \quad & \lambda \geqslant 0. \end{aligned} \tag{71}$$

The problem in Equation 71 is called the *Lagrange dual problem* associated with the *primal problem* in Equation 67. Notably, the Lagrange dual problem is a convex optimization problem, because the objective function to be maximized is concave and the constraint is convex. Any local minimum is the convex optimization problem must be a global minimum, which can facilitate solving the problem [30].

However, the formula with the pointwise infimum cannot always expressed analytically, as well as in our scenario. Therefore, we adopt the barrier method to solve the optimization as stated in Section 4.5.3. Mathematically, the barrier method solves k such problems as

$$\begin{aligned} \text{minimize} \quad & f(x) - c_k \cdot \ln(-g(x)) \\ \text{subject to} \quad & x \in \{x \mid g(x) \leqslant 0\}, \end{aligned} \tag{72}$$

where $c_k > 0$ and $\{c_k\}$ is a sequence tending to zero. In our implementation, we choose $k = 1, 2, ..., 8$, $c_1 = 1000$, and $c_{k+1} = c_k/10$. If in certain step of the iteration, the decrease of the value of the objective is less than 1% with respect to the value of the objective after last iteration, the iteration will be terminated.

4.5.4.2 *Linear sparse solvers*

Many linear sparse systems need to be solved in the nonlinear optimization as discussed in Section 4.5.3. Hence, it is important to select an efficient linear sparse solver for our implementation. In general, two categories of solvers are available, namely, direct solvers [43] and iterative solvers [137]. Since the matrices in our linear sparse systems are symmetric and positive definite, Cholesky factorization based method [43] and preconditioned conjugate gradient [137] are candidates respectively for direct solvers and iterative solvers. A comparison is conducted using the linear sparse systems with different sizes in our scenario, and different implementations of the same solvers are tested in order to avoid bias as shown in Figure 37. CHOLMOD [37] and Intel MKL PARDISO [1] are direct solvers using Cholesky factorization. Two implementations of preconditioned conjugate gradient

respectively from Eigen [76] and Intel MKL [1] are tested. The backslash in MATLAB [115] is examined as well. By the figure, for our scenario preconditioned conjugate gradient outperforms Cholesky factorization based method.

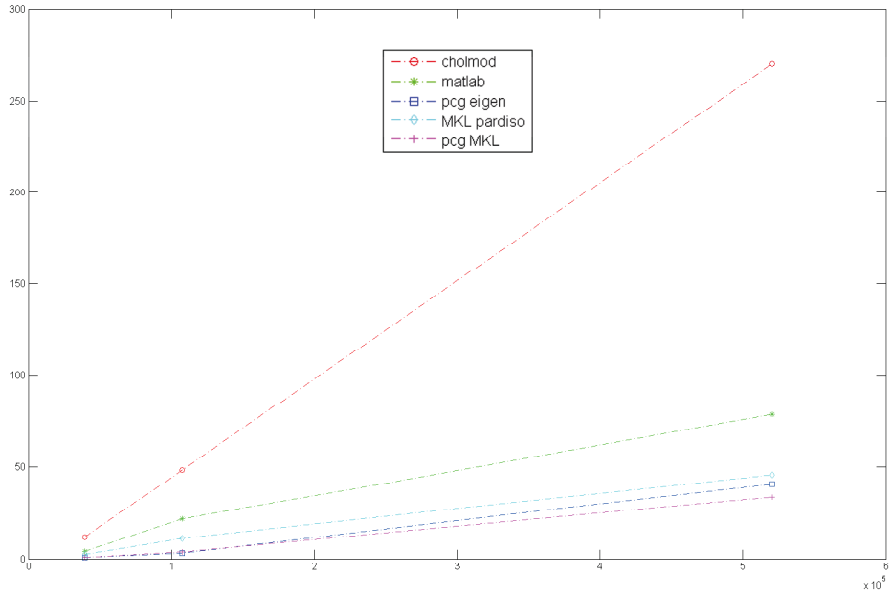


Figure 37: A comparison of different linear sparse solvers: x axis is the size of the linear sparse system and y axis is the solving time in seconds.

4.6 EXPERIMENTAL RESULTS

The smoothing method, namely, bundle adjustment constrained smoothing, for multi-view point data is proposed in last section. The problem is formulated as a nonlinear optimization with constrains defined by the reprojection error originating from bundle adjustment. In this section, the proposed method is analyzed and evaluated by using synthetic and real world data. A comparison is also conducted between the method and the straightforward solution, i.e., bundle adjustment with smoothing regularization, presented in Section 4.4.

4.6.1 *Synthetic data preparation*

Synthetic data is extensively used to help researchers with analyzing algorithms in computer graphics and computer vision. Since ground truth is available in synthetic data, algorithm performance can be assessed objectively.

In our experiments, synthetic data is used and the data is generated by means of Blender [61] which is an open source 3D creation suite. Blender integrates features including 3D modeling, texture mapping, camera tracking, rendering and so forth. In the experiment, it is used

to simulate capturing multiple images of same scenes by cameras from different views. The entire process of simulation can be operated automatically with the help of Blender Python API.

As shown in Figure 38, a 3D model is imported into Blender and displayed in the center of the workspace. The lighting is set up, as

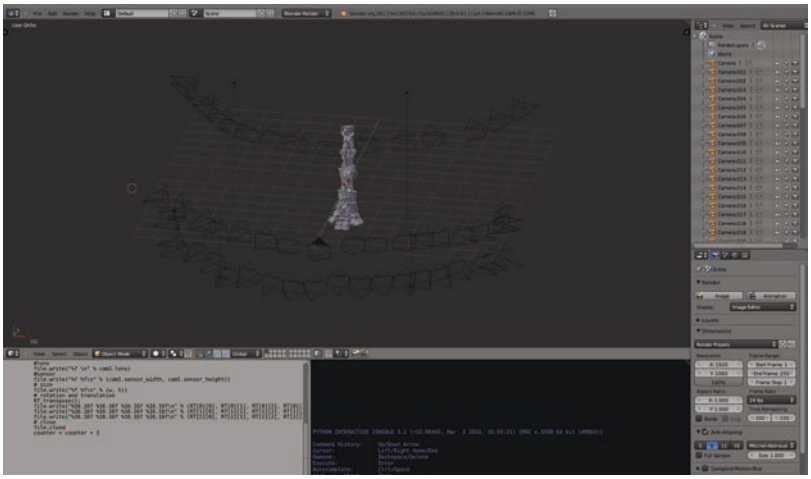


Figure 38: The user interface of Blender [61] displays using multiple cameras to capture a 3D scene.

well as multiple cameras in different locations towards to the model. The camera parameters discussed in Section 2.2 can be conveniently modified in Blender to produce rendered images. These parameters include position, orientation, focal length, image width, image height and sensor size. Sensor size equals pixel width or/and height under the assumption of square pixels. The camera capture simulation results in multiple images from different views. To generate synthetic data, 3D points of the model are reprojected on each image to compute image correspondences. Notably, the obtained synthetic data is noise-free and all the ground truth is available for experiments. In order to imitate the high frequent errors ubiquitously existing in real data, Gaussian noise [42] is artificially introduced into synthetic data via two ways:

- Noise is directly added to the image correspondences (Figures 43, Figure 47).
- Noise is added to the original model and the image correspondences are recomputed as weighted average of the noiseless and the noisy projections (Figure 42).

4.6.2 *Comparison*

In Sections 4.4, the simple solution, i.e., bundle adjustment with smoothing regularization is proposed by introducing an additional regularization term into the cost function of bundle adjustment. The reg-

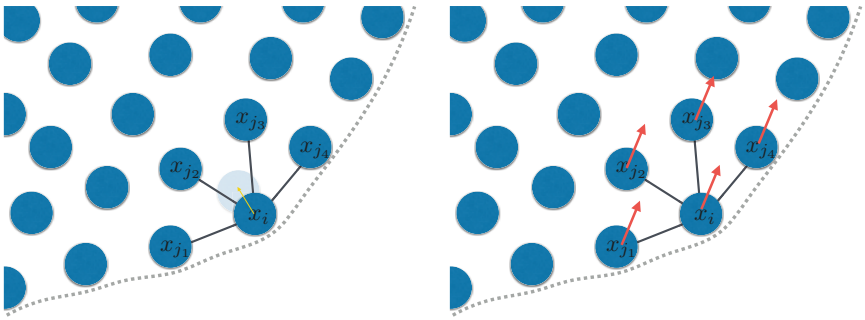
ularization term is defined using the Laplacian operator to smooth data. On the other hand, the method proposed in Section 4.5, i.e., bundle adjustment constrained smoothing, is more sophisticated. It formulates the smoothing problem as a nonlinear optimization with constrains defined using the reprojection error. In this section, the two approaches are analyzed and compared.

4.6.2.1 *Shrinkage and over-smoothing*

In bundle adjustment with smoothing regularization, the Laplacian regularization term is $||L(x)||^2$ which can be written as

$$||L(x)||^2 = \sum_i ||x_i - \frac{\sum_j x_j}{N_i}||^2, \tag{73}$$

if L is chosen as the discrete Laplacian proposed in [158], where N_i is the number of x_i's neighbors. Since the simple solution smooths data by minimizing the objective function Equation 55, x_i tends to be moved towards to the barycenter of its neighbors via the optimization. If x_i locates close to the area where data is partially missing as illustrated in Figure 39a, the shrinkage problem is inevitable, which has been mentioned in Section 4.4. This is a significant disadvantage of Laplacian regularization.

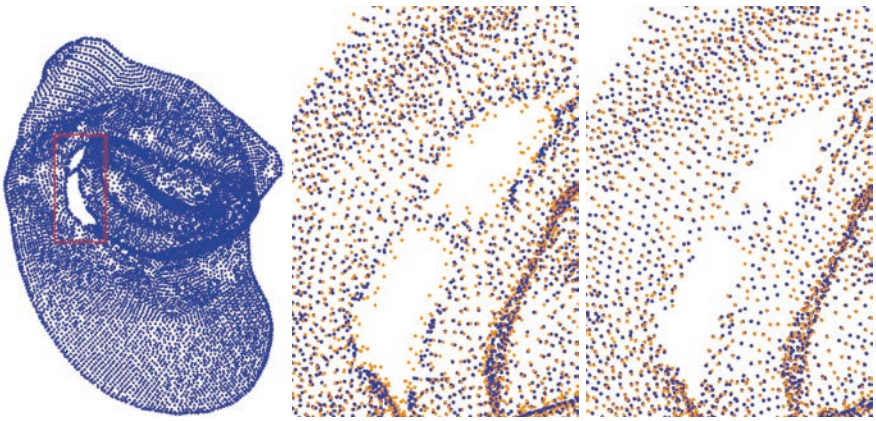


(a) (b)

Figure 39: (a): Laplacian regularization moves x_i towards the barycenter of its neighbors during the minimization procedure. (b): Using normals (red arrows) allows x_i not necessarily to move towards the barycenter in order to achieve the optimal position.

On the other hand, bundle adjustment constrained smoothing successfully prevents the shrinkage problem by introducing normals into the objective function Equation 57. The effect of normals is illustrated in Figure 39b. If x_i has a similar normal as its neighbors, the optimal x_i for the term $(n_{ij} \cdot (x_i - x_j))^2$ can be obtained by moving along the normal. The experiment demonstrates the theoretical analysis as well. As shown in Figure 40a, two holes are present in the point cloud. Such scenario is fairly common in multi-view reconstruction, because only visible parts can be reconstructed from images and cam-

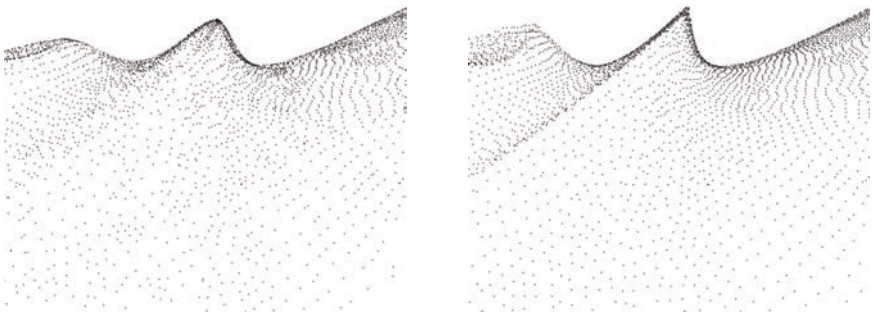
eras are often not enough to cover the whole object. The point cloud is used to test the two methods respectively, namely, bundle adjustment with smoothing regularization and bundle adjustment constrained smoothing. As shown in Figure 40c and Figure 40b, bundle adjustment constrained smoothing addresses the data missing issue and avoids the shrinkage problem but bundle adjustment with smoothing regularization fails.



(a) (b) (c)

Figure 40: A zoom on the ear model (a) illustrates the shrinking effect of Laplacian regularization (see blue point cloud in (b)). Constrained smoothing is more robust to such artifacts (see blue point cloud in (c)). In both results, the original data is shown in orange.

Due to the same reason as the shrinkage, bundle adjustment with smoothing regularization tends to over-smooth data as illustrated in Figure 41a. In contrast, bundle adjustment constrained smoothing manages to preserve the sharp features as shown in Figure 41b.



(a) (b)

Figure 41: (a) illustrates the over-smoothing issue of Laplacian regularization, whereas bundle adjustment constrained smoothing doesn't suffer from this problem as shown in (b).

4.6.2.2 *The reprojection error*

An objective of this chapter is to propose a method to improve the quality of multi-view point data without impairing image correspondences. The quality of image correspondences can be measured using the reprojection error from bundle adjustment as discussed in Section 3.4. However, in real world data exactly precise image correspondences are impossible to obtain, therefore, synthetic data is used to examine algorithm performance with respect to the reprojection error.

To prepare the synthetic data, we follow the way presented in Section 4.6.1. In the synthetic data, the artificial Gaussian noise is added to the 3D point cloud and the image correspondences are recomputed as weighted average of the noiseless and the noisy projections. Notably, the ground truth of image correspondences is also available conveniently.

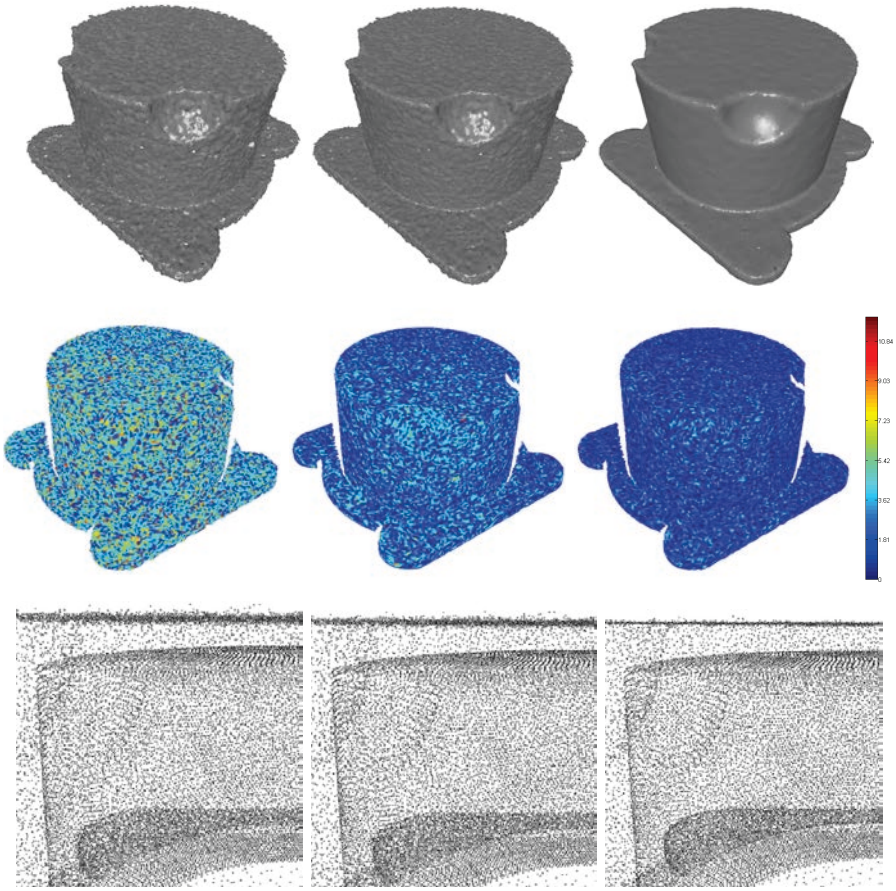


Figure 42: The figure shows a comparison using the Deckel dataset. The noisy point cloud (left-top) is processed using bundle adjustment with Laplacian regularization (middle-top) smoothing and bundle adjustment constrained smoothing (right-top), all views are shown in splating mode. The middle row shows the reprojection error for the same view. The bottom row shows a zoom on the corresponding point cloud data.

A synthetic noisy point cloud is displayed in Figure 42 (left-top). Figure 42 (middle-top) and Figure 42 (right-top) show the smoothed results by applying bundle adjustment with smoothing regularization and bundle adjustment constrained smoothing respectively. Point clouds are rendered using the splatting technique [27] which is easier to visualize noise than plotting massive dots. Apparently, bundle adjustment constrained smoothing results in a smoother point cloud. Three images in the middle row of Figure 42 depict the reprojection errors with respect to the point clouds in the first row. The colormap encodes the errors, concretely, red represents a high value and blue represents a low value. The result (right-middle) by bundle adjustment constrained smoothing indicates lower reprojection errors. Surprisingly, the result (right-middle) even has less errors than the original data although it has a smoother appearance. This demonstrates that our optimization formulation in Equation 60 is reasonable for the problem. In the bottom row, three zoom views of the three point clouds show that noise on the top of the object is greatly filtered out by bundle adjustment constrained smoothing.

4.6.2.3 *Summary*

In a word, bundle adjustment constrained smoothing outperforms bundle adjustment with smoothing regularization in terms of preventing shrinkage and over-smoothing, and preserving low reprojection errors, though the latter solution can be easy to adapt to the Levenberg-Marquardt agorithm and implement as discussed in Section 4.4.

4.6.3 *Robust norm*

In bundle adjustment constrained smoothing, the objective function employs a robust norm approximating $\mathbf{L}^1$-norm in order to preserve sharp features after smoothing. According to [18], $\mathbf{L}^1$-norm can effectively reduce overfitting caused by outliers and thus preserve features. Since $\mathbf{L}^1$-norm is resistant to outliers in data, it is robust.

We also test the different performances using $\mathbf{L}^2$-norm and the robust norm as shown in Figure 43. The original point data is generated synthetically as discussed in Section 4.6.1. The artificial noise is directly added in the image correspondences. The ground truth of the object geometry is available. Hence, we can compute the Hausdorff distances [39] between the true object and results using $\mathbf{L}^2$-norm and the robust norm respectively. In Figure 43, the colormap encodes the Hausdorff distance. Red indicates a high value and blue indicates a low value. As you can see in the figure, the result using the robust norm has lower distances, which means it is closer to the truth.

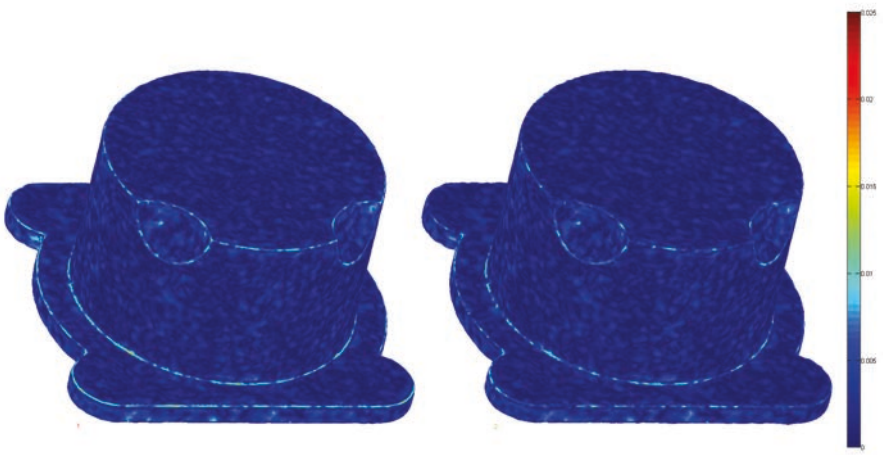


Figure 43: The result of the proposed approach on the noisy input Deckel dataset using $\mathbf{L}^2$-norm (left) and the robust norm (right). The color coding shows the Hausdorff distance to the ground truth point cloud.

4.6.4 *Other results*

Bundle adjustment constrained smoothing is tested on the two synthetic data sets as shown in Figure 44 and Figure 45. Both data sets are generated using the way stated in Section 4.6.1 and an artificial Gaussian noise is added into the image correspondences. The former point data consists of about 50K points and is reconstructed from 11 images, while the latter point data consists of about 90K points and is reconstructed from 10 images. The point data is rendered by means of the splatting technique [27].

Figure 47 shows the performance of the proposed approach on a larger synthetic data set. 56 cameras are set up in Blender and a Gaussian noise with a unit variance and a peak 3 is introduced into the image correspondences across the views. In Figure 47, left displays the original noisy data and right shows the result by bundle adjustment constrained smoothing. Two close-up views with respect to the same data set are shown in Figure 48.

A typical result of the approach on real world data is shown in Figure 49. The splatting rendering is applied for the visualization as well. In this example, 6 views were combined using the quasi-dense propagation approach of [104] to generate the initial point cloud. The detailed multi-view setup is presented in Section 2.2.4. The approach reduces the asperities in the point cloud and yields a smoother result. Even in regions such the eye area which are generally difficult to construct, the proposed approach helps smoothing out the noise to a great extent. Lastly, Table 2 summaries dataset sizes and time-costs with respect to the experiments in Section 4.6. All the experiments

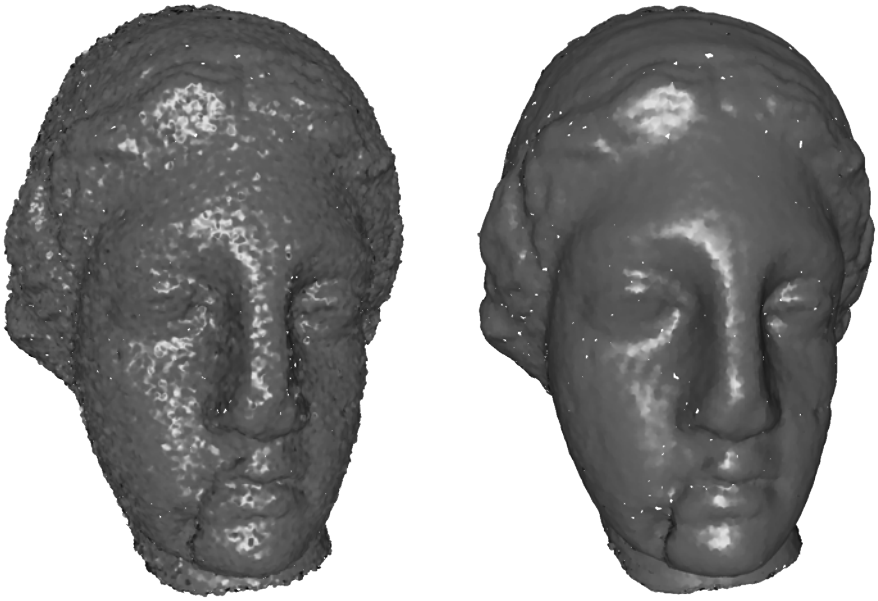


Figure 44: Illustration of the proposed method on the Venus dataset (about 50K points). Image correspondences across 11 views were perturbed by a Gaussian noise which yields the noisy reconstruction (left). The result of the proposed approach is shown to the right. All views are shown in splatting mode.

Figure 45: Illustration of the proposed method on the Julius dataset (about 90K points). Image correspondences across 10 views were perturbed by a Gaussian noise which yields the noisy reconstruction (left). The result of the proposed approach is shown to the right. All views are shown in splatting mode.

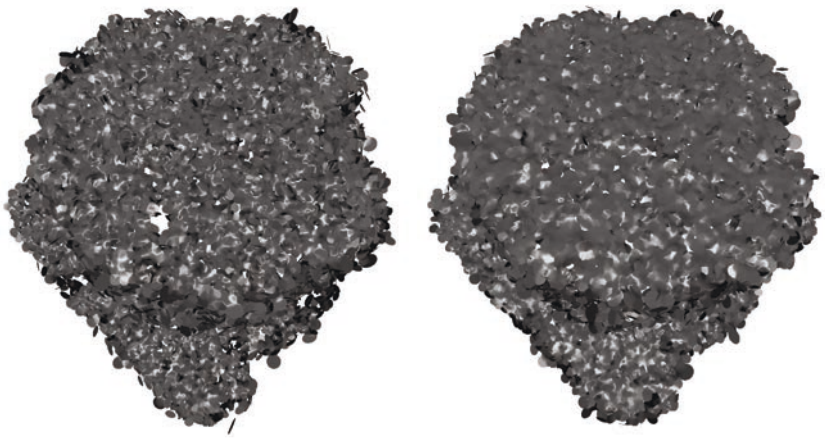
are conducted in a Mac laptop with a 1.4GHz dual-core CPU and 4GB memory.

Model	# Points	# Views	Time (seconds)
Deckel (Figure 42 right-top)	35491	16	381.31
Venus (Figure 44)	49164	11	460.82
Julius (Figure 45)	90793	10	816.59
Statuette (Figure 47)	162156	56	2291.49
Shizhe (Figure 49)	61063	6	542.51

Table 2: The statistics of the experiments performed in a Mac laptop with a 1.4GHz dual-core CPU and 4GB memory

4.6.5 *Discussion*

The proposed approach, i.e., bundle adjustment constrained smoothing, requires a good initial estimate, e.g., results from bundle adjustment and also a sufficient sampling of the data in order to yield optimal results. However, when sampling points are missing in some region, as illustrated in Figure 40, the approach does not suffer from shrinkage and still produces coherent results. When the initial data



(a) (b)

Figure 46: (a): The synthetic data suffer a fairly large noise on the image correspondences and thus also has a very noisy 3D appearance. (b): The proposed method produces a result slightly smoother than the original data but cannot fully recover the true geometry. All views are shown in splatting mode.

suffers from large noise, the approach can improve the results but only within the limits allowed by the reprojection error control. In this scenario, the noise reflects extensive errors in the matching, projection matrices or both and therefore enforcing reprojection errors based on such corrupt data might not be a viable goal. Nevertheless,

relaxing the reprojection error bound would induce smoother geometric results as shown in Figure 46.

Figure 47: Illustration of the proposed method on the large Statuette dataset (200K points). Image correspondences across 56 views were perturbed by a Gaussian noise with a unit variance and a peak of 3 which yields the noisy reconstruction (left). The result of the proposed approach is shown to the right. All views are shown in splatting mode.

4.7 SUMMARY

In this chapter, the effect of combining a smoothness measure with bundle adjustment is studied. To overcome limitations of using simple strategies such as Laplacian regularization or constrained spatial smoothing, a robust and efficient approach is developed based on containing the reprojection error while enforcing the smoothness of the

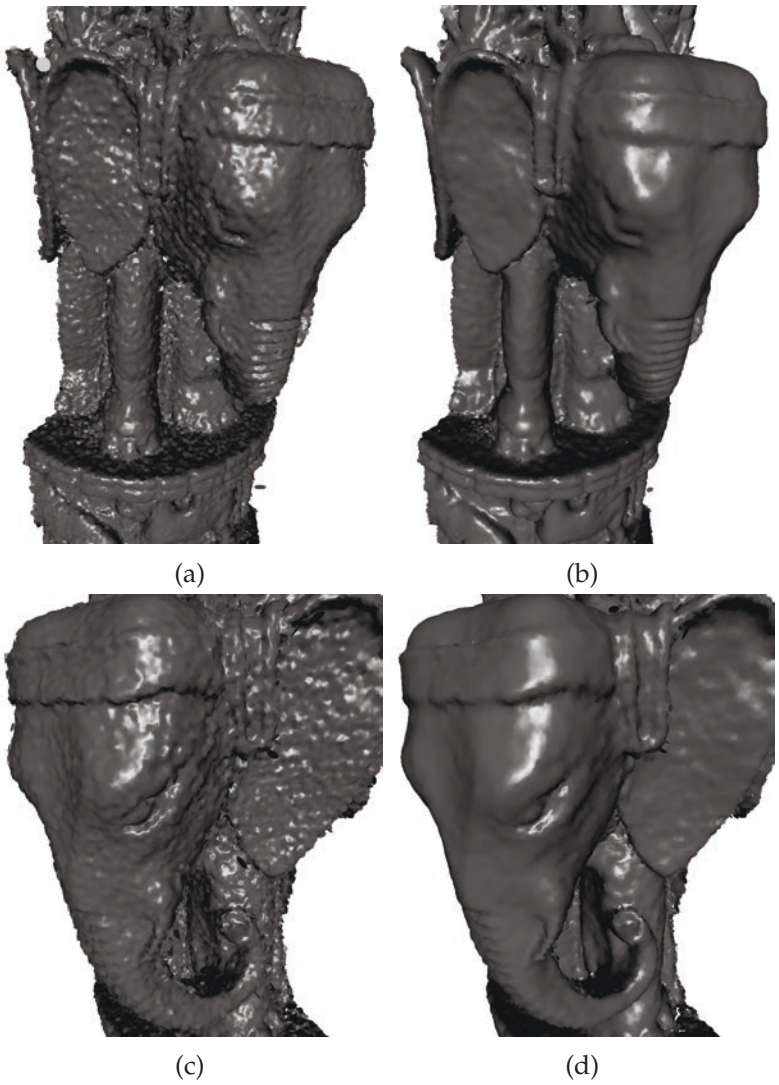


Figure 48: Two close-up views on the elephant head of Figure 47 are displayed. The noisy data is displayed in left. The results of the proposed approach is shown in right. All views are shown in splatting mode.

point cloud data. In future work, the use of the proposed approach on time-dependent data will be investigated.

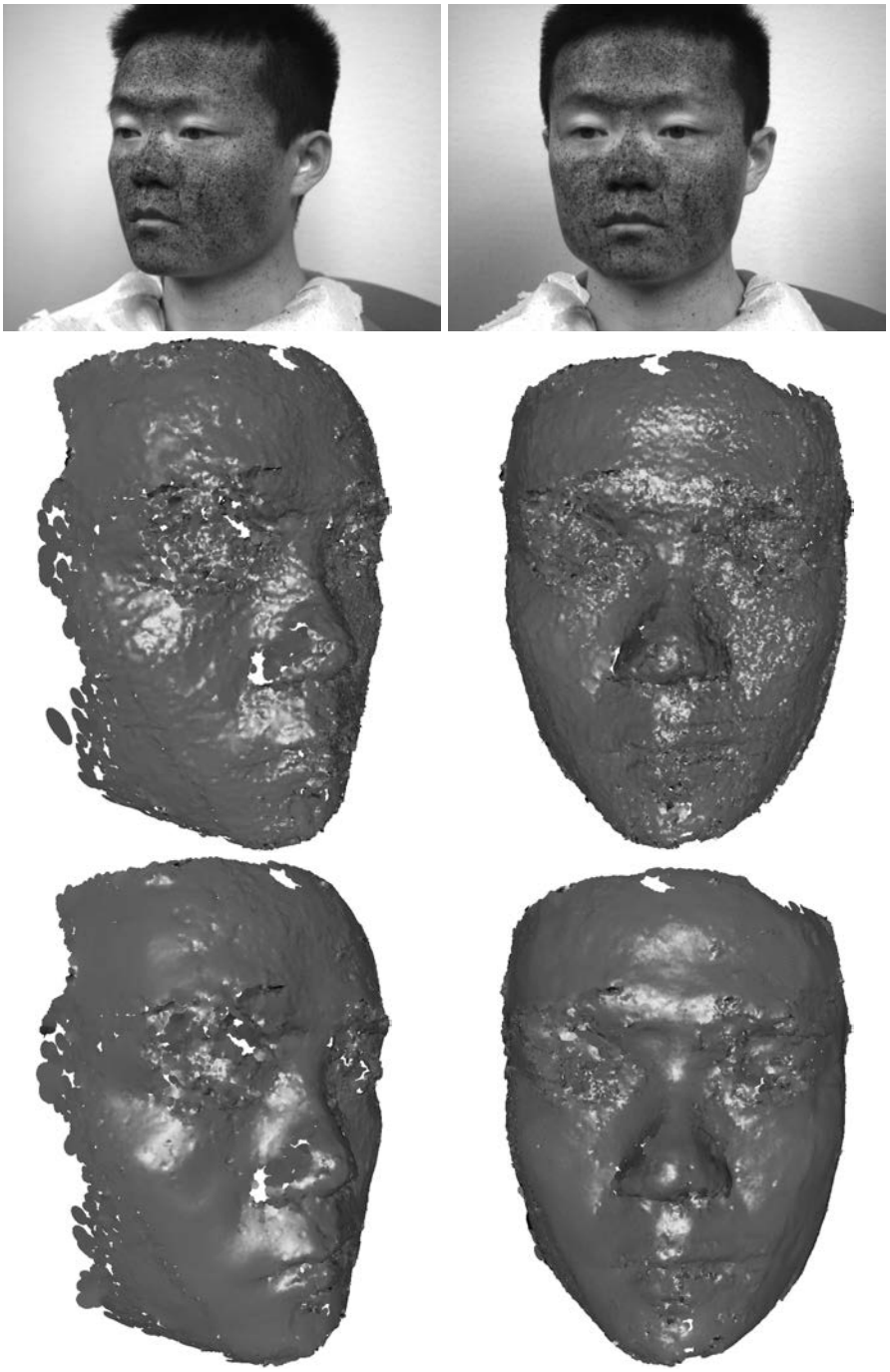


Figure 49: The Shizhe dataset: Sample images (the first row) out of a set of 6 wide base-line images were used to generate a quasi-dense point cloud (the second row) using the propagation approach in [104]. The result of the proposed method (the third row) shows an overall quality improvement of the point cloud. Point clouds are shown in splatting mode.

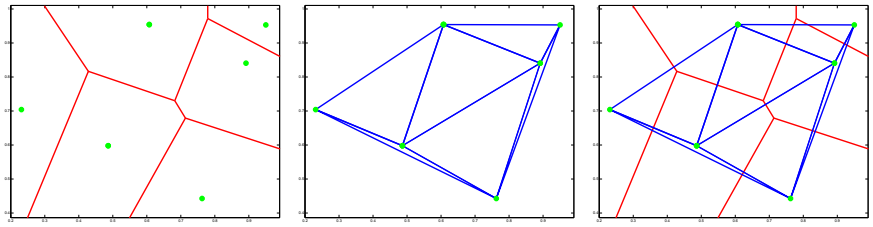
5 STATE OF THE ART IN SURFACE RECONSTRUCTION

3D reconstruction is widely applied in both academia and industry. An important reason is the fact that many methods can be utilized to acquire objects in the real world, e.g., image-based algorithms [147, 68], kinect [123] and the scanning system presented in [8]. Generally, these techniques yield a point cloud to represent the original object. Although several point cloud based methods have been proposed for data processing [175] and rendering [7], surface reconstruction remains a standard procedure in industry. Mesh representation is required in many applications such as visualization, reverse engineering and medical imaging. Furthermore, in contrast to point cloud based methods, mesh processing methods are well established [158, 107, 84], and the mesh representation blends seamlessly into existing mainstream rendering APIs such as OpenGL [92] and Direct3D [40]. In this chapter, a survey of previous surface reconstruction methods is presented and a novel surface reconstruction algorithm is proposed in Chapter 6.

5.1 INTRODUCTION

In computer graphics, surface reconstruction has been studied for over thirty years. A Delaunay based algorithm for surface reconstruction was first proposed in [23] and the topic of surface reconstruction started to become popular after the works [82, 81]. Surface reconstruction is still an active research area and the large body of work on the subject keeps growing every year. These efforts can be mainly classified into three categories [5, 140], i.e., computational geometry methods, implicit surface methods and machine learning methods.

Generally computational geometry methods have strict mathematical proof of convergence under certain sampling conditions. These guarantees are beneficial theoretically, however, sampling conditions such as $(\epsilon, \delta, \kappa)$-sample proposed in [45] are too rigorous to satisfy in practice. In reality, noise and outliers in the original point clouds can easily cause failures to satisfy those sampling requirements. Implicit surface methods are more practical and common which compose the largest branch in surface reconstruction methods. Once an implicit surface is determined by local fitting, e.g., *moving least squares* (MLS) [130], the mesh can be typically computed by means of marching cubes [109]. This group of methods generally produces smooth results, and are generally robust to noise. Moreover, several works



(a) (b) (c)

Figure 50: 2D example illustrations: (a) displays the Voronoi diagram of the six points (green); (b) illustrates the Delaunay triangulation of the same points as (a); (c) shows the Voronoi diagram (red) in (a) and the Delaunay triangulation (blue) in (b) together in the same figure.

based on the theory of machine learning were developed as well, e.g., [172, 85, 86], which proposed a distinct direction to address the problem.

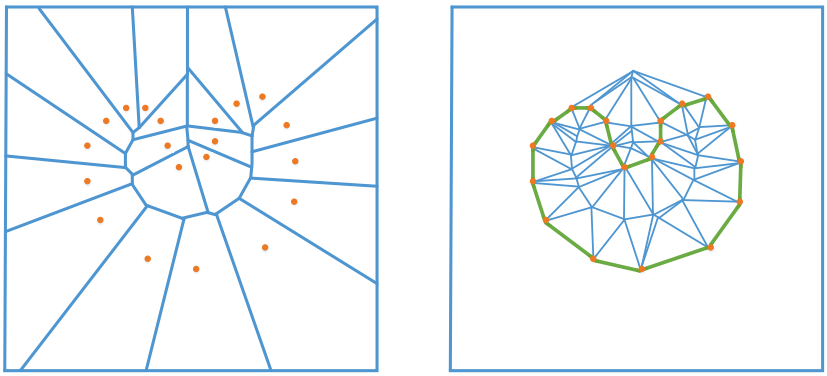
As a way of surface reconstruction, mesh generation is ubiquitously applied in numerical simulation [108, 149, 145]. Finite element method in numerical simulation requires discretizing a continuous domain into a set of discrete sub-domains, e.g., triangle mesh. A noticeable difference between mesh generation in computer graphics and numerical simulation is the inputs being meshed. Usually mesh generation in computer graphics constructs a mesh from a point cloud, whereas a continuous domain is approximated by a generated mesh in numerical simulation. Another distinction is different points concerned during the process of mesh generation. Finite element method prefers to avoid generated mesh containing "skinny" triangles. The triangles with this shape can introduce the problem of numerical unstability [62]. On the other hand, since point cloud is provided as the input in computer graphics, the information about real surface is partially missing. Therefore, the resulting mesh is required to unbiasedly represent the unknown surface, e.g., preserving features.

5.2 COMPUTATIONAL GEOMETRY RECONSTRUCTION METHODS

Computational geometry methods are based on the theories of *Voronoi diagram* and *Delaunay triangulation*. In computational geometry, Voronoi diagram and Delaunay triangulation are well studied topics and mature theory has already been established [129]. Let $P = \{p_1, p_2, ..., p_m\}$ be a set of points in Euclidean space $\mathbb{R}^n$, the *Voronoi region* of p_i is defined as

$$V(p_i) = \{x \| p_i - x| \leqslant |p_j - x|, \forall j \neq i\}. \tag{74}$$

Figure 51: The yellow curves are the medial axis of the green curves.



(a) (b)

Figure 52: A Voronoi diagram of a point set P (orange) is shown in (a). The Delaunay triangulation of $P \cup V$ is illustrated in (b), and the green line represents the crust, where V is the set of Voronoi vertices.

Voronoi diagram is simply $\{V(p_i)\}$ which forms a partition of $\mathbb{R}^n$ (see Figure 50a). A *Delaunay triangulation* for the point set P is a triangulation which satisfies that no point in P is inside the circumcircle of any triangle. Figure 50b depicts an example of Delaunauy triangulation. The Voronoi digram and Delaunay triangulation are dual to each other in the sense of graph theory as illustrated in Figure 50c. In 2D plane, Delaunay triangulation maximizes the minimum angle of triangles. This is highly desirable, because this property can prevent to generate skinny triangles. The triangles with this bad shape always introduce potential troubles in applications, e.g., finite element method [57].

5.2.1 *Crust algorithms*

Amenta et al. [12] demonstrated a curve reconstruction method using Voronoi diagram and Delaunay triangulation. In the method, the so-called *crust* is introduced to reconstruct smooth curve. The crust is a graph of planar points and a subset of Delaunay triangulation (see Figure 52b). The intuition behind the definition of crust is that the Voronoi vertices V approximate the *medial axis* of the original curve

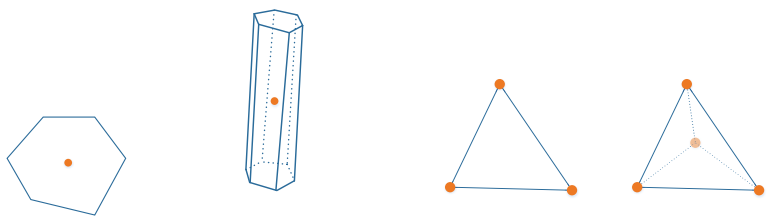


Figure 53: a): 2D Voronoi region; (b): 3D Voronoi region/cell; (c): A triangle in 2D Delaunay triangulation; A tetrahedron in 3D Delaunay triangulation.

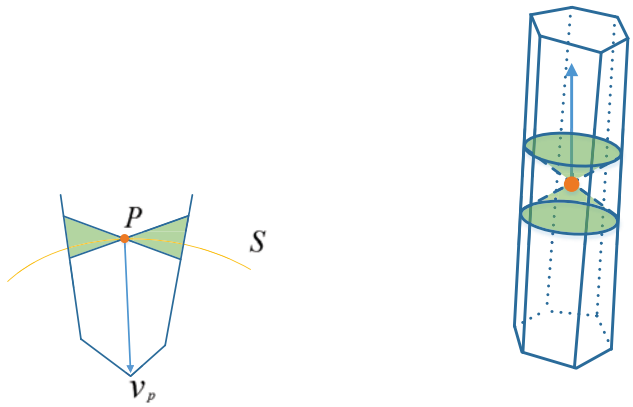
as shown in Figure 52a. The *medial axis* of a curve S is defined as the closure of the set of point in the plane which have two or more closest points in S [22]. An example of medial axis is illustrated in Figure 51. Noticeably, the yellow medial axis encapsulates general geometry of the green curve. Since the *Voronoi vertices* V is an approximation of the medial axis, the reconstruction can be facilitated as long as V is computed, which is datailedly discussed in the immediate paragraph.

An illustration of the algorithm is depicted in Figure 52. Let P is a point set sampled along a curve in $\mathbb{R}^2$, i.e., the orange dots in Figure 52a, the problem arises as reconstructing a polygonal approximation of the curve from the point set P. The approach proposed in [12] first calculates the Voronoi diagram of the point set P as shown in Figure 52a. The *Voronoi vertices* are acquired naturally, i.e., those points are equidistant to three or more points in P, and the set of Voronoi vertices is denoted as V. Then a Delaunay triangulation of $P \cup V$ is computed as illustrated in Figure 52b. A crust can be easily extracted from the edges of the triangles, where the crust edges only use points in P as their ends (see green edges in Figure 52b). The extracted crust is regarded as an approximation of the curve represented by P. By this way, a 2D curve can be reconstructed from a point cloud based on Voronoi diagram and Delaunay triangulation.

5.2.2 *Cocone algorithms*

Amenta et al. [13]. proposed the cocone algorithm which generalized the 2D crust algorithm [12] to surface reconstruction in $\mathbb{R}^3$. In the 3D problem, a Voronoi region is a 3D cell and triangles in curve reconstruction are changed to tetrahedrons as shown in Figure 53.

In the cocone algorithm, first a set of candidate triangles is selected by using *cocone* and then a piecewise-linear manifold mesh is extract from the set. The shapes of cocone in 2D and 3D are respectively illustrated in Figure 54a and 54b. Once the Voronoi cells of samples are computed, the normals can be estimated using *poles* introduce in [11]. For each Voronoi cell, the Voronoi vertice farthest from the sample is defined as a pole. As illustrated in Figure 54a, V_p is a pole



(a) (b)

Figure 54: The cocone for a sample in $\mathbb{R}^2$ (a) and $\mathbb{R}^3$ (b). In (a) the cocone is in green, while the boundary of cocone is colored with green in (b). The blue bounded region is the Voronoi cell of the red sample.

and the vector $\overrightarrow{PV_p}$ is an approximation of the normal at sample P. For an angle θ, the *cocone* is defined as the complement of the double cone with apex P making an angle $\pi/2 - \theta$ with the axis that is aligned with the estimated normal at P as shown in Figure 54.

Using the computed cocone, a set of triangles T can be obtained from Delaunay triangulation by filtering. In addition, it is able to be proved that a piecewise linear manifold can be selected from the set T. Furthermore, the cocone algorithm guarantees that under ϵ-sampling condition the output is homeomorphic to the original surface. It is the first such topological guarantee for the reconstruct problem. This means the reconstructed surface has the same topological properties as the original one such as genus and fundamental group [114]. The rigorous mathematical proof is provided in [13].

A comprehensive discussion on cocone are presented in the book [45]. The general cocone algorithm [13] gives rise to a series of subsequent work. To produce a water tight model, the tight cocone method is suggested [46]. Additionally the super cocone approach [50] is proposed to handle very large data. Unexceptionally, like other computational geometry methods the cocone methods can be significantly effected by noise. In order to address this problem, the robust cocone algorithm is studied [47]. Lately, several methods are proposed to preserve feature lines by detecting boundaries, sharp ridges and corners, e.g., [52, 49]. With same purpose, several other methods are also demonstrated but using totally different techniques. In Section 5.2.5, a hybrid method using both Delaunay triangulation and graph cuts is discussed in detail.

5.2.3 *Convection driven methods*

Allegre et al. [9] developed a convection driven method based on the work [35]. It extracts a triangulation from a Delaunay triangulation of an input point cloud and simultaneously performs a mesh simplification. The simplification can be operated interactively by users or automatically by using prescribed sampling constraints.

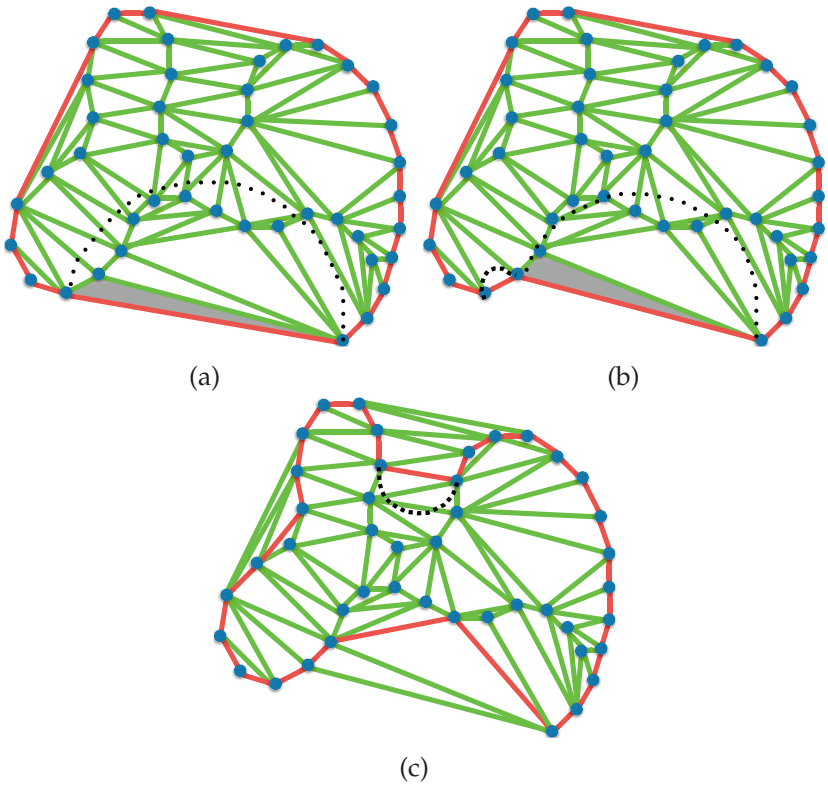


Figure 55: An illustration of geometric convection towards a 2D point set. (a) and (b): The reconstructed curve is initialized as the convex hull of the input points and the curve is updated by using the orientated Gabriel property. (c): The current edge cannot be updated towards the input points in a concavity.

As many other computational geometry methods, the method [35] first computes a Delaunay triangulation of an input points and extracts a mesh from the Delaunay triangulation. The input point cloud is supposed to be sufficiently dense to satisfy the ϵ-sample condition [11]. Figure 55 is a simple illustration of the geometric convection method [35] on 2D points. The reconstructed curve (mesh in 3D) is initialized as the convex hull of the input points (see the red curve in Figure 55a). The initialized curve is updated towards the input points by means of the *orientated Gabriel property*. An orientated Delaunay facet meets the orientated Gabriel property if and only if the half of the minimum enclosing sphere of the facet does not contain any points from the input point cloud. For example, in Figure 55a the bottom edge (red) of the Delaunay triangle (gray) is examined. Its

half minimum enclosing circle (dashed) contains a few input points and thus the edge does not meet the orientated Gabriel property. The gray Delaunay triangle is removed and the red curve is updated as shown in Figure 55b. Two minimum enclosing circles are displayed in Figure 55b. The associated Delaunay triangle of the shorter edge is maintained as the edge meets the orientated Gabriel property. Occasionally the convection can be stopped preliminarily in concavities such as the case shown in Figure 55c. This problem can be addressed by comparing the triangle facet normal and associated vertex normals. If the normals are very different, the associated Delaunay triangle of the facet is removed and the convection is continued. The convection stops until all facets meet the orientated Gabriel property and none of them block in concavities.

The method [9] initializes the reconstructed result by computing the convex hull of the input point cloud as the work [35]. However, it does not compute Delaunay triangulation explicitly. In the method [9], a Delaunay tetrahedron is created locally using the facet and the chosen point from the half Gabriel sphere to maximizes the radius of the circumsphere. Moreover, redundant points are removed around the chosen point using the two following criteria

$$|n(p_i) \cdot n(p)| > \rho \quad \text{and} \quad |n(p_i) \cdot \frac{p - p_i}{\|p - p_i\|}| < \rho', \tag{75}$$

where n is a normal and p_i is a neighbor point of p.

5.2.4 *Scale space methods*

Digne et al. proposed an approach [54] integrating the *mean curvature motion* (MCM) and the *ball-pivoting* method [20]. The approach is developed based on a scale space strategy and consists of three steps listed as follows:

1. Use MCM to smooth the input point data.
2. Perform the ball-pivoting algorithm to generate a mesh.
3. Back project the mesh onto the original points.

In [54], the mean curvature motion, or the intrinsic heat equation, is used to smooth the input point cloud. It has the mathematical form as

$$\frac{\partial p}{\partial t} = H(p)\vec{n}(p), \tag{76}$$

where $H(p)$ and $\vec{n}(p)$ are respectively the mean curvature and the normal at p. This motion is a series of projections of each point on the regression plane determined by its radical neighborhood. In addition, each point moves at the speed of the mean curvature in the nor-

mal direction. After smoothing the input point data using the mean curvature motion, a standard ball-pivoting method [20] is applied to create a mesh. The ball-pivoting method is a tipycal *advancing front*

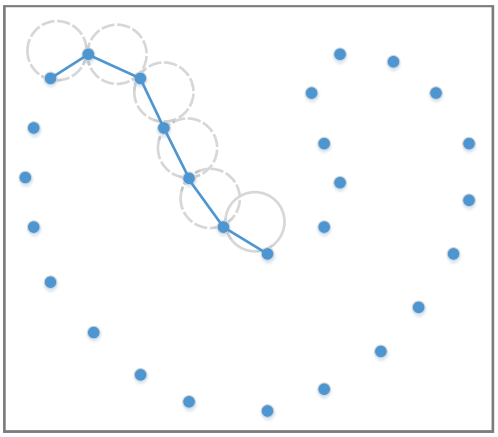


Figure 56: An illustration of ball-pivoting method [20] in 2D.

method [145] to build mesh connectivity by using pivoted spheres of the prescribed same size. A 2D example of ball-pivoting is illustrated in Figure 56. The principle is connecting two points if the prescribed circle touches both points without containing any other points. In 3D case, a sphere is pivoted along an existing edge and a triangle is created if the sphere touches three points without containing any other points. In the last step of [54], the reconstructed mesh by ball-pivoting is projected back to the original input points. Therefore, imperfections in the input data cannot be removed. Moreover, the reconstructed mesh uses all the input points without decimating as the ball-pivoting method computes a mesh interpolating the input points.

5.2.5 *Hybrid methods*

In the early stages, computational geometry methods are purely based on the theories of Voronoi diagram and Delaunay triangulation. As such these methods severely lack robustness to noise and outliers, in order to address this problem, a few preprocessing methods were specially proposed to refine input data, e.g., AMLS algorithm for point cloud smoothing [48]. In this way, sharp feature can be detected and noise/outliers are able to be filtered as well.

For the preprocessing, a primitive-based method is proposed in [99]. It is tailored for reconstructing large scene with many planar parts. Therefore, a set of planar primitives is required to precompute from the original point cloud as illustrated in Figure 57. This preliminary computation can be addressed by using the existing works such as RANSAC [144], region growing [100] and Gaussian sphere mapping [36]. The reconstruction procedure in [99] operates in two steps. First, a structuring process is performed to label every point in the original

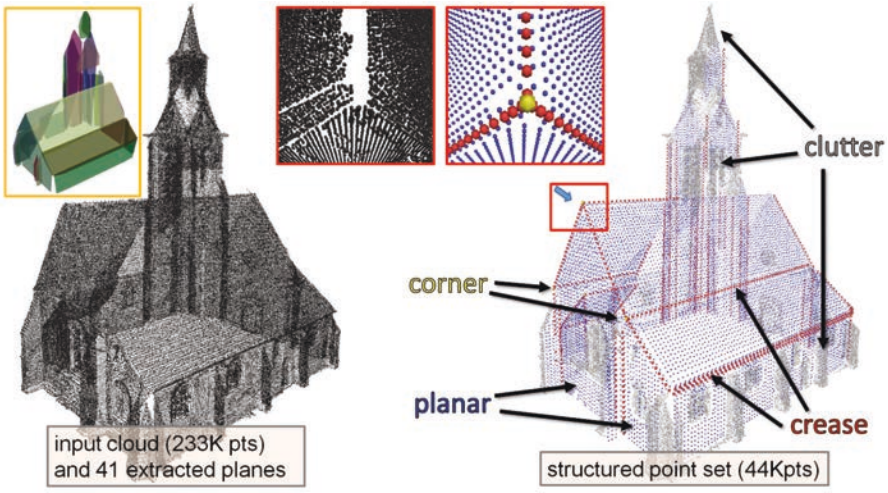


Figure 57: A figure from [99]. Left model shows plane extraction from the input point cloud. The right one illustrates the process of structuring. Plane, creases, and corners are depicted in blue, red and yellow respectively. The other parts such as windows and doors are regarded as clutters (grey). Also note that the left point cloud is re-sampled to produce right point cloud.

point cloud. The label can be one of the four structural types, i.e., *planar*, *crease*, *corner* and *clutter*, as shown in Figure 57. Meanwhile, the original point cloud is re-sampled as well and a simplified point cloud is obtained without losing details and features. Next, the Delaunay triangulation of the structured point cloud is computed and a reconstructed surface is determined by means of graph cuts.

Generally, Delaunay triangulation already contains an approximation of shape to be reconstructed. As illustrated in Figure 58, the boundary between inner triangles and external ones is a good approximation of the original curve. As previously stated in Section 5.2.1, the crust algorithm determines this boundary by checking if two end points of an edge belong to the original point cloud. Some other more sophisticated method are applied for defining the boundary such as graph cuts, which is also utilized in [99]. In computer vision, graph cuts are widely used such as image segmentation. The graph cuts problem can be formulated as an optimization of an specified cost function. Since the cost function are defined based on applications, the method is very flexible and can be suitable for many different cases. A min-cut problem is illustrated as Figure 59. The cut with minimum weight is computed to separate the graph into two disconnected components. It can be proved that max-flow and min-cut are two dual optimization problem [25]. Therefore, the solution of min-cut can be obtained through solving max-flow problem [31]. The technique of min-cut is also used in [99] to extract the reconstructed mesh as well. Once Delaunay triangulation is computed, a graph can be constructed using triangles/tetrahedrons as nodes. An edge con-

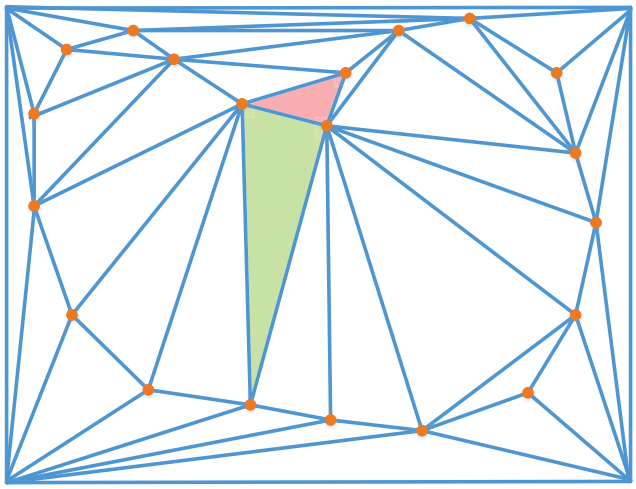


Figure 58: A point set (orange) is sampled along a 2D curve. A Delaunay triangulation of the point set together with the corners in a bounding box is illustrated. Red and green triangles are from the Delaunay triangulation. The green triangle locates in the inner part of the curve and the red one is outside. The edge shared by these two triangles approximates a local part of the curve.

necting two nodes can be built by checking if two triangles/tetrahedrons share a edge/face in the Delaunay configuration. As mentioned before, different applications may have distinct ways to define edge weight, and in [99] the weights are determined according to visibility consideration.

5.2.6 *Summary*

Besides the work aforementioned, a large amount of computational geometry methods have been proposed, e.g., power crust [14] and alpha shapes [56]. Noticeably, ball-pivoting [20] algorithm generates a manifold subset of an alpha shape without computing the 3D Delaunay triangulation. It is an advancing front method, i.e., triangles are incrementally constructed until the reconstructed mesh covers the original point cloud. The principle of ball-pivoting is simple: Three points form a triangle if a ball of a user-specified radius touches them without containing any other point.

A strength of computational geometry methods is the strict mathematical proofs and theoretical guarantees for result quality, e.g., resulting meshes have correct topology and they can converge to original surfaces under certain sampling conditions. However, in reality, 3D acquisitions cannot satisfy these rigorous conditions, moreover, acquired point clouds can inevitably contain various levels of noise and outliers. Therefore, generally these computational geometry methods fail when samples are not sufficiently dense. Furthermore, they can-

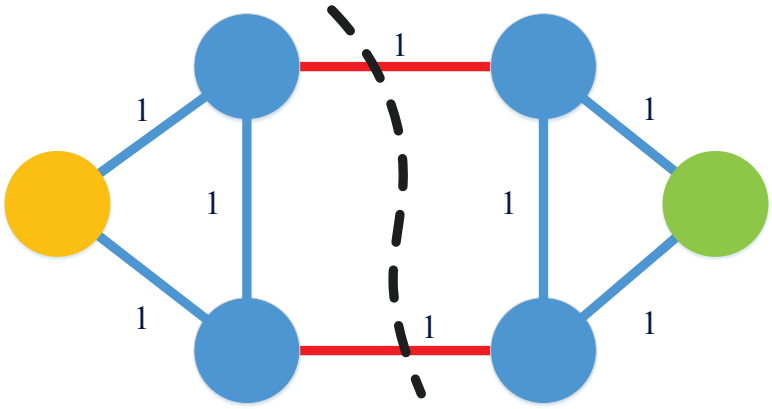


Figure 59: An illustration of min-cut: A graph with 6 nodes is divided to two groups by cuts. One group is connected with the yellow node and another group is connected with the green one. The weights of the cutting edges sum up to a weight for the cut. Min-cut requires the cut with minimum weight. In this example, each edge is associated with weight 1.

not automatically remove redundant samples and thus tend to produce very large data set.

5.3 IMPLICIT SURFACE METHODS

The implicit surface methods aim to find a function $f(x)$ in $\mathbb{R}^3$ (which possibly has no analytical form) such that the *implicit surface* defined by $f(x) = 0$ approximately interpolates the points in $P = \{p_1, p_2, ..., p_m\}$, where $p_i \in \mathbb{R}^3$. Once the function $f(x)$ is determined, a general meshing procedure, i.e., marching cubes [109], is performed to generate a mesh representation.

5.3.1 *Preliminaries*

5.3.1.1 *Implicit function*

In mathematics, *implicit function* [112] is not defined explicitly such as in analytic form, but rather is defined in terms of an algebraic relationship $f(x) = 0$. This algebraic equation is an *implicit* equation. The set of values that satisfy this equation forms a curve if $x \in \mathbb{R}^2$ and a surface if $x \in \mathbb{R}^3$. The curve or surface defined in this way is named as *implicit curve* or *implicit surface*. For example, a 2D circle centering in the origin with unit radius can be regarded as the implicit curve determined by the implicit equation $x^2 + y^2 = 1$.

5.3.1.2 *Marching cubes*

The marching cubes algorithm proposed in [109] is a divide-and-conquer approach to generate resulting mesh from the input implicit

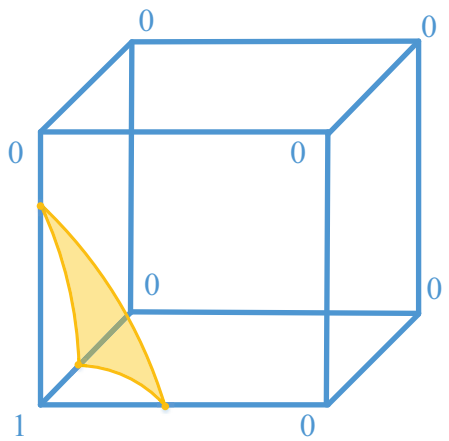


Figure 60: A cube intersects with a surface (orange) defined by a implicit equation $f(x) = 0$ at the dots (orange). Using the implicit function, the vertices of the cube can be labelled with 1 if $f(x) < 0$ or 0 if $f(x) > 0$.

equation. It adopts a case table strategy to define the topology of reconstructed triangle mesh. First, an octree is constructed to enclose the original surface defined by the implicit equation $f(x) = 0$. Afterwards, the algorithm determines how the surface intersect each cube in the octree.

As illustrated in Figure 60, the values of $f(x)$ in the cube's vertices can be used to determine if the cube intersects the implicit surface. If two ends in one edge have$f(x)$ values with different signs, the intersection in the edge can be approximated by linear or high degree interpolation. As previously stated in last paragraph, a case table is utilized to define triangle topology in each cube. Each cube has eight vertices and every vertex has two possible labels 1 or 0. Therefore, there are totally $2^8 = 256$ cases that a surface can intersect a cube. Moreover, due to symmetry property of cube, the 256 cases actually are able to reduced to 15 patterns as shown in Figure 61. Among these 15 patterns, the simplest one is no intersection found and thus no triangle is reconstructed as the first one illustrated in Figure 61.

In each cube, the mesh topology inside the cube is determined as shown in Figure 61 and the intersections can be easily computed using interpolation. The configuration of the resulting mesh is obtained by checking all cubes surrounding the original surface. Marching cubes algorithm is simple but efficient. Moreover, it is very suitable for parallel implementation due to the independence of different cubes. Therefore, it is regarded as a general mesh generation technique for implicit surface methods such RBF (Section 5.3.2) and Poisson reconstruction (Section 5.3.3). However, one drawback of marching cubes is the difficulty to control the quality of triangles in resulting mesh, i.e., skinny triangles cannot to be avoided.

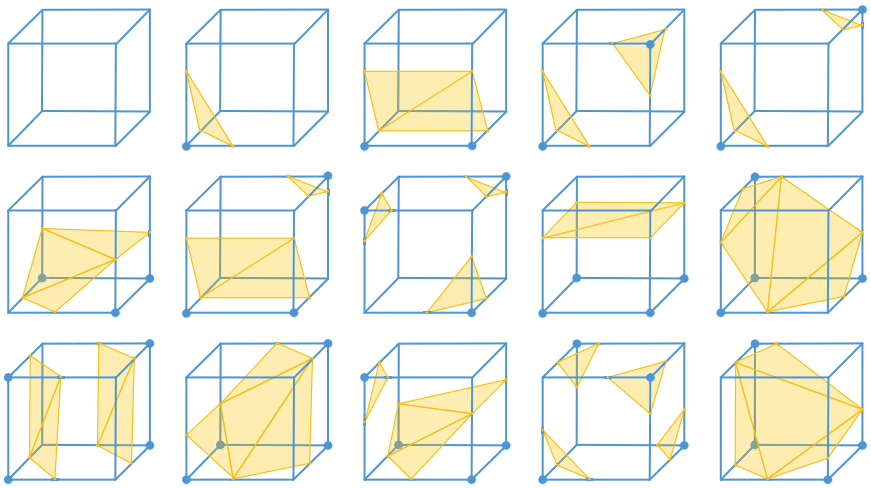
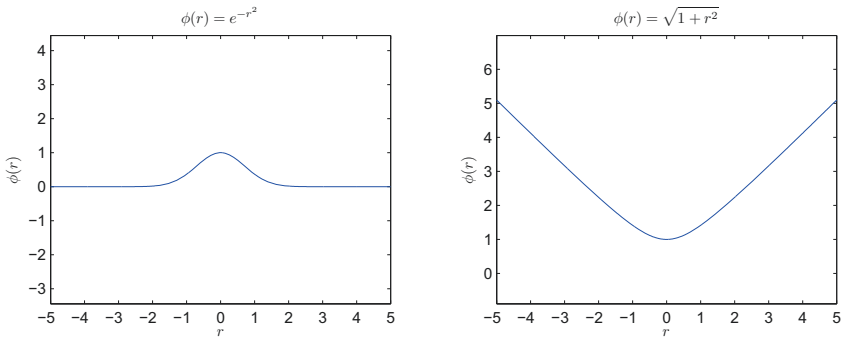


Figure 61: In marching cubes algorithm, in a cube 256 triangulating cases can be reduced to 15 patterns because of symmetry property of cube.



(a) (b)

Figure 62: An illustration of two types of radial basis functions: Gaussian (a) and multiquadric (b).

5.3.2 *Radial basis function (RBF) methods*

5.3.2.1 *Radial basis function (RBF)*

A radial basis function (RBF) is a real-value function $f(x)$ whose value only depends on the distance from origin, mathematically, i.e., there exits an function $\phi(r)$ so that $f(x) = \phi(|x|)$, where $x \in \mathbb{R}^n$ and $r \in \mathbb{R}$. Figure 62 depicts the shapes of two common types of RBF, i.e., Gaussian and multiquadric. RBF is a powerful tool which has already been applied in many areas. For example, radial basis function network is a successful case in artificial intelligence [21]. RBFs are typically used to build up function approximation in the form

$$f(x) = \sum_{i=1}^{N} \omega_i \phi(\|x - x_i\|), \tag{77}$$

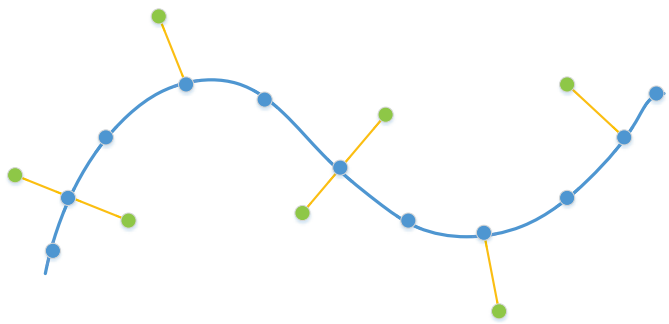


Figure 63: In RBF reconstruction, off-surface points (green) along surface normals are introduced to avoid the trivial solution. These points may be specified on either or both sides of the surface, or not at all. On-surface points are colored in blue.

where the function $f(x)$ is represented as the sum of N weighted RBF and ω_i is the weight coefficient. Given a set of samples $\{x_i\}$ and the corresponding function values $\{f(x_i)\}$, once a type of RBF function $\phi(r)$ is chosen, the coefficients $\{\omega_i\}$ can be computed using linear least squares. A detailed survey on the theory and applications of radial basis function was presented in the book [33].

5.3.2.2 *RBF in surface reconstruction*

In surface reconstruction, a point set $P = \{p_1, p_2, ..., p_m\}$ is given, and a reconstructed surface is computed to fit points in P. Therefore, surface reconstruction intrinsically is a interpolation problem. Due to the favorability of RBF for interpolation [33], Carr et al. [34] proposed a surface reconstruction method based on the theory.

In the approach demonstrated in [34], the original surface is assumed to be defined by an implicit equation $f(x) = 0$ and the function $f(x)$ can be represented by RBF. Concretely, the function $f(x)$ determining the implicit surface is supposed to be in the form

$$f(x) = p(x) + \sum_{i=1}^{m} \omega_i \phi(|x - p_i|), \tag{78}$$

where p is a linear polynomial, ω_i is real coefficient and ϕ is a radial basis function. Since the implicit surface is used to approximate the original surface, the points in the input $P = \{p_1, p_2, ..., p_m\}$ satisfy

$$f(p_i) = 0, \quad i = 1, ..., m, \tag{79}$$

where $\{p_i\}$ are points lying the original surface. In order to prevent trivial solution that $f(x) \equiv 0$ for all x, as shown in Figure 63, the off-

surface points are introduced and used to add non-zero values of $f(x)$ to the problem,

$$\begin{aligned} f(p_i) &= 0, \quad i = 1, ..., m && \text{(on-surface points)}, \\ f(p_i) &= d_i, \quad i = m+1, ..., m+N && \text{(off-surface points)}. \end{aligned} \tag{80}$$

As illustrated in Figure 63, the off-surface points can be computed by using orientated normals and d_i is the distance between p_i and the surface. Therefore, obviously $f(x)$ can be viewed as a signed-distance function. It defines the distance between x and the implicit surface as well as a sign determined by orientated normals.

