Pacific Graphics 2017
J. Barbic, W.-C. Lin, and O. Sorkine-Hornung
(Guest Editors)

Volume 36 (2017), Number 7

Saliency-aware Real-time Volumetric Fusion for Object Reconstruction

Sheng Yang[1], Kang Chen[1], Minghua Liu[1], Hongbo Fu[2] and Shi-Min Hu[1]

[1]Tsinghua University
[2]City University of Hong Kong

Abstract

We present a real-time approach for acquiring 3D objects with high fidelity using hand-held consumer-level RGB-D scanning devices. Existing real-time reconstruction methods typically do not take the point of interest into account, and thus might fail to produce clean reconstruction results of desired objects due to distracting objects or backgrounds. In addition, any changes in background during scanning, which can often occur in real scenarios, can easily break up the whole reconstruction process. To address these issues, we incorporate visual saliency into a traditional real-time volumetric fusion pipeline. Salient regions detected from RGB-D frames suggest user-intended objects, and by understanding user intentions our approach can put more emphasis on important targets, and meanwhile, eliminate disturbance of non-important objects. Experimental results on real-world scans demonstrate that our system is capable of effectively acquiring geometric information of salient objects in cluttered real-world scenes, even if the backgrounds are changing.

CCS Concepts

•***Computing methodologies*** → *Reconstruction; Object detection;*

1. Introduction

As a milestone in real-time depth fusion, *KinectFusion* [IKH*11] has aroused great research interests among the vision and graphics communities. Various works have been proposed to improve *KinectFusion* in different aspects [RV12, WKF*12, NZIS13, KPR*15]. However, robust and high-precision real-time reconstruction of real-world scenes with consumer-level depth sensors is still a hard problem. The difficulties come from two-folds: first, depth information acquired by cheap sensors (e.g., *Microsoft Kinect*) is noisy, distorted and incomplete; second, real-world environments are typically complex and cluttered, where scanner-unfriendly surface materials, featureless areas and dynamic objects frequently occur. Even the state-of-art fusion techniques [KPR*15, DNZ*16] cannot easily produce clean, complete and accurate results. Actually, reconstruction qualities are affected by many factors, including the conditions of target scenes, strategies adopted by fusion algorithms, parameter configurations, experiences of users, etc. Thus, different application scenarios favor different kinds of reconstruction solutions. For instance, placing target objects on a rotating platform under fixed camera(s) would always be an ideal condition, if possible.

In this work, we aim to improve real-time reconstruction performances in a specific scenario, i.e., reconstructing objects in *complex real-world backgrounds* with a hand-held *Kinect*-style RGB-D sensor. Although *KinectFusion* and its follow-up works already have the ability to solve this problem to some extent, they do not distinguish between the background environment and the target objects therein. As a consequence, sensor noise and depth distortion will cause equal influences to each voxel, which is reasonable when reconstructing the whole scene since each RGB-D pixel is of equal importance. However, in the scenario of object reconstruction, registration precision of regions belonging to target objects is clearly much more important. With the presence of noise and distortion, such differences between object and scene reconstruction would result in very clear changes. We kindly refer readers to [CZMK16] to find the state-of-the-art object reconstruction techniques, which however do not always produce satisfactory reconstruction results. In addition, existing methods are sensitive to dynamic background objects, which often occur in real-world scenes, e.g., walking pedestrians or windblown curtains. Any changes in background during scanning, may easily break up the registration process, which leads to a strict restriction upon existing reconstruction systems.

We observed that if a reconstruction framework is smart enough to understand user intentions, the aforementioned issues can all be addressed. We can thus put more emphasis on important targets during registration, and meanwhile, eliminate disturbance of non-important objects, even if they are moving in the background. The idea of protecting local geometry around points of interest in scans was first introduced and successfully used in an off-line dense reconstruction pipeline [ZK13]. However, their global cam-

era tracking algorithm for POI (point of interest) cannot be adapted to meet our real-time requirements. