Survey of 3D modeling using depth cameras

Hantong XU, Jiamin XU, Weiwei XU

State Key Lab of CAD&CG, Zhejiang University, Hangzhou 310058, China

* Corresponding author, xww@cad.zju.edu.cn
Received: 28 July 2019  Accepted: 15 September 2019

Supported by National Natural Science Foundation of China (61732016).

Citation: Hantong XU, Jiamin XU, Weiwei XU. Survey of 3D modeling using depth cameras. Virtual Reality & Intelligent Hardware, 2019, 1(5): 483—499
DOI: 10.1016/j.vrih.2019.09.003

Abstract  Three-dimensional (3D) modeling is an important topic in computer graphics and computer vision. In recent years, the introduction of consumer-grade depth cameras has resulted in profound advances in 3D modeling. Starting with the basic data structure, this survey reviews the latest developments of 3D modeling based on depth cameras, including research works on camera tracking, 3D object and scene reconstruction, and high-quality texture reconstruction. We also discuss the future work and possible solutions for 3D modeling based on the depth camera.

Keywords  3D Modeling; Depth camera; Camera tracking; Signed distance function; Surfel

1 Introduction

Three-dimensional (3D) modeling is an important research area in computer graphics and computer vision. Its target is to capture the 3D shape and appearance of objects and scenes in high quality to simulate 3D interaction and perception in digital space. The 3D modeling methods can be primarily categorized into three types:

1. Modeling based on professional software, such as 3DS Max and Maya: It requires the user to be proficient with the software. Thus, the learning curve is usually long; further, the creation of 3D content is usually time consuming.

2. Modeling based on two-dimensional (2D) color (RGB) images, such as multiview stereo\(^{(1)}\), structure from motion\(^{(2,3)}\): As image sensors demonstrate low-cost and easy to deploy characteristics, 3D modeling from multi-view 2D images can significantly automate and simplify the modeling process; however, it relies heavily on image quality and are easily affected by lighting and camera resolution.

3. Modeling based on professional, active 3D sensors\(^{(4)}\), including 3D scanner and depth camera: The sensors actively project structured light or laser stripes to the surface of the object and then reconstruct spatial shape of objects and scenes, i.e., sampling a large number of 3D points from the 3D surface according to the spatial information decoded from the structured light. The advantage of 3D sensors is that they are robust to interference in the environment and are able to scan 3D models in high precision automatically. However, professional 3D scanners are expensive, which limits their applicability for ordinary users.

In recent years, the development of depth cameras, i.e., an RGB camera in association with depth sensing devices, called as RGB-D camera, has led to the rapid development of 3D modeling. The depth
camera is not only low-cost and compact, but also capable of capturing pixel-level color and depth information with sufficient resolution and frame rate. Because of these characteristics, depth cameras have a huge advantage in the case of consumer-oriented applications when compared to other expensive scanning devices. The KinectFusion algorithm, proposed by the Imperial College London and Microsoft Research in 2011, performs 3D modeling with depth cameras\(^5,6\). It is capable of generating 3D models with high-resolution details in real time using color and depth images obtained using depth cameras, which is attracting the attention of researchers in the field of 3D reconstruction. Researchers in computer graphics and computer vision have continuously innovated the algorithms and systems for 3D modeling based on depth cameras and achieved satisfactory research results.

This survey will review and compare the latest methods of 3D modeling based on depth cameras. We first present two basic data structures that are predominantly used for 3D modeling using depth cameras in Section 2. In Section 3, we focus on the progress of geometric reconstruction based on depth cameras. Then, in Section 4, the reconstruction of high-quality texture is reviewed. Finally, we conclude and discuss the future work in Section 5.

1.1 Preliminaries of depth cameras

Each frame of data obtained by the depth camera contains the distance from each point in the real scene to the vertical plane where the depth camera is located. This distance is called the depth value; moreover, these depth values at sampled pixels constitute the depth image of a depth frame. Usually, the color image and the depth image are registered; thus, there is a one-to-one correspondence between the pixels (Figure 1).

Currently, most of the depth cameras are developed based on structured light, such as Kinect V1 by Microsoft and Structure Sensor by Occipital, or time of flight (ToF) technology, such as Microsoft Kinect 2.0. A structured-light depth camera first projects the infrared structured light onto the surface of an object, and then receives the light pattern reflected by the surface of the object. As the received pattern is modulated by the 3D shape of the object, the spatial information of the 3D surface can be calculated by the position and degree of modulation of the pattern on the image. The structured-light camera is suitable for scenes with insufficient illumination; moreover, it can achieve high precision within a certain range and produce high-resolution depth images. The disadvantage of structured light is that it is easy to be affected by strong natural light outdoors, which results in the projected coded light becoming submerged and unusable. At the same time, it is susceptible to plane mirror reflection. The ToF camera uses a continuous emission of light pulses (generally invisible light) onto an object to be observed and then receives a pulse of light reflected back from the object. The distance from the measured object to the camera is calculated by detecting the flight (round trip) time of the light pulse. Overall, the measurement error of the ToF depth camera does not increase with the increase of the measurement distance and the anti-interference ability is strong, which is suitable for a relatively long measurement distance (such as automatic driving). The disadvantage is that the depth image obtained is not high in resolution; moreover, the measurement error is larger than that of
other depth cameras when measuring at a close range.

1.2 Pipeline of 3D modeling based on depth cameras

As the depth data obtained by the depth camera inevitably has noise, the depth map is first preprocessed, i.e., the noise and removed outliers are reduced. Next, the camera poses are estimated to register the captured depth images into a consistent coordinate frame. The classic approach is to determine the corresponding points between the depth maps (i.e., data associations) and then estimate the motion of the camera by registering the depth images. Finally, the current depth image is merged into the global model based on the estimated camera pose. By repeating these steps, the geometry of the scene can be obtained. However, to completely reconstruct the appearance of the scene, it is obviously not sufficient to just capture the geometry. We should also provide the model with color and texture.

According to the data structure used to represent the 3D modeling result, we can approximately categorize 3D modeling into volumetric representation-based modeling and surfel-based modeling. The steps of 3D reconstruction will vary based on the different data structures. This review clarifies how each method solves the classic problems in 3D reconstruction by reviewing and comparing the different strategies adopted by the various modeling methods.

2 Data structure for 3D surfaces

2.1 Volumetric representation

The concept of volumetric representation was first proposed in 1996 by Curless and Levoy\textsuperscript{(7)}. They introduced the concept of representing 3D surfaces using signed distance function (SDF) sampled at voxels, where SDF values are defined as the distance of a 3D point to the surface. Thus, the surface of an object or a scene, according to this definition, is an iso-distance surface whose SDF is zero. The free space, i.e., the space outside of the object, is indicated by a positive SDF value, because the distance increases as a point moves along the surface normal towards the free space. Conversely, the occupied space (the space inside of the object) is indicated by negative SDF values. These SDF values are stored in a cube surrounding the object and the cube is rasterized into voxels. The reconstruction algorithm must define the voxel size and spatial extent of the cube. Other data, such as colors, are usually stored as voxel attributes. As only voxels close to the actual surface are important, researchers often use the truncated SDF (TSDF) to represent the model. Figure 2 shows the general pipeline of reconstruction based on volumetric representation. As the TSDF value implicitly represents the surface of the object, it is necessary to extract the surface of the object using ray casting before estimating the pose of the camera. The fusion step is a simple weighted average operation at each voxel center, which is very effective for eliminating the noise of the depth camera.

![Figure 2: Pipeline of volumetric representation-based reconstruction\textsuperscript{16}.](image-url)
Uniform voxel grids are significantly inefficient in terms of memory consumption and are limited by predefined volume and resolutions. In the context of real-time scene reconstruction, most methods rely heavily on the processing power of modern GPUs. The spatial extent and resolution of voxel grids are often limited by GPU memory. To support a larger spatial extent, researchers have proposed a variety of methods to improve the memory efficiency of volumetric representation-based algorithms. To prevent data loss caused by depth images acquired outside of a predefined cube, Whelan et al. proposed a method of dynamically shifting voxel grids such that the voxel grids follow the motion of the depth camera. When the voxel grids move, the method extracts the parts outside the current voxel mesh and stores them separately, adding their camera pose to the global pose. Although this can achieve a larger scan volume, it requires a large amount of external memory usage; moreover, the scanned surface cannot be arbitrarily revisited. Henry et al. also proposed a similar idea.

Voxel hierarchies, such as octrees, are another way to efficiently store the geometry of 3D surfaces. The octree can adaptively subdivide the scene space according to the complexity of the scene, effectively utilizing the memory of the computer. Although its definition is simple, owing to the sparsity of nodes, it is considerably difficult to utilize the parallelism of the GPU. Sun et al. established an octree with only leaf nodes to store voxel data and accelerate tracking based on octree. Zhou et al. used the GPU to construct a complete octree structure to perform Poisson reconstruction on scenes with 300,000 vertices at an interactive rate. Zeng et al. implemented a 9- to 10-level octree for KinectFusion and extended the reconstruction results to a medium-sized office. Chen et al. proposed a similar three-level hierarchy with equally good resolution. Steinbrücker et al. represented the scene in a multi-resolution data structure for real-time accumulation on the CPU; further, they outputted a real-time grid model at approximately 1 Hz.

Nießner et al. proposed a voxel hashing structure. In this method, the index corresponding to the spatial position of the voxel is stored in a linearized spatial hash table and addressed by a spatial hash function. As only voxels containing spatial information are stored in the memory, this strategy significantly reduces memory consumption and theoretically allows voxel blocks of a certain size and resolution to represent infinitely large spaces. When compared with the voxel hierarchies, the hashing structure has great advantages in data insertion and access, whose time complexity is O(1). However, the hashing function inevitably maps multiple different spatial voxel blocks to the same location in the hash table. Kahler et al. proposed the use of a different hashing method to reduce the number of hash collisions. Hash-based 3D reconstruction has the advantage of small memory requirements and efficient calculations, making it suitable for even mobile devices, such as application to Google Tango.

### 2.2 Surfel

Surfel is another important surface representation, which was originally used as the primitive for model rendering. As shown in Figure 3, a surfel usually contains the following attributes: space coordinates \( p \in \mathbb{R}^3 \), normal vector \( n \in \mathbb{R}^3 \), color information \( c \in \mathbb{R}^3 \), weight (or confidence) \( w \), radius \( r \), timestamp \( t \). The weight of a surfel is used for the weighted average of neighboring 3D points and the judgment of the stable and unstable points. It is generally initialized by the following formula: \( w = e^{-\gamma / \sigma r} \), where \( \gamma \) is the normalized radial distance from the current depth measurement to the center of the camera; further, \( \sigma \) is usually considered as 0.6. The radius of a surfel is determined by the

![Schematic diagram of surfel.](image-url)
distance of the surface of the scene from the optical center of the camera. The further the distance is, the larger the radius of the surfel will be. When compared to the volumetric representation, the surfel-based 3D reconstruction is more compact and intuitive for input from depth cameras, eliminating the need to switch back and forth between different representations. Similar to volumetric representation, octrees can be used to improve the surfel-based data structures\(^\text{[21,22]}\).

**Pros and cons:** Volumetric representation is more convenient in depth denoising and surface rendering as the TSDF computed according to each frame can be efficiently fused at each voxel, as discussed in the next section. In contrast, the surfel representation can be viewed as an extension of the point cloud representation. When the vertex correspondences are necessary in the registration, such information can be directly computed using surfels, while the volumetric representation has to be converted to 3D meshes in this case.

### 3D Reconstruction based on depth cameras

#### 3.1 Depth map preprocessing

The depth image obtained by the depth camera inevitably contains noise or even outliers owing to the complex lighting condition, geometric variations, and spatially variant materials of the objects. Mallick et al.\(^\text{[23]}\) classified these noises into the three following categories. The first is missing depth. When the object is too close or too far from the depth camera, the camera sets the depth value to zero because it cannot measure the depth. It should be noted that even within the measurement range of the camera, zero depth may be generated owing to surface discontinuities, highlights, shadows, or other reasons. The second is depth error. When a depth camera records a depth value, its accuracy depends on its depth itself. The third is depth inconsistency. Even if the depth of the scene is constant, the depth value obtained will change with time.

In most cases, bilateral filtering\(^\text{[24]}\) is used to reduce the noise of the depth image. After noise reduction, the KinectFusion\(^\text{[10]}\) application implies a downsampling to obtain a three-layer depth map pyramid to estimate the pose of the camera from coarse to fine in the subsequent step. For the depth image after noise reduction, according to the internal parameters of the depth camera, the 3D coordinates of each pixel point in the camera coordinate system can be calculated by inverse projection to form a vertex map. The normal of each vertex can be obtained by the cross product of vectors formed by the vertex and its adjacent vertices. For the surfel representation, it is also necessary to calculate the radius of each point to represent the local surface area around a given point while minimizing the visible holes between adjacent points, as shown in Figure 3. Reference [25] calculates the vertex radius \(r\) using the following formula: 
\[
    r = \sqrt{2} \times \frac{d}{f}
\]
where \(d\) represents the depth value of the vertex and \(f\) represents the focal length of the depth camera.

#### 3.2 Camera tracking

Usually, the six-degree of freedom camera pose of a captured depth image is represented by a rigid transformation matrix \(T_c\). This transformation matrix maps the camera coordinate system to the world coordinate system. Given the position of the point \(p_l\) in the local camera coordinate system, it can be converted to its 3D coordinate in the world coordinate system, denoted by \(p_w\), by the following equation: 
\[
    p_w = T_c \times p_l.
\]

Camera tracking, i.e., estimating the camera pose, is a prerequisite for fusing the depth image of the
current frame into the global model, and the fused model is represented by either volumetric representation or a surfel dependent on the choice of each 3D reconstruction system (Figure 4). The iterative closest point algorithm (ICP) might be the most important algorithm in relative camera pose estimation. In the early days, the ICP algorithm was applied for 3D shape registration\cite{26,27}. The rigid transformation is calculated by determining matching points between two adjacent frames and minimizing the sum of squares of the Euclidean distance between these pairs. This step is iterated until a certain convergence criterion is satisfied. A serious problem with this frame-to-frame strategy is that the registration error of two frames is accumulated as the scan progresses.

To alleviate this problem, the depth camera-based reconstruction methods adopted a frame-to-model camera tracking method\cite{6,8,16}. Although frame-to-model tracking significantly reduces the tracking drift of each frame, it does not completely solve the problem of error accumulation, as tracking errors still accumulate over time. The accumulated drift eventually leads to misalignment of the reconstructed surface at the loop closure of the track. Therefore, researchers have introduced the idea of global pose optimization. Henry et al.\cite{29} proposed the concept of the keyframe, which produces a keyframe whenever the accumulated camera pose is greater than a threshold. The loop closure detection is performed only when a keyframe occurs; moreover, the random sample consensus algorithm\cite{30} is used to perform feature point matching on the current and previous keyframes. After the loop closure is detected, there are two globally optimized strategies to choose from. One is the pose graph optimization, where the pose graph is used to represent the constraint between the frames; further, the edge between the frames corresponds to the geometric constraint. The second is sparse bundle adjustment, which globally minimizes the re-projection error of the feature points using sparse bundle adjustment matching on all frames\cite{31-33}. Zhou et al.\cite{34} proposed the use of interest points to preserve local details and combine global pose optimization to evenly distribute registration errors in the scene. Zhou et al.\cite{35} also proposed the segmentation of a video sequence into multiple segments using frame-to-model integration, followed by the reconstruction of locally accurate scene segments from each segment, the establishment of a dense correspondence between overlapping segments, and the optimization of a global energy function to align the fragment. The optimized images can subtly distort the image fragments to correct inconsistencies caused by low-frequency distortion in the input images. Whelan et al.\cite{36} used depth and color information to perform global pose estimation for each input frame. The prediction of the depth and color images used for registration is obtained by using the surface splatting algorithm\cite{19} for the global model. They divided the loop closure problem into global and partial loop closures. Global loop closure detection uses the random fern coding method\cite{37}. For loop closure optimization, they used a spatial deformation method based on the embedded deformation map\cite{38}. A deformation map consists of a set of nodes and edges that are distributed among the models to be deformed. Each surfel is affected by a set of nodes; moreover, the influence weight is inversely proportional to the distance of the surfel to the node.

Regardless of the registration strategy, identifying corresponding points between two frames is an
essential step in camera tracking. The set of corresponding point pairs is brought into the objective function to be optimized, thereby calculating the optimal solution of the camera pose. The process of determining the corresponding points can be divided into sparse and dense methods according to the pair of points used. The sparse method only obtains the corresponding feature points, while the dense method searches for the correspondences for all points.

For feature extraction and matching, the scale-invariant feature transform (SIFT) algorithm\(^\text{[39]}\) is a popular choice. The speeded-up robust features (SURF) algorithm\(^\text{[40]}\) improves the SIFT algorithm and the speed of detecting feature points. The ORB feature descriptor is an alternative to SIFT and SURF, which is faster and correspondingly less stable. Endres et al.\(^\text{[41]}\) exploited these three feature descriptors when extracting feature points and compared their effects. Whelan et al.\(^\text{[8]}\) explored feature point-based fast odometry from vision instead of the ICP method in KinectFusion and adopted SURF feature descriptors for loop closure detection. Zhou et al.\(^\text{[42]}\) proposed a method to track camera based on object contours and introduced corresponding points of contour features to ensure more stable tracking. The recent BundleFusion method\(^\text{[43]}\) used the SIFT feature for coarse registration first and subsequently dense methods for fine registration.

For dense methods, the traditional methods of identifying corresponding points are time consuming. The projection data association algorithm\(^\text{[44]}\) significantly speeds up this process. This strategy projects the three-dimensional coordinates of the input point to a pixel on the target depth map according to the camera pose and the internal parameters of the camera; then, it obtains the best corresponding point through the search in the neighborhood of the pixel. There are many ways to measure the error of the corresponding point, such as point-to-point metric\(^\text{[26]}\) and point-to-plane metric\(^\text{[27]}\). When compared to the point-to-point metric, the point-to-plane metric results in faster convergence in the camera tracking; thus, it is widely used. In addition to using geometric distance as constraints, many methods use optical differences (i.e., color differences) between corresponding points as constraints, such as\(^\text{[43–45]}\). Lefloch et al.\(^\text{[46]}\) considered curvature information as an independent surface property for real-time reconstruction. They considered the curvature of the input depth map and global model when determining dense corresponding points.

### 3.3 Fusion of depth images

**Volumetric representation:** When compared to computing a complete SDF for all voxels, the TSDF is usually calculated because its computational cost is more affordable. The voxel center is first projected to the input depth map after camera tracking to compute the TSDF value corresponding to the current input depth map. Then, the new TSDF value is fused to the TSDF of the global model by weighted averaging, which is effective in removing depth noises.

**Surfel:** After estimating the camera pose of the current input frame, each vertex and its normal and radius are integrated into the global model. The fusion of depth maps using surfels has three steps: data association, weighted average, and removal of invalid points\(^\text{[20]}\). First, corresponding points are determined by projecting the vertices of the current 3D model to the image plane of the current camera. As some model points might be projected onto the same pixel, the upsampling method is used to improve the accuracy. If the corresponding point is identified, the most reliable point is merged with the new point estimate using a weighted average. If it is not determined, the new point estimate is added to the global model as an unstable point. Over time, the global model is cleaned up to eliminate outliers according to the visibility and time constraints.
3.4 3D reconstruction of dynamic scene

The reconstruction of dynamic scenes is more challenging than static scenes because of deforming objects. We have to deal with not only the overall movement of the objects but also their non-rigid deformations. The key issue to dynamic reconstruction is to develop a fast and robust non-rigid registration algorithm to handle non-rigid motions in the scene.

**Volumetric representation:** DynamicFusion\(^{49}\) proposed by Newcombe et al. was the first method to reconstruct a non-rigid deformation scene in real time using the input obtained from a depth camera. It reconstructs an implicit volumetric representation, similar to the KinectFusion method, and simultaneously optimizes the rigid and non-rigid motion of the scene based on the warped field parameterized by the sparse deformation map\(^{50}\). The main problem with DynamicFusion is that it cannot handle topology changes that have occurred during the capturing. Dou et al. proposed the Fusion4D method\(^{49}\). This method used eight depth cameras to acquire a multi-view input, combined the concept of volumetric fusion with the estimation of the smooth deformation field, and proposed a fusion method of multiple camera depth information that can be applied to complex scenes. In addition, it proposed a machine-learning-based corresponding points detection method that can robustly handle frame-to-frame fast motion. Recently, Yu et al. proposed a new real-time reconstruction system, called DoubleFusion\(^{49}\). This system combined dynamic reconstruction based on volumetric representation and skeletal model SMPL of the reference [50], which can reconstruct detailed geometry, non-rigid body motion, and internal shape simultaneously from a single depth camera. One of the key contributions of this method is the two-layer representation, which consists of a fully parametric internal shape and a gradually fused outer surface layer. The predefined node map on the body surface parameterizes the non-rigid body deformation near the body; moreover, the free-form dynamic change map parameterizes the outer surface layer away from the body for more general reconstruction. Furthermore, a joint motion tracking method based on the two-layer representation was proposed to make joint motion tracking robust and fast. Guo et al.\(^{51}\) proposed a shading-based scheme to leverage appearance information for motion estimation and the computed albedo information was fused into the volume. The accuracy of the reconstructed motion of the dynamic object was significantly improved owing to the incorporation of the albedo information.

**Surfel:** Keller et al.\(^{20}\) used the foreground segmentation method to process dynamic scenes. The proposed algorithm removes objects with moving surfaces from the global model so that the dynamic part does not act on the camera tracking. As shown in Figure 5, a person seated on a chair is initially reconstructed and then he begins to move. The system segments the moving person in the ICP step by ignoring the dynamic scene of part (A) (refer Figure 5); thus, it ensures the robustness of the camera tracking to dynamic motion.

Rünz et al.\(^{52}\) proposed to use motion or semantic cues to segment the scene into backgrounds and different rigid foreground objects, simultaneously tracking and reconstructing their 3D geometry over time, which is one of the earliest real-time methods for intensive tracking and fusion of multiple objects. They proposed two segmentation strategies: motion segmentation (grouping points that are consistent in motion in 3D) and object instance segmentation (detect and segment single objects in RGB images for a given semantic tag). Once detected and segmented, objects are added to the active model list, then tracked, and their 3D shape models are updated by merging the data that belongs to the object. Gao et al.\(^{53}\) proposed the utilization of a deformation field to align the depth input of the current moment with a reference geometry (template). The reference geometry is initialized with the deformation field of the previous frame and then optimized by the ICP algorithm, similar to DynamicFusion. Once the deformation field is updated, data fusion is performed to accumulate the current depth observations into the global geometric model.
According to the updated global model, the global deformation field is inversely computed for the registration of the next frame.

### 3.5 Scene understanding

Scene understanding refers to the process of analyzing the scene by considering the geometry and semantic context of the scene content and the intrinsic relationship between them, including object detection and recognition, the relationship between objects, and semantic segmentation. Researches in recent years have shown that the understanding of scenes, such as plane detection and semantic segmentation, can provide significant assistance for the 3D modeling of scenes.

Observing that there are many planes in the indoor scene, such as walls, floors, and desktops, Zhang et al. proposed a method of region growth to detect and label new planes using planarity and orthogonality constraints to reduce the camera tracking drift and flat region bending. Dzitsiuk et al. fit the plane to the symbol distance function value of the volume based on the robust least squares method and merged the detected local planes to determine a global continuous plane. By using the detected plane to correct the SDF value, the noise on the flat surface can be significantly reduced. Further, they used an a priori plane to directly obtain object-level and semantic segmentation (such as walls) of the indoor environment. Shi et al. proposed a method for predicting the co-planarity of image patches for RGB-D image registration. They used a self-supervised approach to train a deep network to predict whether two image patches are coplanar or not and to incorporate coplanar constraints when solving camera pose problems. Experiments demonstrated that this method of using co-planarity is still effective when feature point-based methods cannot achieve loop closure.

SLAM++ proposed by Salas-Moreno et al. is one of the earliest methods for object-oriented mapping methods. They used point-to-point features to detect objects and reconstruct 3D maps at the object level. Although this scheme enables the SLAM system to be extended to object-level loop closure processing, a
detailed 3D model preprocessing of objects is required in advance to learn the 3D feature descriptors. Tateno et al.\(^{(59)}\) incrementally segmented the reconstructed region in TSDF volumes and used the oriented, unique, and repeatable clustered viewpoint feature histogram\(^{(60)}\) feature descriptor to directly match other objects in the database. Xu et al.\(^{(61)}\) proposed a recursive 3D attention model for the active 3D object recognition by a robot. The model can perform both the instance-level shape classification and the continuous next best view regression. They inserted the retrieved 3D model into the scene and built the 3D scene model step by step. In reference [63], the 3D indoor scene scanning task was decomposed to concurrent subtasks for multiple robots to explore the unknown regions simultaneously, where the target regions of subtasks were computed via optimal transportation optimization.

In recent years, with the development of 2D semantic image segmentation algorithms, many methods combine them with the 3D modeling methods to improve the reconstruction quality. McCormac et al.\(^{(64)}\) adopted the convolutional neural network (CNN) model\(^{(65)}\) proposed by Noh et al. to perform 2D semantic segmentation. The Bayesian framework for 2D-3D label transfer was used to fuse 2D semantic segmentation labels into 3D models. Then, the model was post-processed using convolutional random field. Rünz et al.\(^{(54)}\) adopted a mask region-based CNN model\(^{(66)}\) to detect object instances, as can be seen in Figure 6. They applied ElasticFusion\(^{(36)}\) using surfels to represent each object and static background for real-time tracking and the dense reconstruction of dynamic instances. Differing from the above two methods for semantic labeling of the scene at the object level, Hu et al.\(^{(67)}\) proposed a method for reconstructing a single object based on the labeling of each part of the object. To minimize the labeling work of the user, they used an active self-learning method to generate the data required by the neural network. Schönberger et al.\(^{(68)}\) obtained 3D descriptors by jointly encoding high-level 3D geometric information and semantic information. Experiments have shown that such descriptors enable reliable positioning even under extreme viewing angles, illumination, and geometric variations.

![Figure 6](image_url)  
**Figure 6** Recognition, tracking, and mapping results in MaskFusion\(^{(54)}\).

4 **Texture reconstruction**

In addition to the 3D geometry of objects and scenes, which are of great interest in many applications, surface color and general appearance information can significantly enhance the sense of reality of 3D geometry in augmented reality/virtual reality applications, which is important to simulate the real world in virtual environments. In the following sections, offline and online texture reconstruction methods are reviewed and discussed.
4.1 Offline texture reconstruction

The goal of these techniques is to reconstruct the high-quality, globally consistent texture for the 3D model with the multi-view RGB images after the capturing process is completed. They rely on optimization algorithms to remove the ghosting and over-smoothing effects in the direct blending of the captured images. The frequently explored optimization objectives include color consistency\(^6\),\(^7\), the alignment of image and geometric features\(^7\),\(^8\), and mutual information maximization between the projected images\(^9\),\(^10\).

While these methods can effectively handle camera calibration inaccuracies, they cannot handle inaccurate geometries and optical distortions in RGB images, which is a common problem in the data captured by consumer-level depth cameras. Zhou and Koltun\(^11\) solved the above problems by optimizing for the RGB camera poses and non-rigid warping of each associated image to maximize the photo consistency. In contrast, Bi et al.\(^12\) proposed a patch-based optimization method, which synthesizes a set of feature-aligned color images to produce high-quality texture mapping even with large geometric errors. Huang et al.\(^13\) proposed several improvements based on the method presented by Zhou et al.\(^11\) for 3D indoor scene reconstruction (refer Figure 7). After calculating a primitive abstraction of the scene, they first corrected color values by compensating for the varying exposure and white balance; then, the RGB images were aligned by optimization based on sparse features and dense photo consistency. Finally, a temporally coherent sharpening algorithm was developed to obtain clean texture for the reconstructed 3D model.

![Figure 7: Primitive abstraction and scene completion\(^13\).](image)

4.2 Online texture reconstruction

Online texture reconstruction is used to generate high-quality textures in parallel to the scanning of the 3D-shaped objects. Its advantage is that the user can observe the real-time textured model during online scanning; however, it may demonstrate inferior texture quality as it cannot exploit the unseen images. Whelan et al.\(^14\) refused to fuse the color values of the edges of the object into the model as this will result in artifacts or discontinuities. In the subsequent work\(^15\), they estimated the position and orientation of the light source in the scene to further reject the fusion from samples containing only highlights. Although this has significantly improved the visual quality of the texture, artifacts may still occur. As high dynamic range (HDR) images provide more image details than low dynamic range images, they are also adopted in the 3D reconstruction based on RGB-D cameras\(^16\),\(^17\). First, the pre-calibrated response curve is used to linearize the observed intensity values. Then, the HDR pixel colors based on the exposure time are computed and fused into the texture of the 3D model in real time. A recent contribution by Xu et al. proposed a dynamic atlas texturing scheme for warping and updating dynamic texture on the fused geometry in real time, which significantly improved the visual quality of 3D human modeling\(^18\).
5 Conclusions and future work

In the past few years, 3D modeling using depth cameras has evolved rapidly. Its current development includes the entire reconstruction pipeline and brings together various levels of innovation from deep camera hardware to high-level applications, such as augmented reality and teleconference. Research works on 3D modeling using depth cameras, including camera tracking, 3D object and scene reconstruction, and high-quality texture reconstruction, are reviewed and discussed in this paper.