As stated previously, the surface reconstruction is formulated as a scatted data interpolation problem defined in Equation 80. To address the problem, the solution is restricted in $BL^{(2)}(\mathbb{R}^3)$, i.e., the Beppo-Levi space of distribution on $\mathbb{R}^3$ with square integrable second derivatives [55]. Thus the side conditions are imposed on the coefficients $\{\omega_i\}$

$$\sum_{i=1}^{N} \omega_i q(x_i) = 0, \text{ for all polynomial } q \text{ with degree at most } m. \tag{81}$$

Equation 80 and Equation 81 can be written in matrix form as

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} \omega \\ c \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \; A_{i,j} = \phi(|p_i - p_j|), \; B_{i,j} = b_j(p_i), \tag{82}$$

where $\{b_i(x)\}$ are polynomial bases and $p(x) = c_i b_i(x)$. Solving Equation 82 becomes impossible when the size of the set P is larger than hundreds thousand. This is due to the memory requirement to store dense matrix A. A straightforward solution is using compactly support RBF to make A sparse. However, numerical experiments shows it causes many undesirable artifacts [34]. On the other hand, in [34] a RBF center reduction method is proposed and Fast Multipole Method [72] is also applied. It makes use of the idea that when computation is operated, far points can be represented as a cluster. These techniques increase the algorithm performance with a reasonable memory requirement and allow the method to handle large data.

5.3.2.3 *Discussion*

The RBF method generally generates a smooth reconstructed surface and is able to fill holes even with a non-uniform distributed input point cloud. In addition, the resulting surface has a function representation, which means that gradient and curvatures can be computed analytically. However, the method requires orientated normals as input which is not robustly estimated especially for surface with very detailed features. Moreover, it is not an computationally effi-

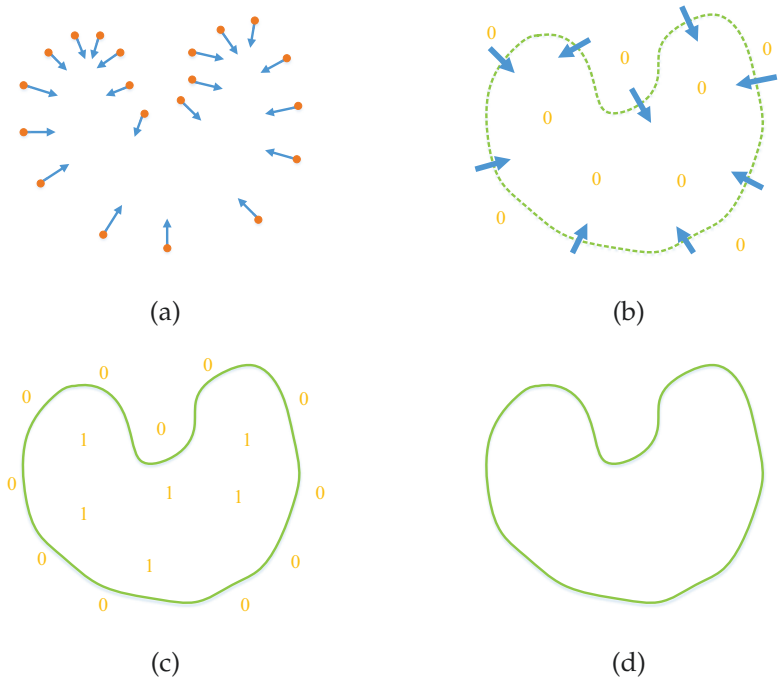


Figure 64: Illustration of Poisson surface reconstruction: A point cloud with orientated normals (a) is given as input for reconstruction. An indicator function as illustrated in (c) is computed to obtain the final surface (d). The indicator function is defined as 1 inside the model and 0 in outside region. Since the function is constant almost everywhere, the indicator gradient is shown as (b).

cient method in contrast to other methods such as Poisson reconstruction [91].

5.3.3 *Poisson surface reconstruction*

5.3.3.1 *Poisson reconstruction*

Poisson's equation is another powerful tool for interpolation problem. In computer graphics, it is widely utilized for image editing [132], mesh deformation [173], as well as surface reconstruction proposed in [91]. Like the RBF method, Poisson reconstruction is a global approach which considers all the data at once. It produces a smooth surface robustly fitting given point cloud possibly containing noise.

Poisson surface reconstruction aims to compute an *indicator function* defined as 1 inside the model and as 0 outside as shown in Figure 64c. Then the shape of the model is able to acquired by using the indicator function (see Figure 64d). The key idea in the method is the relationship between orientated point cloud and scalar filed defined by the indicator function. As illustrated in Figure 64b, the gradient of the indicator function is zero almost everywhere except on the surface. Since the indicator value has 1 inside and 0 outside, the di-

rection of the non-zero gradient is pointing towards the inner part of the surface. If the orientated normals direct to internal as well, then they exactly coincide with the gradients without considering vector length, as shown in Figure 64a and Figure 64b.

Let χ is the unknown indicator function and V are the oriented normal field of the input point set P, as previously stated, the gradient field of the indicator function can be regarded as the normal field, i.e.,

$$\nabla\chi = V. \tag{83}$$

If applying divergence operator on both side of Equation 83 and noticing that Laplacian is equal to the divergence of the gradient, the Equation 84 is arisen

$$\nabla \cdot \nabla\chi = \Delta\chi = \nabla \cdot V. \tag{84}$$

Therefore, Poisson surface reconstruction is formularized as the problem of solving the Poisson's equation in Equation 84. Due to the local support property of Laplacian, the problem can reduces to solving a well-conditioned sparse linear system, which is different from the RBF method. Furthermore, to address the problem in Equation 84, a multiresolution structure motivated by multigrid [74] is applied and a solver with a block Gauss-Seidel solver is operated as well.

Poisson reconstruction [91] always produces a smooth resulting surface which potentially suffers the oversmoothing problem. To address this issue, a modified version of Poisson surface reconstruction was proposed lately, i.e., screened Poisson reconstruction [90]. Based on Poisson surface reconstruction, a screening term is defined over a sparse set of points in [90]. Adding the screened term can significantly improve the geometric fidelity and thus preserves features in original surface.

5.3.3.2 *Discussion*

As the RBF method, Poisson surface reconstruction can robustly generate a smooth surface from a point cloud possibly within noise. Generally, the method is an efficient method for various point clouds acquired by different ways. However, it requires orientated normals as input which sometimes cannot be estimated correctly. In addition, the approach is not able to generate a mesh adaptively which means that the mesh size could be very large in order to well reconstruct fine detailed regions. Moreover, due to using marching cube algorithm to extract final mesh, the resulting meshes can contain skinny triangles discussed in Section 5.3.1 and thus it needs an extra remeshing step for applications such as physically-based animation [171].

5.3.4 *Moving least squares (MLS) methods*

5.3.4.1 *Moving least squares (MLS)*

Besides RBF and Poisson's equation, the method of *least squares* is a standard approach for solving fitting problem, such as linear regression in statistics [93]. As a generalized version of least squares, the *moving least squares* (MLS) approach for surface reconstruction was presented in [102]. In contrast to the global property of the least squares method, the moving least squares method is performed by applying weighted least squares around the local region centering at the point where the reconstructed information is required. Thus, generally the moving least squares consists many least squares procedures. By this way, the global function $f(x)$ used to represent the surface can be formulated as

$$f(x) = f_x(x). \tag{85}$$

In order to calculate f_x, approximating plane and MLS projection are operated sequentially.

Let a plane with normal n_x and passing point a_x, the plane can be computed by solving the optimization problem

$$\underset{a_x, n_x}{\text{minimize}} \quad \sum_{i \in I} \theta(|p_i - a_x|)(n_x, p_i - a_x)^2, \tag{86}$$

where $\theta(x)$ is a weight function and I is the index set of the points locating in the local region centered in x. Once the plane is determined, a local coordinate frame can be acquired naturally. Within this local frame, hypothetically the original surface close to x can be approximated by a polynomial surface. Therefore, f_x can be computed by solving

$$\underset{f_x}{\text{minimize}} \quad \sum_{i \in I} \theta(|p_i - a_x|)(f_x(p_i) - f_i)^2, \tag{87}$$

where f_x is a polynomial and f_i is the height of p_i to the local plane obtained previously.

5.3.4.2 *Feature preserving MLS*

By means of the MLS method presented in the previous section, a C^∞ smooth surface can be reconstructed from the point set $P = \{p_1, p_2, ..., p_m\}$. Therefore, the MLS method can robustly generate a surface from a potentially noisy point cloud as the RBF method and Poisson reconstruction. However, when reconstructing models with sharp features, C^∞ is not a complimentary property. To address this problem, the variants of MLS [130, 60] were proposed to preserve

sharp features. The insight of solutions is using a "better" norm than L_2 in Equation 87.

The approach proposed in [60] makes use of robust statistics to create neighborhoods for the fitting problem in the MLS method. Similar to the consolidation method [18] using L_1 norm, the median of residuals is utilized in the minimization

$$\underset{f_x}{\text{minimize}} \quad \underset{i \in I}{\text{median}} |f_x(p_i) - f_i|, \tag{88}$$

which is different from Equation 87. This technique used is motivated by the previous work [117] applied in computer vision application. On the other hand, robust local kernel regression is applied in [130]. The approach modifies the Equation 87 to the form

$$\underset{f_x}{\text{minimize}} \quad \sum_{i \in I} \theta(|p_i - a_x|)\rho(f_x(p_i) - f_i), \tag{89}$$

where the Welsch's function is chosen for ρ, i.e., $\rho(x) = \frac{\sigma^2}{2}(1 - e^{-(\frac{x}{\sigma})^2})$.

5.3.4.3 *Discussion*

The MLS method can generate a smooth surface from a potentially noisy point cloud like the RBF method and Poisson reconstruction. Since the method is not a global method but independently constructs local surface, it can be integrated with advancing fronts methods such as [142]. Noticeably, the MLS method is also widely applied for point cloud related applications, e.g., rendering [7, 75]. A comprehensive discussion about the subject can be found in [94]. However, the computational cost can increase significantly when the local model assumption is very complicated [130]. As the approach requires marching cube, resulting meshes possibly contain skinny triangles.

5.3.5 *Partition of unity methods*

5.3.5.1 *Multi-level partition of unity (MPU) reconstruction*

The partition of unity is widely used finite element methods such as [19] to integrate local approximations to an global one. The basic insight of partition of unity is to subdivide to data space to different parts, approximate data in each subdomain independently, and then combine the local approximations to a global solution using a smooth and local weights.

Concretely, let Ω is the data domain and the weights function $\{\varphi_i\}$ satisfy

$$\sum_i \varphi \equiv 1 \quad \text{on} \quad \Omega. \tag{90}$$

An approximation of the function $f(x)$ on Ω can be given as

$$f(x) \approx \sum_i \varphi_i(x) Q_i(x), \tag{91}$$

where $Q_i(x)$ is a local approximation in subdomaim Ω_i.

Ohtake et al. [127] proposed multi-level partition of unity (MPU) method to reconstruct an implicit surface over a point cloud. To create the implicit representation, a box first is constructed to bound the original point cloud. The box is then subdivided into an octree cells. In each cell, the points in it are approximated by a local shape function. Like the RBF method in Section 5.3.2, the function is computed as a signed-distance function i.e., the values of the functions are zeros close to the original surface and become positive/negative away from the surface. The signs are determined by normals and thus orientated normals are required as input for MPU reconstruction which is same to Poisson reconstruction discussed in Section 5.3.3. If the created shape function cannot correctly approximate the points data in a cell, i.e., without satisfied accuracy, the cell is subdivided and the procedure is repeatedly applied until the error of the approximation is lower than a prescribed threshold. Once all the local approximations are obtained, the global implicit surface is generated by blending local shape functions as Equation 91.

5.3.5.2 *Discussion*

Multi-level partition of unity reconstruction can efficiently compute a smooth reconstructed implicit surface from a possibly noisy point cloud. Like other implicit methods aforementioned, it does not need overhead computation for topology. The conception of MPU is simple and also applied in other areas, e.g., point cloud smoothing [122]. However, the MPU reconstruction does not provide a meshing procedure as well. To generate a mesh, an extra meshing algorithm is necessarily performed such as marching cube [109].

5.3.6 *Summary*

In previous sections, RBF, Poisson, MLS and MPU methods are studied in detail. Actually, due to the ubiquity of implicit surface reconstruction, the discussion cannot cover all the works in this category. More discussions can be found in the surveys [5, 140] on surface reconstruction. Generally, implicit surface method can approximate a surface by interpolating a potentially noisy point cloud. However, since marching cubes is used to accomplish the last meshing step, skinny triangles cannot be avoided. The quality of resulting meshes is not able to satisfy several applications, such as finite element meth-

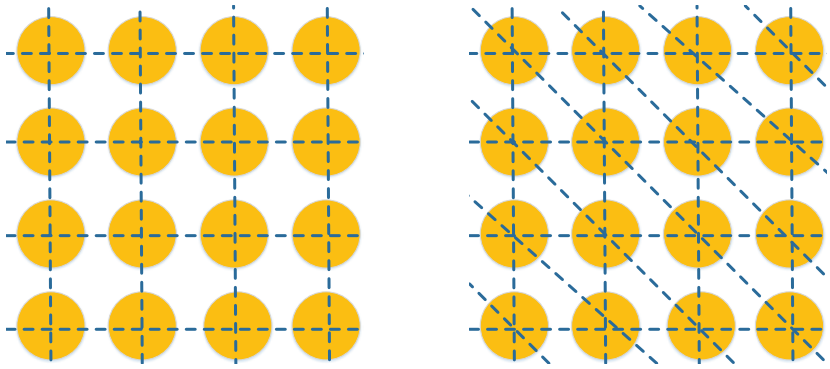


Figure 65: Self-organizing maps with rectangular (a) and triangular (b) configurations.

ods [150]. Therefore, extra post-processing is required, e.g., remeshing [171].

5.4 MACHINE LEARNING METHODS

Compared to computational geometry methods and implicit surface methods aforementioned, machine learning methods is a small category. However, it presents an alternative direction to address surface reconstruction problem. These arising methods results from the increasing popularity of machine learning techniques which have been commonly utilized in computer vision.

In computer graphics, a *self-organizing* map (SOM) was introduced for surface reconstruction as well. A *self-organizing map* is a type of neural network that is trained by unsupervised learning [96]. It has the similar structure to a two dimensional grid as shown in Figure 65. Each node in the grid is named as *cell* denoted by C_i and a weight ω_i is associated with cell C_i. Let $P = \{p_1, p_2, ..., p_m\}$ is the input point set in $\mathbb{R}^3$, the procedure of the training can be illustrated as iteratively deforming the self-organizing map to fitting the configuration of input set P. This training algorithm for self-organizing map is suitable for surface reconstruction: C_i can be considered as the vertex in the resulting mesh and ω_i is the 3D coordinate of the vertex. Concretely, for point p_i the winner cell with smallest distance to p_i is found and is denoted as $C(t)$. The weights $\{\omega_k\}$ are updated as

$$\omega_k(t+1) = \begin{cases} \omega_k(t) + K(t,d)(p_i - \omega_k(t)), & d = \mathrm{dist}(C_k, C(t)) \leqslant \delta(t), \\ \omega_k(t), & \mathit{otherwise}, \end{cases} \tag{92}$$

where $K(t,d)$ is a function $0 \leqslant K(t,d) \leqslant 1$ and $\delta(t)$ is a distance threshold. Detailed discussions are presented in [96, 172] and [172]

also proposed a surface reconstruction method based on self-organizing map.

Using similar ideas, [85] and [86] demonstrated modified versions of self-organizing map for surface reconstruction, i.e., growing cell structure and neural meshes respectively. Generally, the methods of this category pre-build a template mesh and a resulting mesh is generated by deforming the template to fit scattered points using certain criteria originating from machine learning. This strategy also exits in other literature such as [113, 125]. Both of them are based on an optimization framework to deform a template mesh to approximate another point cloud and mesh respectively.

Most recently in the research [169], the surface reconstruction problem was also addressed by means of *dictionary learning* [4] which has been widely applied for sparse representation of signals in machine learning. In [169], surface reconstruction is formulated as a procedure of dictionary learning where the vertices of the resulting meshes are treated as the dictionary and the connectivity is regarded as the sparse coding. Mathematically, the arising problem is to solve a minimization optimization which produces an optimal triangluar mesh approximating the input point cloud in geometry. In order to avoid excessive post processing operations such as remeshing and mesh smoothing, the regularization terms are introduced into the optimization as well to guarantee mesh quality. However, the eventual problem poses a huge challenge to solve it numerically as it is a complex non-convex optimization, which makes the solving procedure quite complicated even without a convergence guarantee.

One strength of these methods discussed previously is able to integrate meshing, smoothing and hole filling into the same procedure. Therefore, generally resulting meshes do not need post-processing. However, when the original surface has complicated topology such as high genus, the methods are not able to recover the topology correctly. Furthermore, unfortunately these methods generally need very expensive computation and thus cannot handle a big data set.

5.5 SUMMARY

Surface reconstruction is an important topic in computer graphics. It is usually the preliminary step in many applications such as 3D visualization and physically-based animation. This chapter covers the three mainstream categories of methods for surface reconstruction, i.e., computational geometry methods, implicit surface methods and machine learning methods. Different methods are compared and discussed as well including both their favorable advantages and uncomplimentary disadvantages. Computational geometry methods usually have theoretical guarantees, but cannot handle noise directly or produce economical meshes with fewer points than the input. The sam-

pling conditions required in some computational geometry methods are too difficult to satisfied. Implicit surface methods have the ability to deal with noise, however, skinny triangles are often introduced. Machine learning methods generally do not need mesh processing afterwards, whereas their computations are relatively expensive.

Some methods are not easy to categorize such as [99], because they simultaneously utilize several various techniques together in one method, which is one trend of recent works. Another distinction is that new methods prefer to be specialized for a particular subject. For instance, the algorithm proposed in [99] only deals with large scanning models with many canonical components such as planar parts. The work [168] aims to reconstruct not only geometric shapes but schematic diagrams as well.

6 CONTRIBUTION: SPHERE PACKING AIDED SURFACE RECONSTRUCTION

The contribution in this chapter was published in

- Kun Liu, Patricio A. Galindo and Rhaleb Zayer, Sphere Packing Aided Surface Reconstruction for Multi-View Data, ISVC 2014, 10th International Symposium on Visual Computing, LNCS Volume 8888, pp 173-184, 2014.

Surface reconstruction, which is a fundamental problem in geometry processing, has long been targeted at scanned data. The development of 3D acquisition technologies as stated in Chapter 2 proposed new challenges which motivates the evolution of surface reconstruction techniques.

With the rise of multi-view acquisition, existing surface reconstruction methods often turn out to be ill adapted to the highly irregular sampling and multilayered aspect of such data. In this chapter, a novel surface reconstruction technique is presented to address these new challenges by means of an advancing front guided by a sphere packing methodology. The method is fairly simple and can efficiently triangulate point clouds into high quality meshes. The substantiated experimental results demonstrate the robustness and the generality of the proposed method.

6.1 INTRODUCTION

Reconstructing a surface from a point cloud is a well studied problem in geometry processing [109, 91, 45]. Historically, these methods have been tailored for range scan data. The rapid development of technologies for point cloud acquisition raises new challenges as the point data obtained by different approaches exhibit distinct properties in terms of density, accuracy, distribution, and so forth. Moreover, additional post-processing operations such as mesh smoothing [44, 59], remeshing [171, 17] and mesh simplification [83, 79] are often necessary to refine the reconstructed surfaces and make them exploitable in practice. Therefore, it is still challenging to design an all-purpose surface reconstruction algorithm.

In this chapter, a novel method for surface reconstruction is developed with special attention to point clouds generated by multi-view stereo reconstruction. An ideal surface reconstruction algorithm is characterized by several properties such as robustness to noise, low computational cost, and high quality of resulting meshes. However,

these attributes are often hard to combine within a single approach. In fact, many existing methods generally suffer from some limitations. As discussed in the literature survey on surface reconstruction [140], for instance, classical Delaunay-based methods [56, 12, 13, 45] cannot directly handle noisy data, implicit surface methods [109, 34, 127, 102, 91] inevitably create skinny triangles, and the efficiency of learning based methods [172, 96, 85, 86] remains a significant problem.

Awareness of the particular aspects of multi-view data has prompted recent interest in filling the gap left by classical geometry reconstruction methods. For instance, [139] relies on a combination of restricted Voronoi diagrams and depth maps, whereas methods such as [163] recast the reconstruction problem as the recovery of a visibility consistent surface from an initial Delaunay triangulation. This problem is addressed as finding a minimum *s*-*t* cut over an adaptive domain. Although these approaches can achieve highly accurate results the algorithms are rather intricate and time consuming.

The method proposed in this chapter operates locally by means of an *advancing front* strategy [142] guided by a global criterion based on *sphere packing*. Sphere packing is well known to approximate Voronoi diagram commonly used for producing high quality meshes [149]. In this way, robustness to noise in the input data and efficiency are achieved while keeping algorithmic approach simple. Noticeably, the method uses only points from the original point cloud as the vertices of the resulting mesh (subject to small perturbation possibly). This feature can benefit several applications such as tracking in computer vision and photogrammetry as it reduces the additional re-projection errors across frames. The proposed approach is evaluated both on multi-view stereo and other point data. The results also demonstrate the efficiency and the robustness of the proposed approach.

The rest of this chapter is organized as follows. Previous work on sphere packing and advancing front are reviewed in Section 6.2. The reconstruction algorithm is presented in detail in Section 6.3, and experimental results are discussed in Section 6.4.

6.2 RELATED WORK

Existing surface reconstruction methods have been comprehensively studied in Chapter 5. To keep the exposition succinct, this section will be restricted to the literature related to the methods underlying the reconstruction approach proposed in this chapter, namely sphere packing and advancing front. For an extensive review of classical surface reconstruction techniques, the interested reader is referred to Chapter 5 or [5, 140].

6.2.1 *Sphere packing*

Sphere packing is a classical problem in geometry. It seeks an arrangement of non-overlapping spheres in a given metric space [151], where the sizes of the spheres can be either identical or different. Sphere

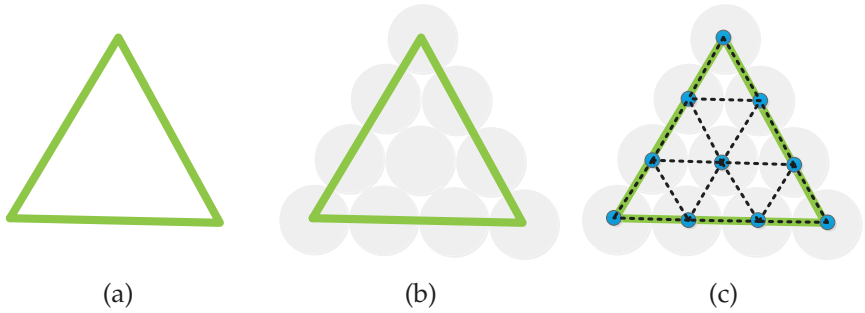


Figure 66: The intuition of using sphere packing in surface reconstruction: (a) shows a triangular region to be triangulated. Gray spheres are tightly packed in the region as shown in (b). A triangulation (black dot lines) in (c) is obtained using the tightly packed spheres.

packing has been used to generate meshes for finite element analysis under the bubble mesh analogy [149]. The basic idea of the bubble mesh algorithm in [149] consists of packing spheres tightly within a predefined domain. As illustrated in Figure 66, spheres are closely arranged in a triangular region. New vertices are placed in the centers of spheres and edges are built by connecting adjacent spheres. By this way, a triangular mesh is obtained. Specifically the problem shown in Figure 66 is *circle packing* [154] which is a special case of sphere packing in two dimensional Euclidean space.

In general, the pattern of the tightly packed spheres mimics a *Voronoi diagram* and thus the generated triangulation can be regarded as an approximation of a *Delaunay triangulation* as illustrated in Figure 67. As a Delaunay triangulation maximizes the minimum angle of the triangles in the triangulation [129], this property of sphere packing results in well-shaped triangular meshes whose elements are approximately equilateral, which can benefit many applications especially finite element analysis [51].

Unlike many other reconstruction methods, the bubble mesh method proposed in [149] doesn't triangulate a point cloud but a continuous domain, mostly representing a CAD model, in terms of algebraic equations. The method is performed by first placing spheres in the domain using a node spacing function. Gaps and overlaps are inevitable and hence afterwards a physically-based global optimization is operated to obtain an optimal tight packing. Once the domain is packed with spheres, mesh vertices are created as the sphere centers and mesh edges are created using constrained Delaunay triangulation [129]. For the global optimization, inspired by van der

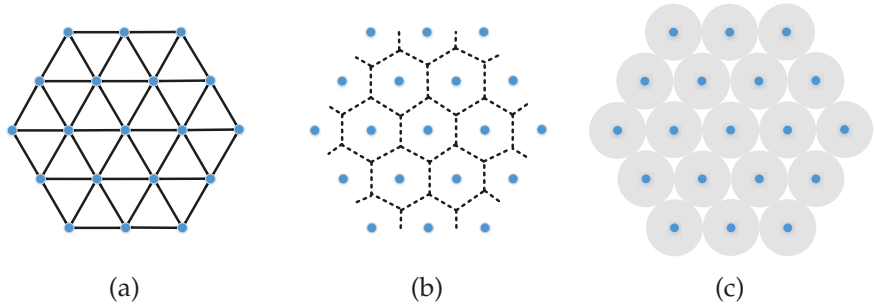


Figure 67: (a) shows a Delaunay triangulation of a point set (blue). The dash lines in (b) represent a Voronoi diagram which is dual to the Delaunay triangulation. (c) displays closely packed spheres (gray) whose configuration mimics the Voronoi diagram.

Waals force the attractive and the repulsive forces are defined between spheres. To obtain a tight packing, parameters need to be tuned carefully in the optimization.

Sphere packing importance has been also recognized in the context architectural freeform designs where it is used to generate the so called *circle packing meshes* [143]. A circle packing mesh is a triangular mesh approximating an arbitrary freeform shape. The incircles of triangles in the mesh form a circle packing and the associated orthogonal spheres centering in vertices form a sphere packing. Since the resulting mesh is a torsion-free support structure, it exhibits remarkable geometrical and aesthetic properties which are favorable in architecture. To generate such mesh, an initial mesh is computed and then morphed to a circle packing mesh which also maintains a close approximation of the input freeform shape. The problem is formulated as a nonlinear least square problem which is solved using the damped Gauss-Newton algorithm. The approach has satisfactory performance on shapes of simple topology, but easily fails if the input shape consists of complex topological components. Therefore, it is hard to apply for triangulating surfaces with complicated details.

Both the bubble mesh and the circle packing mesh methods can generate a triangulation for an input domain which possibly represents a CAD model or an architectural surface. The input domain is usually defined as an algebraic form. On the other hand, in this chapter a point cloud, which is a type of discrete geometry representation, is triangulated. Hence, intermediate interpolation is necessary during surface reconstruction in order to predict the absent geometry, which makes the problem more challenging. Furthermore, the proposed approach in this chapter does not apply any tedious global optimization as in [149, 143], but still can produce meshes with satisfactory quality demonstrated by the experimental results in Section 6.4.

6.2.2 *Advancing front*

The advancing front strategy is first used in 2D/3D-mesh generation for the finite element method [145, 108, 71]. Currently it is widely used in surface reconstruction. The reconstruction is based on the observation that each mesh element can be fully determined by local geometry.

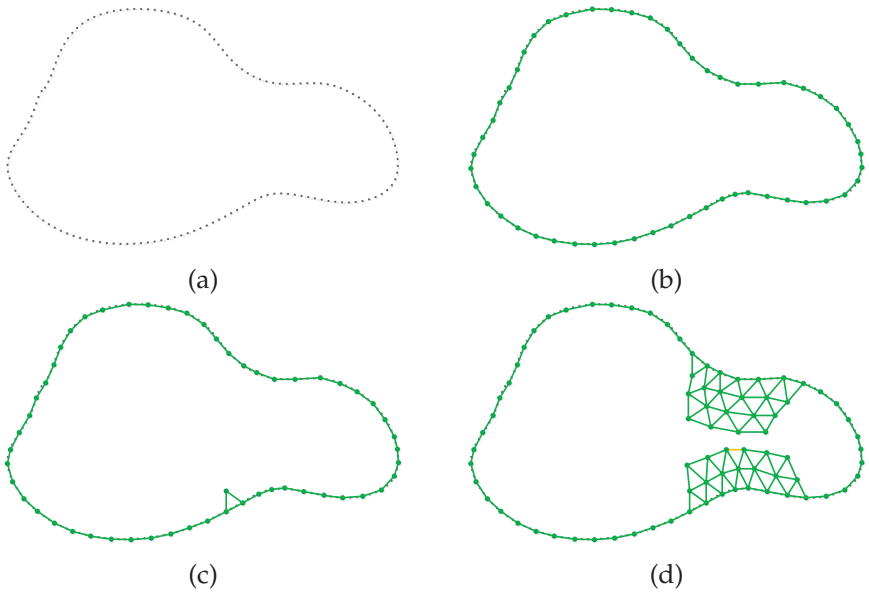


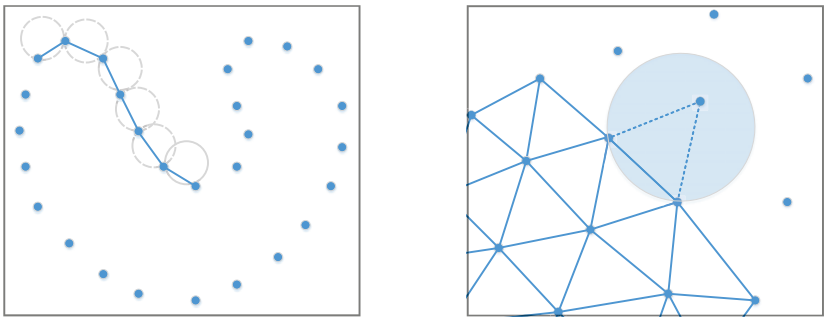
Figure 68: A region in (a) is to be triangulated using advancing fronts. Generally, initial front edges are generated along the boundary of the region as shown in (b). A principle for advancing is adopted to determine triangles being added as (c). The problem of merging front edges also needs to be solved, which is illustrated in (d). The yellow edge in (d) represents the next front edge which is to be propagated.

In previous work especially mesh generation for the finite element method, the boundary of the region to be triangulated is initialized as the initial fronts as shown in Figure 68b. The key of advancing front is the principle of advancing, i.e., how to define step size and generate new triangles. In general, different principles are applied in different methods. In an advancing front method, two or more fronts sometimes meet in the same location as illustrated in Figure 68d, and thus a way to merge multiple fronts usually is required as well.

The Ball-Pivoting method [20] adopts the advancing front strategy to triangulate a point cloud into a mesh. The concept of Ball-Pivoting is simple - three points form a triangle if the pivoting sphere touches them without enclosing other points as illustrated in Figure 69. Once a triangle is created, the process of pivoting is repeated around another edge. In this way, a triangular mesh is reconstructed progressively. The resulting meshes by the method is a subset of alpha shape [56] of the input points. Therefore, geometric and topolog-

ical correctness are guaranteed theoretically under certain sampling conditions. However, these conditions are too rigorous in practice.

The basic idea of the surface reconstruction methods using advancing front is propagating, i.e., growing the reconstructed mesh from a seed triangle until certain termination conditions are satisfied. In [142], a guidance field based on curvatures is introduced to direct the propagation and an adaptive mesh is generated with the bounded reconstruction error. In addition, due to moving least squares (MLS) used for the interpolation in the method, noise in point data can also be handled.



(a) (b)

Figure 69: An illustration of the Ball-Pivoting method: A simple two dimensional case of the Ball-Pivoting procedure is displayed in (a). A fixed size circle is used to create connections between points. A three dimensional case is shown in (b). A sphere (blue) is pivoted around an edge until touching a point. The triple of points enclosed by the sphere is used to create a new triangle.

Advancing front can benefit surface reconstruction in several aspects. Since distant triangles are created independently, parallel computing can be easily employed as demonstrated in the parallel Ball-Pivoting method [53]. Moreover, as advancing front algorithms only process points locally, it is not necessary to load the whole point cloud into the memory simultaneously. This facilitates triangulating huge point clouds by means of streaming techniques. In view of the advantages, the advancing front strategy is also applied for remeshing [146] in computer graphics. In this chapter, advancing front is adopted and sphere packing is used to direct the propagation during the reconstruction. The proposed method is efficient and elegant without tedious global optimization. To use the strategy of advancing front, three issues need to be carefully addressed, i.e., initialization, propagation and termination condition, which is comprehensively discussed in Section 6.3.2.

6.3 SPHERE PACKING AIDED SURFACE RECONSTRUCTION

The main steps of the proposed approach are outlined as shown in Figure 70. In this section, these steps are presented in detail, as well

as how sphere packing can be used to drive the advancing front in order to triangulate a point cloud into a mesh.

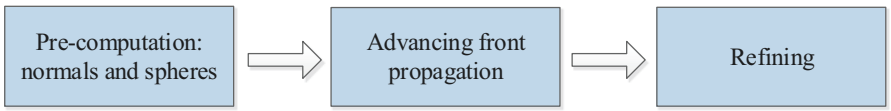


Figure 70: The outline of the proposed method.

6.3.1 *Sphere candidates generation*

In order to deal with the irregular sampling commonly encountered in multi-view data, first a sphere is sought to associate with every point in the data set in a similar way to the work of [128]. For this purpose, a normal direction n for each point p of the cloud is estimated using *principal component analysis* [82], i.e., n is defined as the unit eigenvector of the local covariance matrix and the eigenvector corresponds the smallest eigenvalue. In the proposed method, oriented normals are not required and normal directions are sufficient. Therefore, unlike [82], the *minimum spanning tree* algorithm is not necessary to make orientation of normal directions consistent.

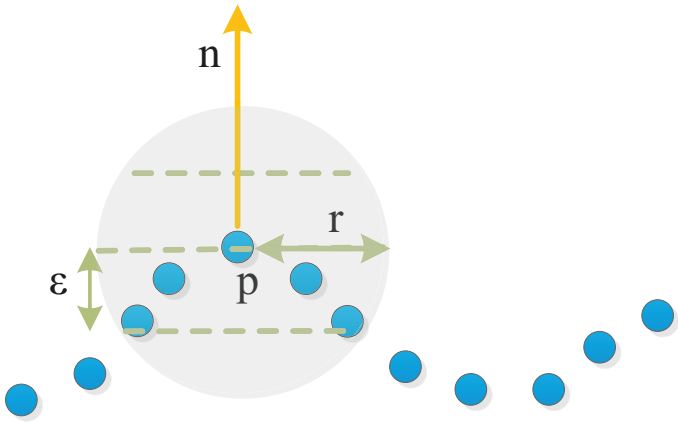


Figure 71: An illustration of generating sphere candidates.

As shown in Figure 71, with the computed normal direction n on the point p, a local plane is determined and the sphere candidate centered at p is defined as the largest sphere such that points in the sphere are all less than a prescribed ϵ distant from the plane. Therefore, the parameter ϵ can be used to bound the reconstruction error. This local linear hypothesis is used in the method [79] for mesh simplification as well. Since during the reconstruction redundant points are removed, the ϵ is also considered as a threshold on the size of resulting meshes [128]. In addition, another parameter r_{max} is also prescribed to bound the maximum size of spheres to avoid the infinite radii on the almost flat region.

6.3.1.1 *Discussion*

The work [128] uses a similar spherical covering to accomplish mesh reconstruction, however, spheres generated in our method are different than the ones from [128] in several respects as follows.

- In our method each point from the input is associated with a sphere centered in the original point. On the other hand, it is not the case in [128]. The randomly generated spheres from [128] do not use the original points as centers.
- Mesh reconstruction in [128] relies entirely on sphere intersections, thus relatively larger spheres are created in order to generate intersections. Using sphere intersections to build a mesh is not as elegant as using sphere packing in our method and a tedious mesh post-processing is required [128].
- The maximum sphere size is bounded in our method by a prescribed parameter to prevent generating extremely large spheres in low-curvature regions.

6.3.2 *Advancing front with packing criterion*

In order to build a triangulation based on the candidate spheres obtained in Section 6.3.1, an advancing front strategy is developed and the propagation is steered by sphere packing.

Typically advancing front methods are characterized by three main aspects, namely, **i) front initialization**, **ii) propagation rules** and **iii) termination conditions**. Among these, the key aspect is propagation rules where the principle which drives the front expansion is defined, e.g., pivoting ball as in [20] or curvature field as in [142].

In the proposed method, a criterion based on sphere packing is developed to guide the propagation - growing from a seed triangle to a triangular mesh covering the whole surface. The algorithmic outline of the proposed method is described in Algorithm 2.

6.3.2.1 *Front initialization*

The proposed advancing front algorithm starts from a seed triangle, and the edges of the seed triangle form the initial fronts. To generate a seed triangle, a sphere (v_1, r_1) is chosen randomly, where the sphere centered in v_1 with radius r_1 as shown in Figure 72. The second sphere (v_2, r_2) is selected using the following criterion

$$\min_{(v_2,r_2)} \mathtt{PairError}((v_1, r_1), (v_2, r_2)), \tag{93}$$

Algorithm 2 Advancing front in our method

Input: point cloud p, normals n, spheres s
Output: triangle list *tri_list*

```
fronts ← FRONTS_INITIALIZATION()          ▷ See Section 6.3.2.1
tri_list ← ∅
while fronts.size() > 0 do
    e ← fronts.pop()
    t ← EAR_CUTTING(e)                    ▷ See Section 6.3.2.2
    if t ≠ ∅ then
        tri_list.push_back(t)
        fronts.update()
        continue
    end if
    t ← POINT_ADDITION(e)                 ▷ See Section 6.3.2.2
    if t ≠ ∅ then
        tri_list.push_back(t)
        fronts.update()
        continue
    end if
    t ← MERGING_FRONTS(e)                 ▷ See Section 6.3.2.2
    if t ≠ ∅ then
        tri_list.push_back(t)
        fronts.update()
        continue
    end if
end while
```

where $\mathtt{PairError}((v_1, r_1), (v_2, r_2)) = |r_1 + r_2 - |v_1 - v_2||$. The third sphere (v_3, r_3) is chosen as

$$\min_{(v_3, r_3)} \max\{\mathtt{PairError}((v_1, r_1), (v_3, r_3)), \mathtt{PairError}((v_2, r_2), (v_3, r_3))\}. \tag{94}$$

After adjusting the radii to make the three spheres tangent to each other (see Figure 72b), the seed triangle with the vertices v_1, v_2 and v_3 is determined, and thus the fronts are initialized as the edges (v_1, v_2), (v_2, v_3) and (v_3, v_1) as illustrated Figure 72c.

6.3.2.2 *Propagation rules*

Once the fronts are initialized, a series of operations are sequentially applied to the current front until a new triangle is successfully created. These operations include *ear cutting*, *point addition*, and *merging fronts* as described in Algorithm 2. The specific rules related to each operation will be briefly described below.

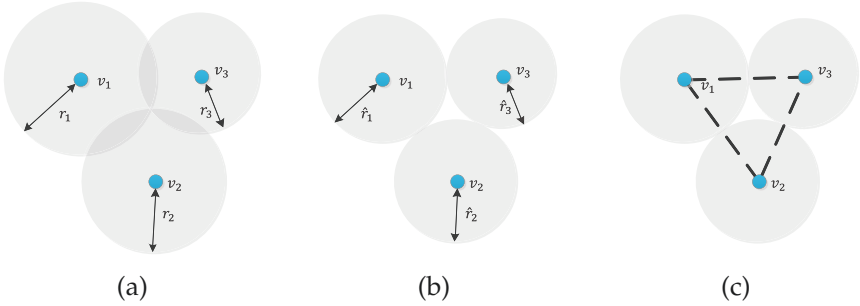


Figure 72: The figure illustrates the generation of a seed triangle. First, a tightly packed triple sphere is chosen (a). Then, the sphere radii are adjusted to make a tangent packing configuration (b). Finally, the centers of the spheres are connected to create the seed triangle (c).

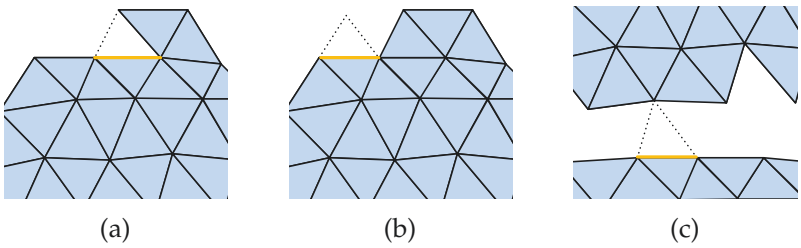


Figure 73: Three operations applied for advancing the current front (yellow): (a) ear cutting; (b) point addition; (c) merging fronts.

EAR CUTTING: This operation is applied to check if the current front edge can create a new triangle with an adjacent edge as illustrated in Figure 73a. Before creating the triangle, a set of four *filters* discussed in Section 6.3.2.4 need to be passed. In this way, unsatisfactory triangle candidates are eliminated, e.g., non-manifold triangles, self-intersected triangles and skinny triangles.

POINT ADDITION: If a new triangle cannot be created after applying the ear cutting operation, the point addition operation is consequentially explored. This operation creates a few triangle candidates by using the current front and its nearby points as shown in Figure 73b. Again in this case, the four filters in Section 6.3.2.4 are used before new triangles are validated. If none of the candidates can pass the filters, the next operation, i.e., merging fronts, is performed. Otherwise, the created triangle is chosen from the candidates using Equation 94. In this way, a tight packing configuration is guaranteed. Once an optimal triangle is found, the spheres centered at the three vertices are shrunk to prevent sphere intersections. In the implementation, a local quadratic surface is constructed [142] before applying the point addition operation. The use of this local surface allows the elimination of noise in the data. In a sense, the local fitting also yields a desirable advancing direction for font propagation. Moreover, the

points in the input point cloud can also be perturbed by projecting on the local surfaces to get a smoother reconstruction result.

MERGING FRONTS: This operation is used to merge different fronts as illustrated in Figure 73c. In principle, it is similar to the point addition operation except two distinctions. First, it creates triangles only using points along fronts. Second, no new points (spheres) are introduced and thus shrinking the spheres is not necessary.

6.3.2.3 *Termination conditions*

In the implementation, a queue data structure is utilized to manage front information. Once a triangle is created, the current front is removed from the queue and the edges of the new triangle are marked as fronts (line 8, 14, 20 in Algorithm 2). Otherwise, only the current front edge is deleted (line 4 in Algorithm 2). The propagation is terminated when the queue is empty. Furthermore, the queue is a First-In-First-Out (FIFO) data structure. It leads to an approximate radial propagation which prevents ubiquitous indentations along fronts as demonstrated in Figure 74.

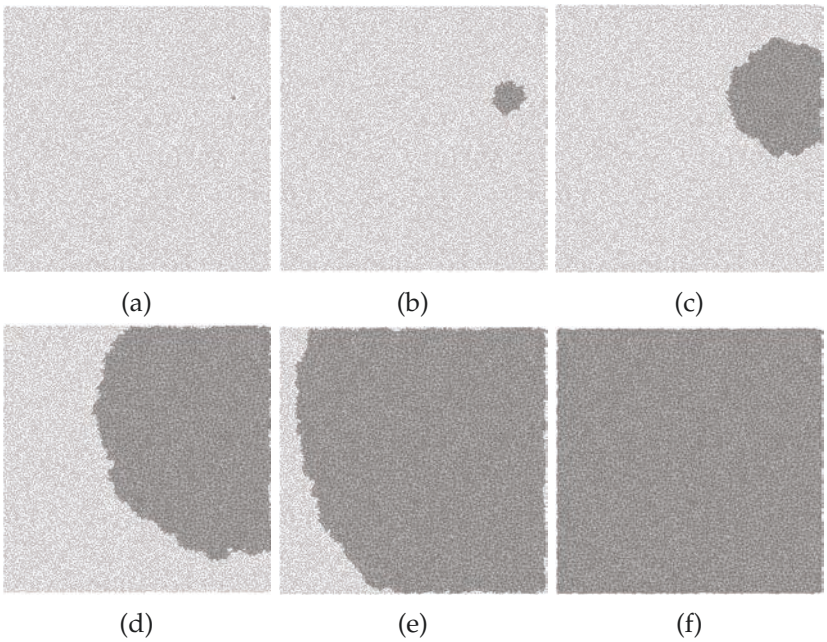


Figure 74: A point cloud (grey dots) representing a square is triangulated using the method proposed in this chapter. The figure illustrates the font propagation. (a) displays the initial seed triangle and (f) shows the final resulting mesh. (b)-(e) are the intermediate reconstructed results. Noticeably the propagation is approximately along radial direction.

6.3.2.4 *Filters in front propagation*

Four filters are utilized throughout propagation operations to control the validity of new triangle candidates.

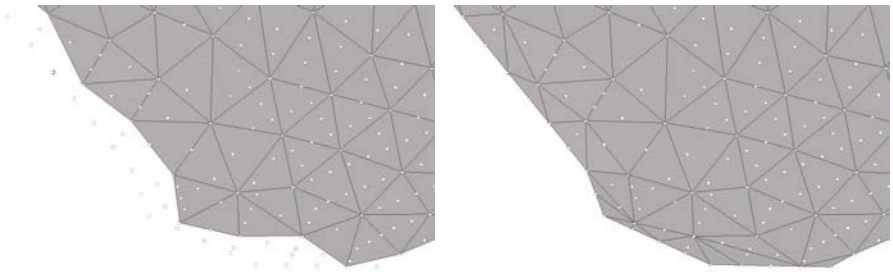
- Non-manifold edge filter: checks if new triangles may introduce non-manifold edges. A half-edge data structure [26] is used to facilitate the implementation.
- Triangle orientation filter: prevents orientation flipping. In the experiments, the angle of two adjacent triangles is required to be lower than 90 degrees.
- The triangle overlapping filter: helps avoid spatial overlapping of nearby triangles. To detect the overlapping of two triangles, one triangle is projected to the plane determined by the other one and vice versa. The triangles are spatially overlapped if the projected triangles have intersections in the plane.
- The triangle quality filter: guarantees the quality of triangles by using constraints on triangle angles. Different constraints are used in the different operations. In the ear cutting operation, the angle between the current front and the adjacent edge is required to be less than 90 degrees. In the point addition operation, the minimum angle in the new triangle is bounded by a threshold. In our experiments, 30 degrees is prescribed. In the merging fonts operation, the two angles associated with the current front are required be acute.

6.3.3 *Refinement*

Once a new triangle has been created during advancing front, an edge swapping operation can be applied if the edge length decreases by the swapping [16]. This operation helps improve the quality of the resulting mesh.

Boundary treatment: Once the mesh is constructed by the advancing front method presented in Section 6.3.2, a supplementary propagation can be employed to refine boundary regions illustrated in Figure 75a. In principle, this refinement operates similarly to the propagation in Section 6.3.2.2 except that it uses weaker triangle quality filters, i.e., with relaxed parameters. By applying the supplementary propagation, the results can be significantly improved as shown in Figure 75b.