Despite the achieved success, 3D modeling based on depth cameras still encounters many challenges. First, the depth cameras are actually short-range 3D sensors. It is still time consuming to apply them in the large-scale scene reconstructions. Thus, it is necessary to investigate how to integrate the data of long-range sensors, such as laser scanner, with depth data to expedite the 3D scene reconstruction. A multi-source large-scale 3D data registration algorithm, including correspondence detection and subsequent distance minimization, should be developed in this case. We hypothesize that domain decomposition technique, which supports the alternative minimization of the objective function using a subset of optimization variables in each iteration, can be applied to handle such large-scale registration problems. It was also interesting to investigate the active vision approach to drive the robots with depth cameras in a scene for a complete scanning\cite{49}. Second, the moving objects in the scene are not friendly to the depth camera tracking. We need to explore how to accurately separate the moving objects from the static scene in the captured depth to improve the accuracy of the camera tracking-based static scene. One possible solution is to integrate the object detection and motion flow together, where the movement of the objects conforming to the camera motion should be static with respect to the camera. Third, to support the application of depth cameras integrated into mobile phones, it is necessary to develop real-time and energy-saving depth data processing and camera tracking algorithms for the hardware of mobile phone. The convergence acceleration algorithms, such as Anderson acceleration, can be explored to speed-up the camera tracking\cite{84}.

References

 DOI:10.1109/cvpr.2006.19

 DOI:10.1109/iccv.2003.1238654

 DOI:10.1109/ismar.2007.4538852

 DOI:10.1145/566654.566600

 DOI:10.1145/2047196.2047270

DOI: 10.1109/ismar.2011.6092378

DOI: 10.1145/237170.237269


DOI: 10.5244/c.26.112

DOI: 10.1145/344779.344899

DOI: 10.1145/1360612.1360634

DOI: 10.1109/tvcg.2010.75

DOI: 10.1016/j.gmod.2012.09.002

DOI: 10.1145/2461912.2461940

DOI: 10.1109/icra.2014.6907

DOI: 10.1145/2508363.2508374

DOI: 10.1109/tvcg.2015.2459891

DOI: 10.1007/s10458-017-9624-2

DOI: 10.1145/344779.344936

DOI: 10.1109/3dv.2013.9

DOI: 10.1109/mfi.2012.6343050


DOI:10.1145/1276377.1276478

DOI:10.1016/j.cviu.2007.09.014

DOI:10.1109/icra.2012.6225199

DOI:10.1109/cvpr.2015.7298662

DOI:10.1145/3072959.3073596

DOI:10.1109/34.400574

DOI:10.1109/iccv.2013.405

DOI:10.1109/tpami.2017.2648803

DOI:10.1109/cvpr.2015.7298631

DOI:10.1145/2984511.2984517

DOI:10.1109/cvpr.2018.00761

DOI:10.1145/2816795.2818013

DOI:10.1145/3072959.3126786

DOI:10.1109/icra.2017.7989518


DOI:10.1109/2945.965346

70 Pulli K, Shapiro L G. Surface reconstruction and display from range and color data. Graphical Models, 2000, 62(3): 165–201
DOI:10.1006/gmod.1999.0519

DOI:10.1006/gmod.2001.0554

DOI:10.1006/cviu.2002.0963

DOI:10.1111/j.1467-8659.2009.01552.x

DOI:10.1007/s11263-012-0552-5

DOI:10.1145/2601097.2601134

DOI:10.1177/0278364914551008

DOI:10.1145/3072959.3073610

DOI:10.1177/0278364916669237

DOI:10.1109/ismar.2013.6671774

DOI:10.1109/3dv.2016.40

81 Alexandrov S V, Prankl J, Zillich M, Vincze M. Towards dense SLAM with high dynamic range colors. In Computer Vision Winter Workshop (CVWW), 2017

DOI:10.1109/tpami.2019.2915229

DOI:10.1145/3130800.3130812

DOI:10.1145/3197517.3201290

499