Hole filling: Due to the nature of multi-view data, often there are small regions where data can be missing (for instance due to the lack of matches across views). For such cases, the hole filling algorithm [106] is used in post processing step. Although the algorithm has a $O(n^3)$ time complexity, most of holes are small and thus it op-



(a) (b)

Figure 75: The comparison of (a) without and (b) with the supplementary propagation. The white dots represent the input point cloud.

erates reasonably. In the experiments, the algorithm is automatically applied to small holes with less than 50 edges.

6.4 RESULTS

The proposed method in Section 6.3 is evaluated on point cloud data acquired by multi-view stereo reconstruction as well as other point data to show the generality of the method. A series of experiments are performed to demonstrate the efficiency. Point normals are not necessary as the input, because point coordinates are sufficient for the proposed method as discussed in Section 6.3. Data with different geometry properties are also used to test the robustness of the method.

6.4.1 *Discussion on mesh quality*

Mesh quality is crucial in mass applications such as geometric modeling, rendering and numerical simulation. It has been extensively studied in the literature [51, 10, 131, 28, 148] and various quality criteria have been proposed. In general, different criteria are applied in different applications [28]. Two popular used criteria are listed as follows:

- Shape of the mesh elements. It is directly motivated by the computation stability of numerical simulation, as well as the requirement of efficient rendering. For triangular meshes, a well-shaped triangle normally has no small angles or large angles. For example, meshes with such well-shaped triangles are good for matrix conditioning [148].
- Geometry accuracy. The generated mesh is a satisfied approximation to the original geometry [10]. The approximation often refers to locations and/or higher-order differential properties.. It is usually realized by tuning an error metric to create a mesh with suitable resolution. Generally a more accurate mesh means

the higher resolution as well as mesh complexity. Therefore, it is desirable to find an optimal trade-off between geometry accuracy and mesh complexity.

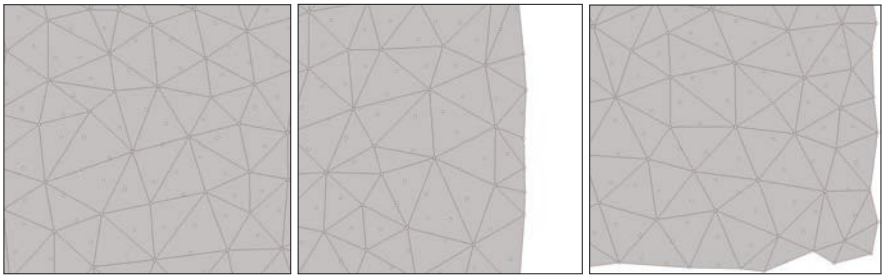
In this work, the two quality criteria are also employed to analyze and evaluate the experimental results, To check shape of the mesh elements, the distribution of triangle angles is examined. For geometry accuracy, two measures are used with regard to distance error and completeness respectively. Concretely, as illustrated in Figure 83 and Figure 84, the two measures are fulfilled via computing depth errors and coverage areas occupied by reprojected 3D point cloud on 2D images.

6.4.2 *Results of synthetic data*

The first experiment is performed on a synthetic point data. The point cloud is generated by uniformly sampling 13224 points on a unit square region as shown in Figure 74. First a seed triangle is constructed as Figure 74a. The fronts are advanced by using the proposed method in Section 6.3 and triangles are progressively created (see Figure 74b-Figure 74e). Noticeably, the mesh growing is approximately along radial direction. This is a good property that prevents ubiquitous indentations along fronts, which can avoid heavy processing such as merging edges and filling holes. After the front propagation, the whole square region is triangulated as Figure 74f. Three more close-up views are displayed in Figure 76 to show the inner part, the boundary part and the corner part of the resulting mesh respectively. As shown in the close-up views, the reconstructed mesh generally consists of high quality triangles. Most of triangles are well-shaped and only a few less equilateral triangles are along the boundary and the corners. This is due to a trade-off between triangle quality and geometry correctness. In order to correctly reconstruct the boundary and the corner geometry, some less well-shaped triangles are inevitably introduced.

6.4.3 *Results of multi-view data*

For the multi-view setting tests are performed on the benchmark proposed in [155]. The point clouds are obtained based on the quasi-dense matching approach of [88] using multi-view data from [155]. Only point coordinates are given as input. The proposed method can handle the orientation problem as stated in Section 6.3. The results are shown for the Herz-Jesu in Figure 77, 78 and the fountain-P11 data sets in Figure 79, 80. The input point clouds and the reconstructed meshes are shown in Figure 77 and 79. Two close-up views of the resulting meshes of the two data sets are displayed in Figure 78 and 80 respectively. From the results, the meshes faithfully



(a) (b) (c)

Figure 76: Three close-up views of the triangulated result (see Figure 74f) are displayed. (a) shows an inner part of the resulting mesh. (b) and (c) illustrate the results of the boundary and the corners respectively. The small white dots represent the input point cloud.

preserves the topology and the geometry of the input point cloud. The close-up views also demonstrate that proposed method can produce high-quality meshes, e.g., the triangles in the resulting meshes are well-shaped.

The results of the proposed method are compared to the Poisson reconstruction method [91] as it is one of the most popular surface reconstruction methods especially for multi-view stereo data. A zoom on the reconstructed Herz-Jesu, Figure 81, reveals a considerable number of skinny triangles resulting from the Poisson reconstruction, Figure 81a, whereas the triangles generated by our method are generally are well shaped, Figure 81b. This is confirmed by the histogram representation of triangle angle values in Figure 81c and Figure 81d. The same experiment is also conducted on the data set fountain-P11 as shown in Figure 82.

The corresponding Table 3 lists detailed statistics about angles. The resulting Poisson reconstruction exhibits many angles lower than 30 degrees. However, most of angles in our resulting mesh are between 30 degrees and 90 degrees as well as within a smaller standard deviation, which indicates the generated triangles are close to equilateral triangles.

During the meshing process, most of existing methods methods such as Poisson reconstruction [91] introduce additional points into resulting meshes. The proposed approach is different - only points from input point cloud are used as vertices of the resulting mesh. This property prevents additional errors caused by interpolation, which can benefit many applications such as tracking in computer vision and photogrammetry. This is demonstrated by the evaluation benchmark proposed in [155] where ground truth data (from a laser scan) can be used to evaluate the reconstruction error. The proposed approach is compared to different mesh reconstruction techniques using point data from [70], namely, the Ball-Pivoting [20], the method in [128], Poisson reconstruction [91]. As a top-rank method in the

	Herz-Jesu		Fountain	
	PR	Ours	PR	Ours
0°-30°	13.05%	2.91%	10.01%	2.97%
30°-60°	40.06%	51.66%	45.88%	51.60%
60°-90°	33.54%	39.16%	33.43%	38.90%
90°-120°	12.80%	5.76%	10.34%	6.06%
120°-150°	0.54%	0.43%	0.32%	0.40%
150°-180°	0.01%	0.08%	0.01%	0.07%
standard deviation	26.50°	18.76°	24.94°	18.94°

Table 3: The statistics about the angles in the results Figure 81 and Figure 82. PR is short for Poisson Reconstruction.

Middlebury benchmark [147], the method presented in [66] is also examined in the comparison.

The evaluation results are summarized in Figure 83 and Figure 84. The red pixels represent locations where no results are obtained or locations where errors larger than 30σ. The gray scale encodes the variance weighted depth difference. Brighter color corresponds to lower error w.r.t ground truth. In general, the proposed method performs relatively well and yields fewer red pixels, which suggests the results of the proposed method are closer to the ground truth. Detailed evaluation of the corresponding relative errors and completeness is summarized in Table 4.

Method	Herz-Jesu		Fountain	
	compleatness	rel. err.	compleatness	rel. err.
BP	75.83%	2.87	73.51%	2.22
YO	76.79%	2.99	79.96%	2.89
PR	78.83%	3.14	82.79%	3.33
YF	80.04%	2.98	79.60%	2.04
Ours	86.38%	2.91	91.74%	0.88

Table 4: The completeness and the rel. err. (relative error) of the results in Figure 83 and Figure 84.

6.4.4 *Results of other data*

As a sanity check, the reconstruction on point cloud data with added Gaussian synthetic noise is further performed. The reconstruction results are shown in Figure 85 and Figure 86 as well as the histograms about triangle angles, which demonstrates the generality of our method. The point clouds with different geometry properties

are used in the experiments, and the algorithm robustness is demonstrated as well. The sizes of all point clouds and resulting meshes in our experiments are also listed in Table 5.

Model	# Points	# Vertices	# Triangles
Herz-Jesu (Figure 77b)	630057	54126	106057
Fountain (Figure 79b)	359840	39470	77957
Buddha (Figure 85a)	543652	92689	185236
Dragon (Figure 85b)	655980	88031	176058
Armadillo (Figure 85c)	196269	26860	49716
Statuette (Figure 85d)	180754	24837	48117
Deckel (Figure 86a)	64266	9011	18018
Caesar (Figure 86b)	90797	9747	18412

Table 5: The sizes of the input point clouds and the resulting meshes in Section 6.4.

6.5 SUMMARY

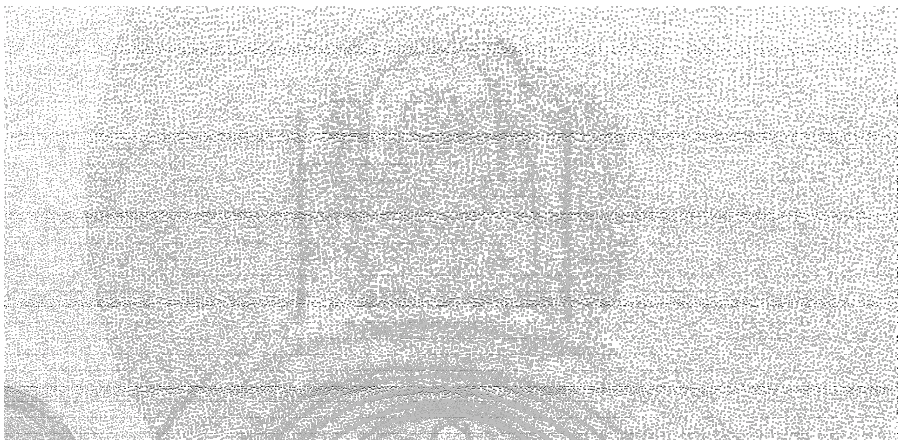
In this chapter, a novel surface reconstruction method is proposed based on intertwining sphere packing with an advancing front strategy. The local nature of the advancing front and the flexibility of sphere packing allow addressing the highly irregular input commonly encountered in multi-view data. The resulting meshes are well-shaped and approximate well the original geometry. Since the proposed method creates triangles only using points from the input point cloud, the results generally have lower reconstruction errors demonstrated in Section 6.4.

Theoretical guarantees in terms of geometric and topological correctness in the light of the work [56] will be explored in future. Another future work is introducing parallel computing into the proposed algorithm as the parallel implementation of the Ball-Pivoting method [53]. Data streaming also can be studied in order to possibly process huge data which doesn't fit memory.

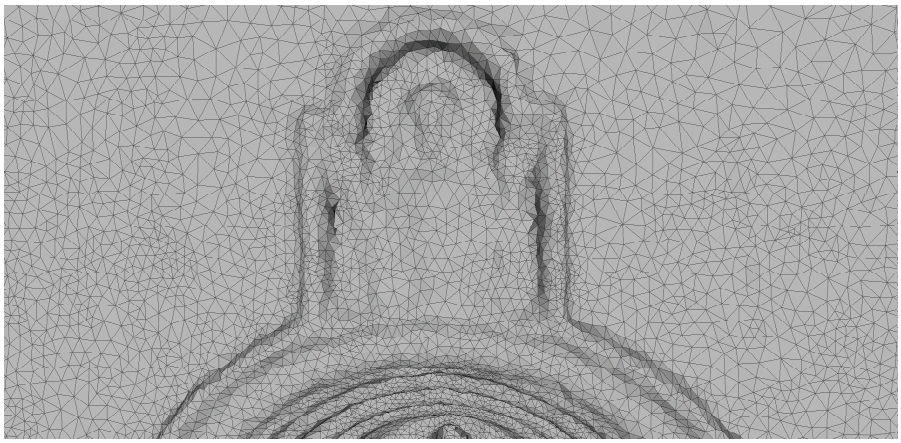
(a)

(b)

Figure 77: The Herz-Jesu data: (a) displays a dense point cloud computed using multi-view stereo reconstruction [88]. (b) shows the resulting mesh using the proposed reconstruction method.



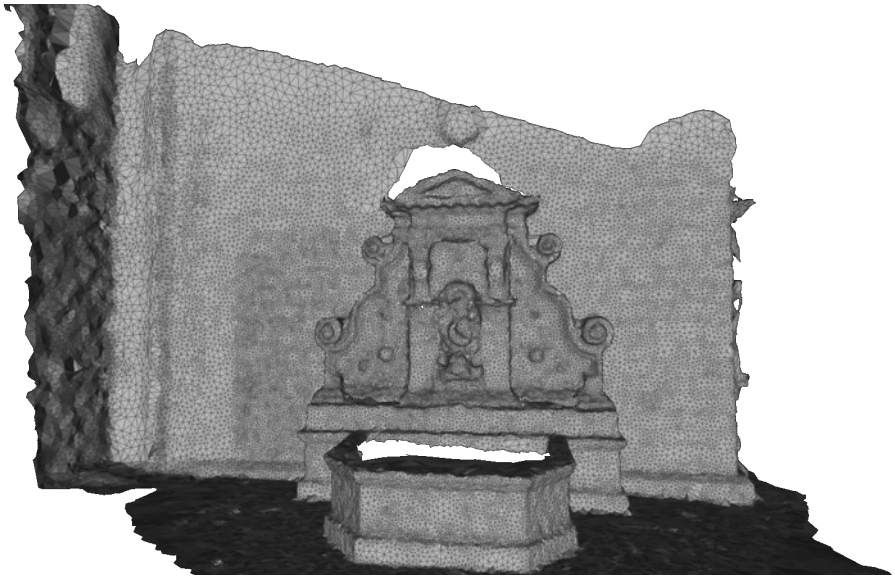
(a)



(b)

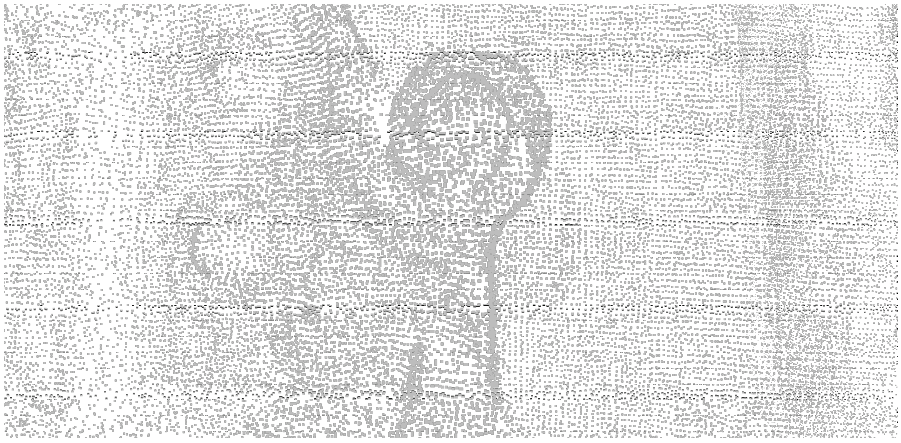
Figure 78: An illustration of a close-up view of the point cloud and the resulting mesh in the Figure 77.

(a)

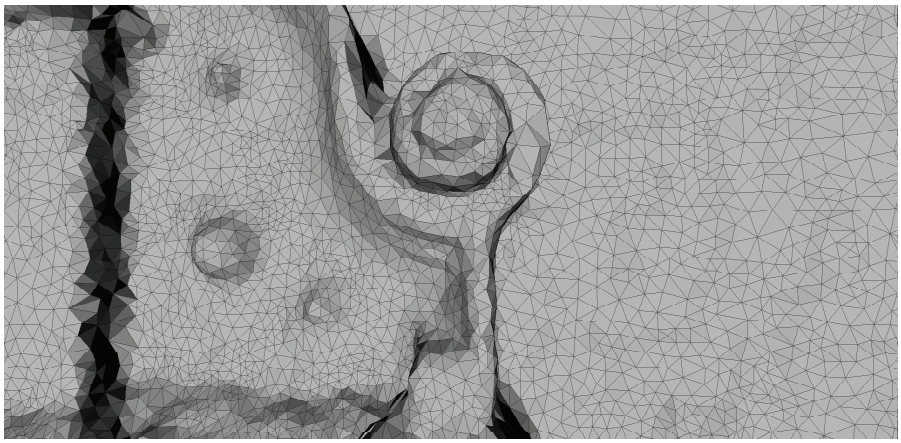


(b)

Figure 79: The Fountain data: (a) displays a dense point cloud computed using multi-view stereo reconstruction [88]. (b) shows the resulting mesh using the proposed reconstruction method.



(a)



(b)

Figure 80: An illustration of a close-up view of the point cloud and the resulting mesh in Figure 79.

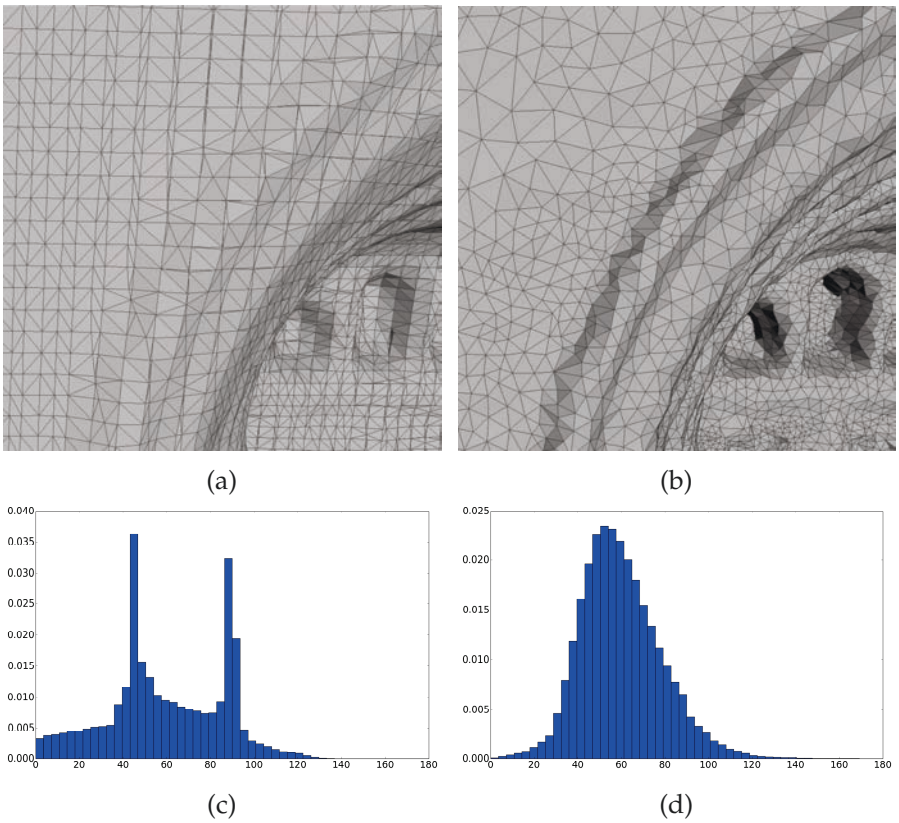


Figure 81: A comparison of Poisson reconstruction (a) and the proposed method (b) is illustrated. The input point cloud is same to the one in Figure 77a and the two meshes are displayed in the same close-up view. (c) and (d) are two corresponding histograms about triangle angle values. The horizontal axis represents angle and the vertical axis represents the percentage of the appearance of corresponding values.

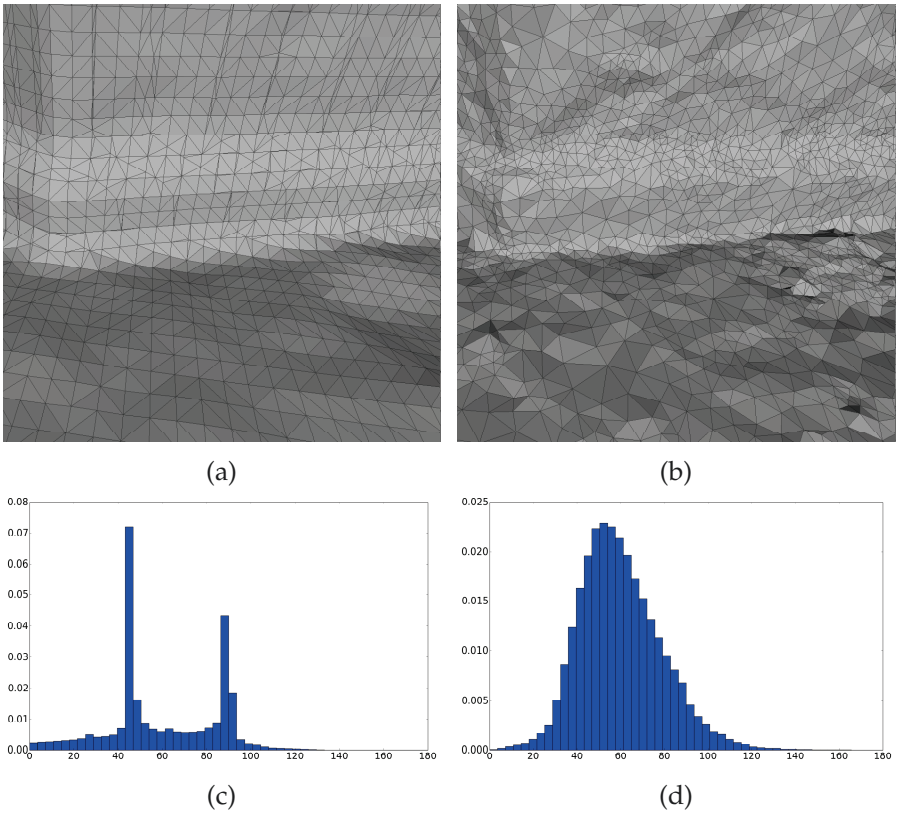
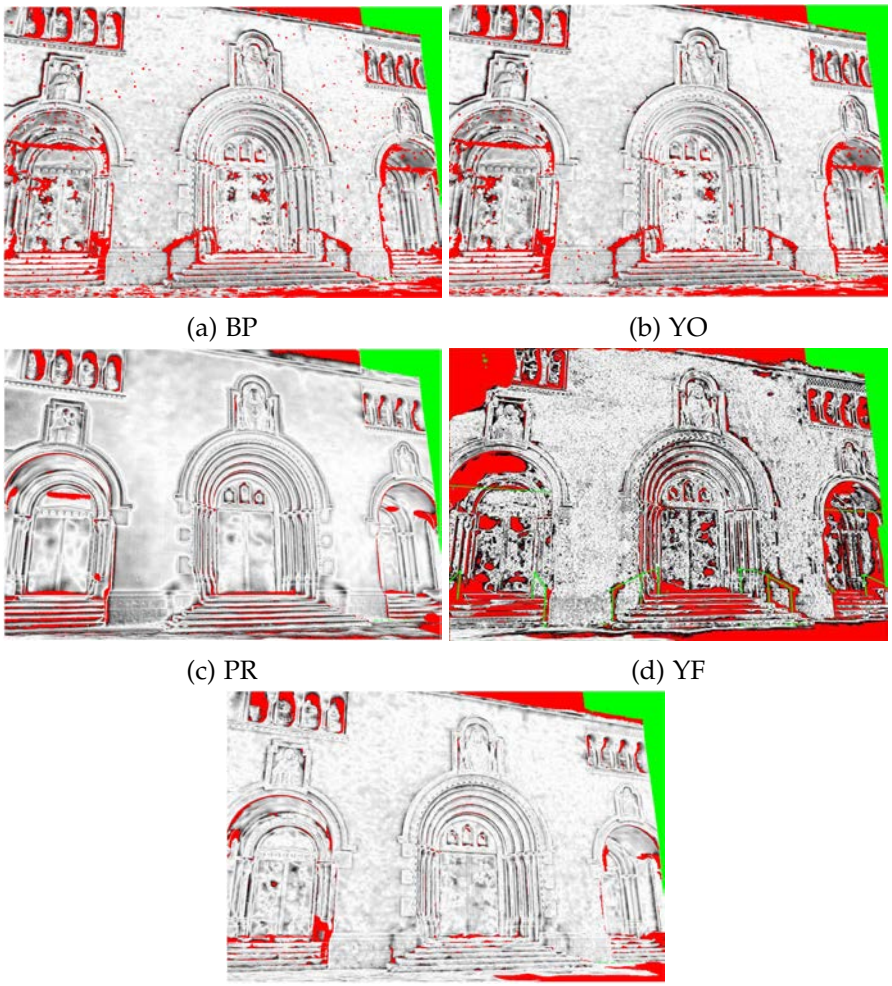


Figure 82: A comparison of Poisson reconstruction (a) and the proposed method (b) is illustrated. The input point cloud is same to the one in Figure 79a and the two meshes are displayed in the same close-up view. (c) and (d) are two corresponding histograms about triangle angle values. The horizontal axis represents angle and the vertical axis represents the percentage of the appearance of corresponding values.

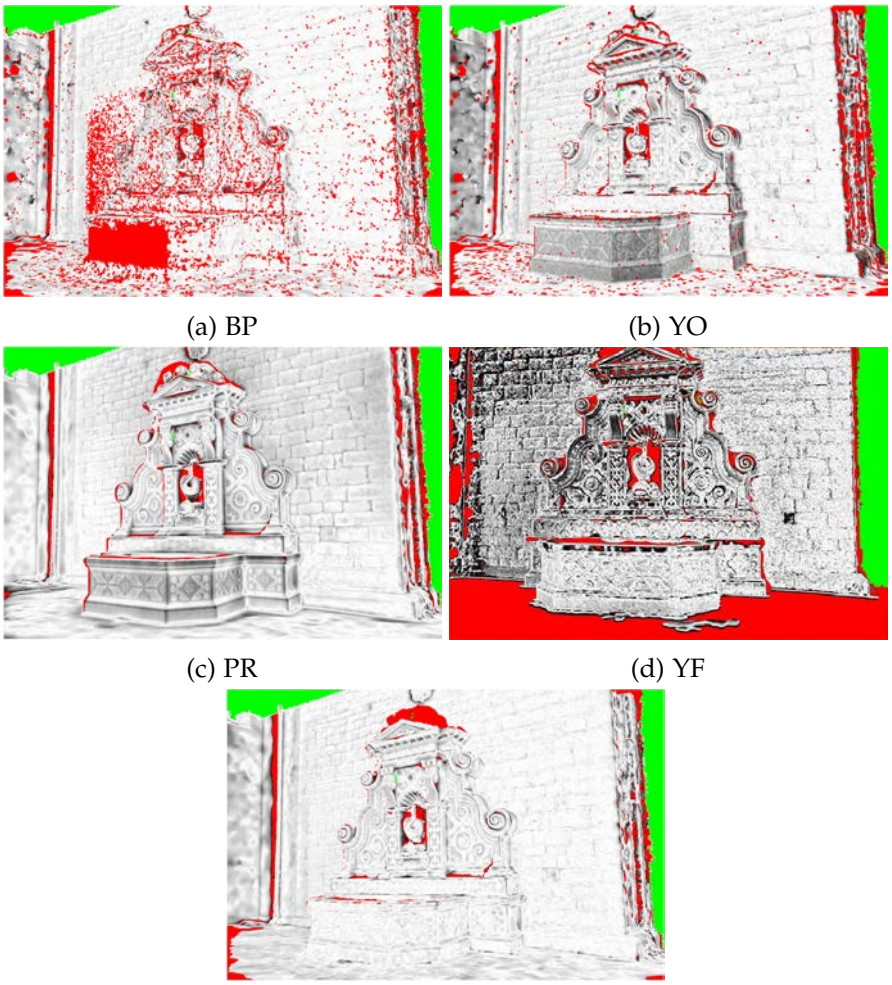


(a) BP (b) YO

(c) PR (d) YF

(e) Ours

Figure 83: The mesh reconstruction results (Herz-Jesu) of BP, YO, PR and YF in [20, 128, 91, 66] respectively, as well as the proposed method in this chapter are evaluated using the method in [155]. YF generates points from images and computes a mesh using Poisson reconstruction [91], while other methods use point data from [70]. The images show the variance weighted depth difference. Red pixels represent locations where no result are obtained or locations where errors larger than 30σ. Green pixels represent the missing scan data of the ground truth. The relative errors between 0 and 30σ are displayed using gray scale from 255 to 0.

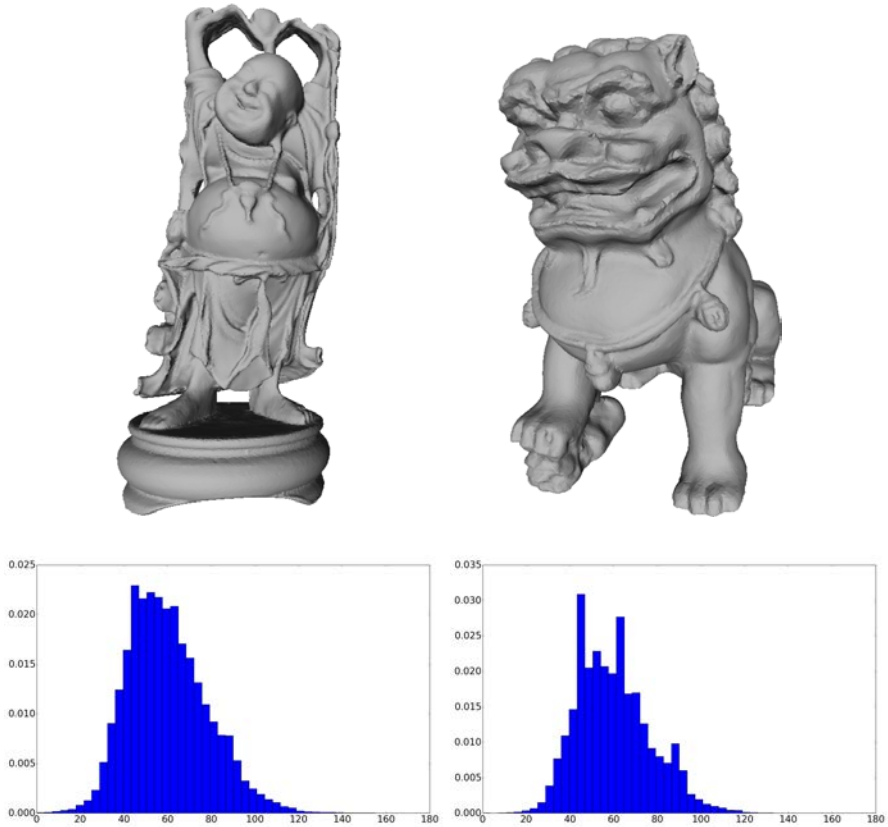


(a) BP (b) YO

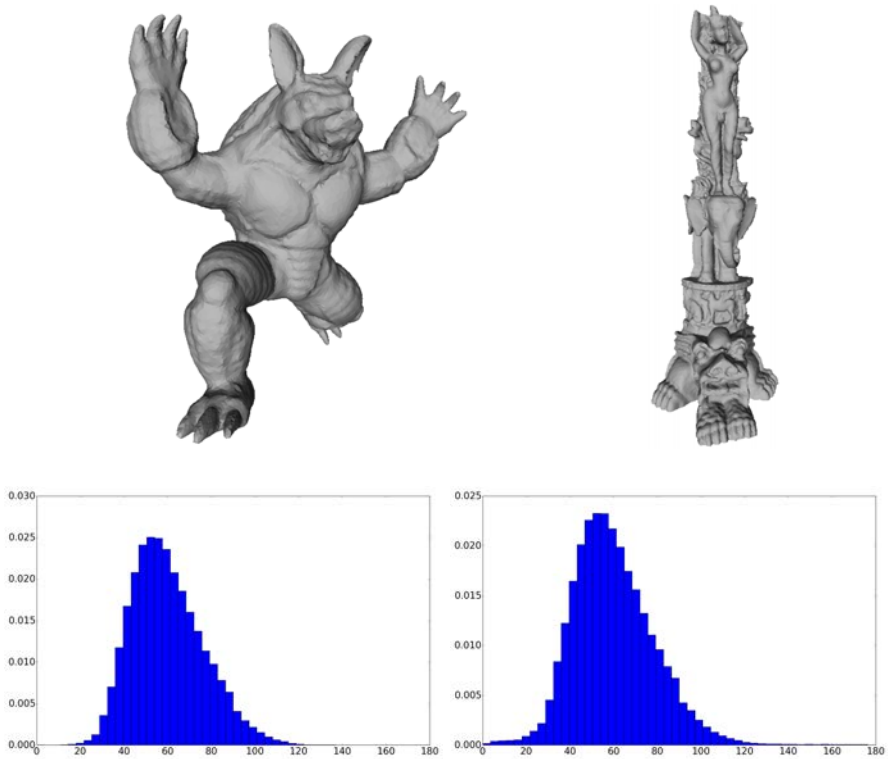
(c) PR (d) YF

(e) Ours

Figure 84: The mesh reconstruction results (Fountain) of BP, YO, PR and YF in [20, 128, 91, 66] respectively, as well as the proposed method in this chapter are evaluated using the method in [155]. YF generates points from images and computes a mesh using Poisson reconstruction [91], while other methods use point data from [70]. The images show the variance weighted depth difference. Red pixels represent locations where no result are obtained or locations where errors larger than 30σ. Green pixels represent the missing scan data of the ground truth. The relative errors between 0 and 30σ are displayed using gray scale from 255 to 0.

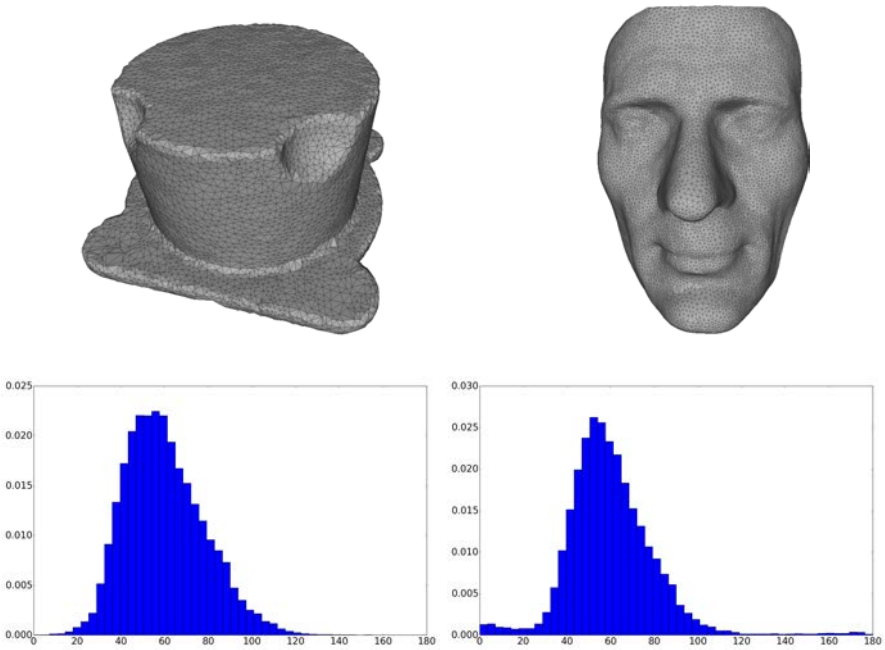


(a) Buddha (b) Dragon



(c) Armadillo (d) Statuette

Figure 85: The resulting meshes of the proposed method (rendering in flat mode) and the corresponding histograms about triangle angles.



(a) Deckel (b) Caesar

Figure 86: The resulting meshes of the proposed method (rendering in wireframe mode) and the corresponding histograms about triangle angles.

7 CONCLUSION

The work in this thesis was motivated by the lack of specialized methods suitable for processing point clouds originating from multi-view reconstruction. The majority of existing work on 3D data processing are oblivious to the particular nature of such data. In order to fill this gap, issues such as the reprojection error, sampling irregularities and noise commonly encountered in the multi-view setting need to be taken into account. In this respect, two novel methods have been developed for point cloud refinement and surface reconstruction. The experimental results substantiate the robustness and efficiency of the proposed methods.

For point cloud refinement, special attention was paid to improving the data quality by filtering noise without impairing correspondences between the scene and images in multi-view data. The method mathematically formulates the problem as a minimization of a smoothness measure combined with bundle adjustment for constraining the reprojection error. Existing work related to bundle adjustment devoted a lot of efforts to numerical optimization, but little work has attempted to fuse or enhance bundle adjustment with other techniques. In this sense, the method proposed herein expands scope of research on bundle adjustment.

The arising constrained optimization problem is resolved efficiently by means of a robust numerical solution procedure. Similar to the bundle adjustment based approach [103], our treatment allows processing large data sets while keeping a reasonable memory usage, but unlike their method we do not require additional sensor data such as the Global Positioning System (GPS) data. As substantiated by experiments on synthetic and real world acquired data, the method successfully preserves geometric features and prevents over-smoothing. Moreover, the experimental result on the synthetic data shows that the reprojection error, surprisingly, can decrease by applying our method. Although the current method requires sufficient sampling in order to produce an optimal solutions, our empirical results show that the shrinkage problem is correctly resolved around areas where data is missing such as holes.

Not only does the proposed method enhance visual appearance of multi-view data, but it also improves the matches in multi-view data. Therefore, it is fairly suitable for applications which require an accurate and consistent position tracking, e.g., photogrammetry. In the future, the method can be generalized to time-dependent data such as a dynamic sequence of point clouds acquired from multi-view videos.

In this scenario, besides the correspondence between the scene and images, the relationships between consecutive point clouds in the sequence need to be addressed. Therefore, a more efficient numerical solution might be required as the arising problem is more complex.

The other focus of this thesis is surface reconstruction. The proposed method addresses new challenges raised by multi-view data such as multilayers and highly irregular sampling which can cause existing approaches to fail. By intertwining a sphere packing methodology with an advancing front strategy, a simple and efficient procedure has been developed to triangulate point clouds into high quality meshes. The experimental evaluation has been performed on both multi-view data and range sensor data.

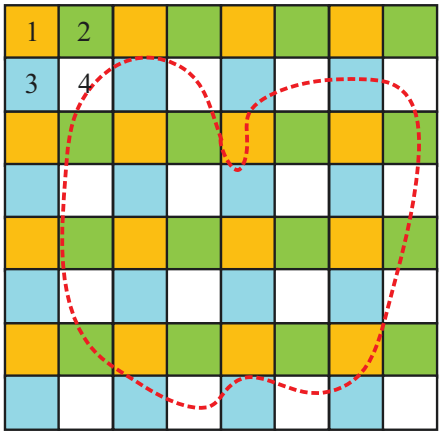


Figure 87: The basic idea to parallelize the method proposed in Chapter 6 is illustrated. The space is divided by the grid cells with the same size. The points in the cells with the same colors can be processed simultaneously when a proper cell size is prescribed.

In principle, the method is based on an advancing front strategy operating locally with a global sphere packing criterion. An optimal balance has been achieved by deliberately combing the local and the global techniques, and thus the method exhibits both favorable local and global properties. The local property is represented by the aspect that distant triangles are created independently. As a result, the techniques of streaming [24] and parallel computing [53] can be easily applied, which improves the practical space and time efficiency of the method. The global property is reflected by the fact that the method can overcome the irregular sampling issue and yield high quality meshes. This originates in sphere packing which approximates a Voronoi diagram. The feature is particularly favorable for applications using the finite elements method such as physical-based animation. In spite of the global property, the proposed method doesn't apply any tedious global optimization while still can produce promising results, which contrasts with other sphere packing based methods [149, 143], Last but not least, the method creates triangles only

using the points from the input point data (possibly with slightly constrained movement) to diminish the potential errors, which can benefit applications that require accurate position information such as tracking in computer vision.

As a part of future work, the method can be parallelized by taking advantage of its locality. The basic idea proposed in [53, 29] can be reused as illustrated in Figure 87. The space bounding the input point cloud is divided by grid cells. If a proper cell size is prescribed, cells with the same colors can be processed simultaneously, and in this way the method can be parallelized. In the future, theoretical guarantees of the method with respect to geometric and topological correctness can be also explored in the light of the work by [56].

BIBLIOGRAPHY

[1] *Intel Math Kernel Library. Reference Manual*. Intel Corporation, Santa Clara, USA, 2009. ISBN 630813-054US.

[2] Acute3D. Smart3DCapture. http://www.acute3d.com/smart3dcapture/.

[3] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M. Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[4] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54 (11):4311–4322, 2006.

[5] AIM@SHAPE. Survey acquisition and reconstruction. Technical report, 2004.

[6] Amel Aissaoui, Jean Martinet, and Chaabane Djeraba. 3d face reconstruction in a binocular passive stereoscopic system using face properties. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 1789–1792. IEEE, 2012.

[7] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T. Silva. Computing and rendering point set surfaces. *Visualization and Computer Graphics, IEEE Transactions on*, 9(1):3–15, 2003.

[8] Oleg Alexander, Mike Rogers, William Lambeth, Matt Chiang, and Paul Debevec. The digital emily project: photoreal facial modeling and animation. In *ACM SIGGRAPH 2009 Courses*, SIGGRAPH '09, pages 12:1–12:15, New York, NY, USA, 2009. ACM. doi: 10.1145/1667239.1667251. URL http://doi.acm.org/10.1145/1667239.1667251.

[9] Remi Allegre, Raphaëlle Chaine, and Samir Akkouche. Convection-driven dynamic surface reconstruction. In *Shape Modeling and Applications, 2005 International Conference*, pages 33–42. IEEE, 2005.

[10] Pierre Alliez, Giuliana Ucelli, Craig Gotsman, and Marco Attene. Recent advances in remeshing of surfaces. In *Shape analysis and structuring*, pages 53–82. Springer, 2008.

[11] Nina Amenta and Marshall Bern. Surface reconstruction by voronoi filtering. *Discrete & Computational Geometry*, 22(4):481–504, 1999.

[12] Nina Amenta, Marshall Bern, and David Eppstein. The crust and the β-skeleton: Combinatorial curve reconstruction. *Graphical models and image processing*, 60(2):125–135, 1998.

[13] Nina Amenta, Sunghee Choi, Tamal K. Dey, and Naveen Leekha. A simple algorithm for homeomorphic surface reconstruction. In *Proceedings of the sixteenth annual symposium on Computational geometry*, SCG '00, pages 213–222, New York, NY, USA, 2000. ACM. ISBN 1-58113-224-7. doi: 10.1145/336154.336207. URL http://doi.acm.org/10.1145/336154.336207.

[14] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 249–266. ACM, 2001.

[15] ASUS. Asus xtion pro live. http://www.asus.com/Multimedia/Xtion_PRO_LIVE/.

[16] Marco Attene and Bianca Falcidieno. Remesh: An interactive environment to edit and repair triangle meshes. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006*, SMI '06, pages 41–, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2591-1. doi: 10.1109/SMI.2006.29. URL http://dx.doi.org/10.1109/SMI.2006.29.

[17] Marco Attene and Bianca Falcidieno. Remesh: An interactive environment to edit and repair triangle meshes. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, pages 41–41. IEEE, 2006.

[18] Haim Avron, Andrei Sharf, Chen Greif, and Daniel Cohen-Or. ℓ^1-sparse reconstruction of sharp point set surfaces. *ACM Transactions on Graphics (TOG)*, 29(5):135:1–135:12, November 2010. ISSN 0730-0301. doi: 10.1145/1857907.1857911. URL http://doi.acm.org/10.1145/1857907.1857911.

[19] Ivo Babuška, Gabriel Caloz, and John E. Osborn. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM J. Numer. Anal.*, 31(4):945–981, August 1994. ISSN 0036-1429. doi: 10.1137/0731051. URL http://dx.doi.org/10.1137/0731051.

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

[21] Christopher M. Bishop and Nasser M. Nasrabadi. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[22] Harry Blum. A transformation for extracting new descriptors of shape. *Models for the perception of speech and visual form*, 19(5): 362–380, 1967.

[23] Jean-Daniel Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Trans. Graph.*, 3(4):266–286, October 1984. ISSN 0730-0301. doi: 10.1145/357346.357349. URL http://doi.acm.org/10.1145/357346.357349.

[24] Matthew Bolitho, Michael Kazhdan, Randal Burns, and Hugues Hoppe. Multilevel streaming for out-of-core surface reconstruction. In *Symposium on geometry processing*, pages 69–78. Citeseer, 2007.

[25] John Adrian Bondy and Uppaluri Siva Ramachandra Murty. Graph theory (graduate texts in mathematics, vol. 244), 2008.

[26] Mario Botsch, Stephan Steinberg, Stephan Bischoff, and Leif Kobbelt. Openmesh - a generic and efficient polygon mesh data structure, 2002.

[27] Mario Botsch, Michael Spernat, and Leif Kobbelt. Phong splatting. In *Proceedings of the First Eurographics Conference on Point-Based Graphics*, SPBG'04, pages 25–32, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association. ISBN 3-905673-09-6. doi: 10.2312/SPBG/SPBG04/025-032. URL http://dx.doi.org/10.2312/SPBG/SPBG04/025-032.

[28] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon mesh processing*. CRC press, 2010.

[29] John Bowers, Rui Wang, Li-Yi Wei, and David Maletz. Parallel poisson disk sampling with spectrum analysis on surfaces. In *ACM Transactions on Graphics (TOG)*, volume 29, page 166. ACM, 2010.

[30] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

[31] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.

[32] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM Sigmod Record*, volume 29, pages 93–104. ACM, 2000.

[33] Martin Dietrich Buhmann. *Radial basis functions: theory and implementations*, volume 12. Cambridge university press, 2003.

[34] Jonathan C. Carr, Richard K. Beatson, Jon B. Cherrie, Tim J. Mitchell, W. Richard Fright, Bruce C. McCallum, and Tim R. Evans. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 67–76. ACM, 2001.

[35] Raphaëlle Chaine. A geometric convection approach of 3-d reconstruction. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '03, pages 218–229, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. ISBN 1-58113-687-0. URL http://dl.acm.org/citation.cfm?id=882370.882400.

[36] Jie Chen and Baoquan Chen. Architectural modeling from sparsely scanned range data. *International Journal of Computer Vision*, 78(2-3):223–236, 2008.

[37] Yanqing Chen, Timothy A. Davis, William W. Hager, and Sivasankaran Rajamanickam. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Trans. Math. Softw.*, 35(3):22:1–22:14, October 2008. ISSN 0098-3500. doi: 10.1145/1391989.1391995. URL http://doi.acm.org/10.1145/1391989.1391995.

[38] Yizong Cheng. Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8): 790–799, 1995.

[39] Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno. Metro: Measuring error on simplified surfaces. Technical report, Paris, France, France, 1996.

[40] Microsoft Corporation. Direct3D. http://msdn.microsoft.com/en-us/library/windows/desktop/hh309466, .

[41] Microsoft Corporation. Kinect. http://www.xbox.com/en-US/kinect/, .

[42] Wilbur B. Davenport and William L. Root. *Random signals and noise*. McGraw-Hill New York, 1958.

[43] Timothy A. Davis. *Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006. ISBN 0898716136.

[44] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and

curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324, 1999.

[45] Tamal K. Dey. *Curve and surface reconstruction: algorithms with mathematical analysis*, volume 23. Cambridge University Press, 2007.

[46] Tamal K. Dey and Samrat Goswami. Tight cocone: a watertight surface reconstructor. In *Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 127–134. ACM, 2003.

[47] Tamal K. Dey and Samrat Goswami. Provable surface reconstruction from noisy samples. *Computational Geometry*, 35(1): 124–141, 2006.

[48] Tamal K. Dey and Jian Sun. . an adaptive mls surface for reconstruction with guarantees. In *Symposium on Geometry Processing*, pages 43–52, 2005.

[49] Tamal K. Dey and Lei Wang. Voronoi-based feature curves extraction for sampled singular surfaces. *Computers & Graphics*, 2013.

[50] Tamal K. Dey, Joachim Giesen, and James Hudson. Delaunay based shape reconstruction from large data. In *Parallel and Large-Data Visualization and Graphics, 2001. Proceedings. IEEE 2001 Symposium on*, pages 19–146. IEEE, 2001.

[51] Tamal Krishna Dey, Chanderjit L. Bajaj, and Kokichi Sugihara. On good triangulations in three dimensions. *International Journal of Computational Geometry & Applications*, 2(01):75–95, 1992.

[52] Tamal Krishna Dey, Xiaoyin Ge, Qichao Que, Issam Safa, Lei Wang, and Yusu Wang. Feature-preserving reconstruction of singular surfaces. In *Computer Graphics Forum*, volume 31, pages 1787–1796. Wiley Online Library, 2012.

[53] Julie Digne. An analysis and implementation of a parallel ball pivoting algorithm. *IPOL*, pages 1–17, August 2013. ISSN 2105-1232. URL http://liris.cnrs.fr/publis/?id=6246. accepted for publication - to appear.

[54] Julie Digne, Jean-Michel Morel, Charyar-Mehdi Souzani, and Claire Lartigue. Scale space meshing of raw data point sets. In *Computer Graphics Forum*, volume 30, pages 1630–1642. Wiley Online Library, 2011.

[55] Jean Duchon. Splines minimizing rotation-invariant seminorms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer, 1977.

[56] Herbert Edelsbrunner and Ernst P Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics (TOG)*, 13(1):43–72, 1994.

[57] Carlos A. Felippa. Introduction to Finite Element Methods. http://www.colorado.edu/engineering/cas/courses.d/IFEM.d/, 2013.

[58] Vittorio Ferrari, Tinne Tuytelaars, and Luc Van Gool. Simultaneous object recognition and segmentation by image exploration. In T. Pajdla and J. Matas, editors, *European Conference on Computer Vision (ECCV)*, volume I, pages 40–54. Springer, May 2004.

[59] Shachar Fleishman, Iddo Drori, and Daniel Cohen-Or. Bilateral mesh denoising. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 950–953. ACM, 2003.

[60] Shachar Fleishman, Daniel Cohen-Or, and Cláudio T. Silva. Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.*, 24(3):544–552, July 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073227. URL http://doi.acm.org/10.1145/1073204.1073227.

[61] Blender Foundation. Blender. http://www.blender.org/.

[62] Pascal Jean Frey and Paul-Louis George. Mesh generation. application to finite elements. hermes science publ., paris, 2000.

[63] Pascal Fua. Using model-driven bundle-adjustment to model heads from raw video sequences. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999.*, volume 1, pages 46–53, 1999.

[64] Pascal Fua. Regularized bundle-adjustment to model heads from image sequences without calibration data. *International Journal of Computer Vision*, 38(2):153–171, 2000.

[65] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on*, 21(1):32–40, 1975.

[66] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[67] Yasutaka Furukawa and Jean Ponce. Dense 3d motion capture for human faces. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1674–1681. IEEE, 2009.

[68] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1362–1376, 2010.

[69] Yasutaka Furukawa and Jean Ponce. PMVS - Version 2. http://www.di.ens.fr/pmvs/, 2010.

[70] Patricio Galindo and Rhaleb Zayer. Distortion driven variational multi-view reconstruction. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 649–656. IEEE, 2014.

[71] Paul Louis George and Eric Seveno. The advancing-front mesh generation method revisited. *International Journal for Numerical Methods in Engineering*, 37(21):3605–3619, 1994.

[72] Leslie Greengard and Vladimir Rokhlin. A fast algorithm for particle simulations. *Journal of computational physics*, 73(2):325–348, 1987.

[73] Point grey. Ladybug sphereical cameras. http://ww2.ptgrey.com/products/ladybug5/.

[74] Eitan Grinspun, Petr Krysl, and Peter Schröder. Charms: a simple framework for adaptive simulation. In *SIGGRAPH*, volume 2, pages 281–290, 2002.

[75] Gaël Guennebaud, Marcel Germann, and Markus Gross. Dynamic sampling and rendering of algebraic point set surfaces. In *Computer Graphics Forum*, volume 27, pages 653–662. Wiley Online Library, 2008.

[76] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[77] Martin Habbecke and Leif Kobbelt. A surface-growing approach to multi-view stereo reconstruction. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[78] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000.

[79] Paul S. Heckbert and Michael Garland. Optimal triangulation and quadric-based surface simplification. *Comput. Geom. Theory Appl.*, 14(1-3):49–65, November 1999. ISSN 0925-7721. doi: 10.1016/S0925-7721(99)00030-9. URL http://dx.doi.org/10.1016/S0925-7721(99)00030-9.

[80] Vu Hoang Hiep, Renaud Keriven, Patrick Labatut, and J-P Pons. Towards high-resolution large-scale multi-view stereo. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1430–1437. IEEE, 2009.

[81] Hugues Hoppe. *Surface reconstruction from unorganized points* . PhD thesis, University of Washington, june 1994.

[82] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. *SIGGRAPH Comput. Graph.*, 26(2):71–78, July 1992. ISSN 0097-8930. doi: 10.1145/142920.134011. URL http://doi.acm.org/10.1145/142920.134011.

[83] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '93, pages 19–26, New York, NY, USA, 1993. ACM. ISBN 0-89791-601-8. doi: 10.1145/166117.166119. URL http://doi.acm.org/10.1145/166117.166119.

[84] Hui Huang, Dan Li, Hao Zhang, Uri Ascher, and Daniel Cohen-Or. Consolidation of unorganized point clouds for surface reconstruction. In *ACM Transactions on Graphics (TOG)*, volume 28, page 176. ACM, 2009.

[85] Ioannis Ivrissimtzis, Won-Ki Jeong, and Hans-Peter Seidel. Using growing cell structures for surface reconstruction. In *Shape Modeling International, 2003*, pages 78–86. IEEE, 2003.

[86] Ioannis Ivrissimtzis, Won-Ki Jeong, Seungyong Lee, Man Hee Lee, and Hans-Peter Seidel. Neural meshes: surface reconstruction with a learning algorithm. In *SaarbrÃŒcken: Max-Planck-Institut fÃŒr Informatik.*, 2004.

[87] Thouis R. Jones, Frédo Durand, and Mathieu Desbrun. Non-iterative, feature-preserving mesh smoothing. *ACM Transactions on Graphics (TOG)*, 22(3):943–949, July 2003. ISSN 0730-0301. doi: 10.1145/882262.882367. URL http://doi.acm.org/10.1145/882262.882367.

[88] Juho Kannala and Sami S. Brandt. Quasi-dense wide baseline matching using match propagation. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[89] Juho Kannala, Esa Rahtu, Sami S. Brandt, and Janne Heikkila. Object recognition and segmentation by non-rigid quasi-dense matching. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[90] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3):29:1–29:13, July 2013. ISSN 0730-0301. doi: 10.1145/2487228.2487237. URL http://doi.acm.org/10.1145/2487228.2487237.

[91] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP '06, pages 61–70. Eurographics Association, 2006. ISBN 3-905673-36-3. URL http://dl.acm.org/citation.cfm?id=1281957.1281965.

[92] Khronos Group. OpenGL. http://http://www.opengl.org/.

[93] David G. Kleinbaum. *Applied regression analysis and multivariable methods*. CengageBrain. com, 2007.

[94] Leif Kobbelt and Mario Botsch. A survey of point-based techniques in computer graphics. *Comput. Graph.*, 28(6):801–814, December 2004. ISSN 0097-8493. doi: 10.1016/j.cag.2004.08.009. URL http://dx.doi.org/10.1016/j.cag.2004.08.009.

[95] Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 105–114. ACM, 1998.

[96] Teuvo Kohonen. *Self-organizing maps*, volume 30. Springer, 2001.

[97] Pekka Koskenkorva, Juho Kannala, and Sami S. Brandt. Quasi-dense wide baseline matching for three views. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 806–809. IEEE, 2010.

[98] Hideyuki Kume, Takafumi Taketomi, Tomokazu Sato, and Naokazu Yokoya. Extrinsic camera parameter estimation using video images and gps considering gps positioning accuracy. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10, pages 3923–3926. IEEE Computer Society, 2010. ISBN 978-0-7695-4109-9. doi: 10.1109/ICPR.2010.954. URL http://dx.doi.org/10.1109/ICPR.2010.954.

[99] Florent Lafarge and Pierre Alliez. Surface reconstruction through point set structuring. In *Computer Graphics Forum*, volume 32, pages 225–234. Wiley Online Library, 2013.

[100] Florent Lafarge and Clément Mallet. Creating large-scale city models from 3d-point clouds: a robust approach with hybrid representation. *International journal of computer vision*, 99(1):69–85, 2012.

[101] Victor Lempitsky and Yuri Boykov. Global optimization for shape fitting. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR '07.*, pages 1 –8, june 2007. doi: 10.1109/CVPR.2007.383293.

[102] David Levin. Mesh-independent surface interpolation. In *Geometric modeling for scientific visualization*, pages 37–49. Springer, 2004.

[103] Maxime Lhuillier. Fusion of gps and structure-from-motion using constrained bundle adjustments. In *2011 IEEE Conference onComputer Vision and Pattern Recognition (CVPR)*, pages 3025–3032. IEEE, 2011.

[104] Maxime Lhuillier and Long Quan. Match propogation for image-based modeling and rendering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8):1140–1146, August 2002. ISSN 0162-8828. doi: 10.1109/TPAMI.2002.1023810. URL http://dx.doi.org/10.1109/TPAMI.2002.1023810.

[105] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):418–433, 2005.

[106] Peter Liepa. Filling holes in meshes. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, SGP '03, pages 200–205, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. ISBN 1-58113-687-0. URL http://dl.acm.org/citation.cfm?id=882370.882397.

[107] Yaron Lipman, Daniel Cohen-Or, David Levin, and Hillel Tal-Ezer. Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph.*, 26(3), July 2007. ISSN 0730-0301. doi: 10.1145/1276377.1276405. URL http://doi.acm.org/10.1145/1276377.1276405.

[108] Rainald Löhner and Paresh Parikh. Generation of three-dimensional unstructured grids by the advancing-front method. *International Journal for Numerical Methods in Fluids*, 8(10):1135–1149, 1988.

[109] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM Siggraph Computer Graphics*, volume 21, pages 163–169. ACM, 1987.

[110] Manolis IA. Lourakis and Antonis A. Argyros. Sba: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software (TOMS)*, 36(1):2, 2009.

[111] David G. Luenberger and Yinyu Ye. *Linear and nonlinear programming*, volume 116. Springer Verlag, 2008.

[112] Subhash Chandra Malik and Savita Arora. *Mathematical analysis*. New Age International, 1992.

[113] Martin Marinov and Leif Kobbelt. Optimization methods for scattered data approximation with subdivision surfaces. *Graphical Models*, 67(5):452–473, 2005.

[114] William Massey. *A basic course in algebraic topology*, volume 127. Springer, 1991.

[115] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[116] Autodesk Media and Entertainment. Autodesk 123D Catch. http://www.123dapp.com/catch.

[117] Peter Meer, Doron Mintz, Azriel Rosenfeld, and Dong Yoon Kim. Robust regression methods for computer vision: A review. *International journal of computer vision*, 6(1):59–70, 1991.

[118] Theo Moons, Luc Van Gool, and Maarten Vergauwen. *3d Reconstruction from Multiple Images: Part 1: Principles*. Now Publishers Inc, 2010.

[119] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Generic and real-time structure from motion using local bundle adjustment. *Image and Vision Computing*, 27(8):1178–1193, 2009.

[120] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2, 2009.

[121] MV4D. Mantis Vision. http://http://www.mv4d.com/.

[122] Yukie Nagai, Yutaka Ohtake, and Hiromasa Suzuki. Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction. In *Proceedings of the Symposium on Geometry Processing*, SGP '09, pages 1339–1348, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association. URL http://dl.acm.org/citation.cfm?id=1735603.1735615.

[123] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ISMAR '11, pages 127–136, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-1-4577-2183-0. doi: 10.1109/ISMAR.2011.6092378. URL http://dx.doi.org/10.1109/ISMAR.2011.6092378.

[124] Kai Ni, Drew Steedly, and Frank Dellaert. Out-of-core bundle adjustment for large-scale 3d reconstruction. In *Computer Vision,*

2007. ICCV 2007. IEEE 11th International Conference on, pages 1–8. IEEE, 2007.

[125] Vincent Nivoliers, Dongming Yan, and Bruno LÃ©vy. Fitting polynomial surfaces to triangular meshes with voronoi squared distance minimization. In *International Meshing Roundtable conference proceedings*, 2011.

[126] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer verlag, 1999.

[127] Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. Multi-level partition of unity implicits. *ACM Trans. Graph.*, 22(3):463–470, July 2003. ISSN 0730-0301. doi: 10.1145/882262.882293. URL http://doi.acm.org/10.1145/882262.882293.

[128] Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. An integrating approach to meshing scattered point data. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, SPM '05, pages 61–69, New York, NY, USA, 2005. ACM. ISBN 1-59593-015-9. doi: 10.1145/1060244.1060252. URL http://doi.acm.org/10.1145/1060244.1060252.

[129] Joseph o'Rourke. *Computational geometry in C*. Cambridge university press, 1998.

[130] A Cengiz Öztireli, Gaël Guennebaud, and Markus Gross. Feature preserving point set surfaces based on non-linear kernel regression. In *Computer Graphics Forum*, volume 28, pages 493–501. Wiley Online Library, 2009.

[131] Philippe P. Pebay and Timothy J. Baker. A comparison of triangle quality measures. In *Proceedings of the National Academy of Sciences*, pages 9653–9657. Plenum Press, 1991.

[132] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, July 2003. ISSN 0730-0301. doi: 10.1145/882262.882269. URL http://doi.acm.org/10.1145/882262.882269.

[133] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing(3rd ed.)*. Cambridge University Press, 2007.

[134] PrimeSense. PrimeSense Carmine. http://www.primesense.com/.

[135] Stuart Robson, Mona Hess, Francesca Simon Millar, Graeme Were, Ian Brown, Sally MacDonald, and Yean-Hoon Ong. Traceable storage and transmission of 3d colour scan data sets. 2008.

[136] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[137] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003. ISBN 0898715342.

[138] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1352–1359. IEEE, 2013.

[139] Nader Salman and Mariette Yvinec. Surface Reconstruction from Multi-View Stereo. *Lecture notes in computer science*, September 2009.

[140] Oliver Schall and Marie Samozino. Surface from Scattered Points: A Brief Survey of Recent Developments. In Bianca Falcidieno and Nadia Magnenat-Thalmann, editors, *1st International Workshop on Semantic Virtual Environments*, pages 138–147, Villars, Switzerland, 2005. MIRALab.

[141] Oliver Schall, Alexander Belyaev, and Hans-Peter Seidel. Robust filtering of noisy scattered point data. In *Proceedings of the Second Eurographics/IEEE VGTC conference on Point-Based Graphics*, pages 71–77. Eurographics Association, 2005.

[142] Carlos E. Scheidegger, Shachar Fleishman, and Cláudio T. Silva. Triangulating point set surfaces with bounded error. In *Proceedings of the third Eurographics symposium on Geometry processing*, SGP '05, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association. ISBN 3-905673-24-X. URL http://dl.acm.org/citation.cfm?id=1281920.1281930.

[143] Alexander Schiftner, Mathias Höbinger, Johannes Wallner, and Helmut Pottmann. Packing circles and spheres on surfaces. *ACM Trans. Graph.*, 28(5):139:1–139:8, December 2009. ISSN 0730-0301. doi: 10.1145/1618452.1618485. URL http://doi.acm.org/10.1145/1618452.1618485.

[144] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer Graphics Forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[145] Joachim Schöberl. Netgen an advancing front 2d/3d-mesh generator based on abstract rules. *Computing and visualization in science*, 1(1):41–52, 1997.

[146] John Schreiner, Carlos E Scheidegger, Shachar Fleishman, and Cláudio T Silva. Direct (re) meshing for efficient surface processing. In *Computer graphics forum*, volume 25, pages 527–536. Wiley Online Library, 2006.

[147] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 519–528. IEEE, 2006.

[148] Jonathan Shewchuk. What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint). *University of California at Berkeley*, 73, 2002.

[149] Kenji Shimada and David C. Gossard. Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing. In *Proceedings of the third ACM symposium on Solid modeling and applications*, SMA '95, pages 409–419, New York, NY, USA, 1995. ACM. ISBN 0-89791-672-7. doi: 10.1145/218013.218095. URL http://doi.acm.org/10.1145/218013.218095.

[150] Daniel Sieger, Pierre Alliez, and Mario Botsch. Optimizing voronoi diagrams for polygonal finite element computations. In *Proceedings of the 19th international meshing roundtable*, pages 335–350. Springer, 2010.

[151] Neil JA Sloane, John Conway, et al. *Sphere packings, lattices and groups*, volume 290. Springer, 1999.

[152] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *Int. J. Comput. Vision*, 80(2):189–210, November 2008. ISSN 0920-5691. doi: 10.1007/s11263-007-0107-3. URL http://dx.doi.org/10.1007/s11263-007-0107-3.

[153] Soheil Sotoodeh. Outlier detection in laser scanner point clouds. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences XXXVI-5, 297e302*, 2006.

[154] Kenneth Stephenson. *Introduction to circle packing: The theory of discrete analytic functions*. Cambridge University Press, 2005.

[155] Christoph Strecha, Wolfgang von Hansen, Luc Van Gool, Pascal Fua, and Ulrich Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[156] Tomáš Svoboda, Daniel Martinec, and Tomáš Pajdla. A convenient multicamera self-calibration for virtual environments. *Presence: Teleoperators & virtual environments*, 14(4):407–422, 2005.

[157] Richard Szeliski. *Computer vision: algorithms and applications*. Springer, 2010.

[158] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM, 1995.

[159] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[160] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment – a modern synthesis. *Vision algorithms: theory and practice*, pages 153–177, 2000.

[161] LAS Specification Version. Version 1.3–r10. *The American Society for Photogrammetry & Remote Sensing (ASPRS)*, 2009.

[162] Viametris. Viametris I-MMS. http://http://http://viametris.fr//.

[163] Hoang-Hiep Vu, Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. High accuracy and visibility-consistent dense multiview stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):889–901, May 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.172.

[164] Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

[165] Max Wardetzky, Saurabh Mathur, Felix Kälberer, and Eitan Grinspun. Discrete laplace operators: no free lunch. In *Symposium on Geometry processing*, pages 33–37, 2007.

[166] Claus Weitkamp. *Lidar: range-resolved optical remote sensing of the atmosphere*, volume 102. Springer, 2005.

[167] Tim Weyrich, Mark Pauly, Richard Keiser, Simon Heinzle, Sascha Scandella, and Markus Gross. Post-processing of scanned 3d surface data. In *Proceedings of the First Eurographics conference on Point-Based Graphics*, SPBG'04, pages 85–94, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association. ISBN 3-905673-09-6. doi: 10.2312/SPBG/SPBG04/085-094. URL http://dx.doi.org/10.2312/SPBG/SPBG04/085-094.

[168] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M Seitz. Schematic surface reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1498–1505. IEEE, 2012.

[169] Shiyao Xiong, Juyong Zhangy, Jianmin Zheng, Jianfei Cai, and Ligang Liu. Robust surface reconstruction via dictionary learning. *ACM Transactions on Graphics (TOG)*, 33(6):201, 2014.

[170] XYZ RGB Inc. Scannerkiller. http://www.scannerkiller.com/.

[171] Dong-Ming Yan, Bruno Lévy, Yang Liu, Feng Sun, and Wenping Wang. Isotropic remeshing with fast and exact computation of restricted voronoi diagram. In *Computer graphics forum*, volume 28, pages 1445–1454. Wiley Online Library, 2009.

[172] Yizhou Yu. Surface reconstruction from unorganized points using self-organizing neural networks. In *IEEE Visualization*, volume 99, pages 61–64. Citeseer, 1999.

[173] Yizhou Yu, Kun Zhou, Dong Xu, Xiaohan Shi, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Mesh editing with poisson-based gradient field manipulation. *ACM Trans. Graph.*, 23(3): 644–651, August 2004. ISSN 0730-0301. doi: 10.1145/1015706.1015774. URL http://doi.acm.org/10.1145/1015706.1015774.

[174] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666–673 vol.1, 1999. doi: 10.1109/ICCV.1999.791289.

[175] Matthias Zwicker, Mark Pauly, Oliver Knoll, and Markus Gross. Pointshop 3d: an interactive system for point-based surface editing. *ACM Trans. Graph.*, 21(3):322–329, July 2002. ISSN 0730-0301. doi: 10.1145/566654.566584. URL http://doi.acm.org/10.1145/566654.566584.

ABSTRACT

Point cloud refinement and surface reconstruction are two fundamental problems in geometry processing. Most of the existing methods have been targeted at range sensor data and turned out be ill-adapted to multi-view data. In this thesis, two novel methods are proposed respectively for the two problems with special attention to multi-view data. The first method smooths point clouds originating from multi-view reconstruction without impairing the data. The problem is formulated as a nonlinear constrained optimization and addressed as a series of unconstrained optimization problems by means of a barrier method. The second method triangulates point clouds into meshes using an advancing front strategy directed by a sphere packing criterion. The method is algorithmically simple and can produce high-quality meshes efficiently. The experiments on synthetic and real-world data have been conducted as well, which demonstrates the robustness and the efficiency of the methods. The developed methods are suitable for applications which require accurate and consistent position information such photogrammetry and tracking in computer vision.

RÉSUMÉ

Le raffinement de nuage de points et la reconstruction de surface sont deux problèmes fondamentaux dans le traitement de la géométrie. La plupart des méthodes existantes ont été ciblées sur les données de capteur de distance et se sont avérées être mal adaptées aux données multi-vues. Dans cette thèse, deux nouvelles méthodes sont proposées respectivement pour les deux problèmes avec une attention particulière aux données multi-vues. La première méthode permet de lisser les nuages de points provenant de la reconstruction multi-vue sans endommager les données. Le problème est formulé comme une optimisation non-linéaire sous contrainte et ensuite résolu par une série de problèmes d'optimisation sans contrainte au moyen d'une méthode de barrière. La seconde méthode effectue une triangulation du nuage de points d'entrée pour générer un maillage en utilisant une stratégie de l'avancement du front pilotée par un critère de l'empilement compact de sphères. L'algorithme est simple et permet de produire efficacement des maillages de haute qualité. Les expérimentations sur des données synthétiques et du monde réel démontrent la robustesse et l'efficacité des méthodes proposées. Notre méthodes sont adaptées aux applications qui nécessitent des informations de position précises et cohérentes telles que la photogrammétrie et le suivi des objets en vision par ordinateur.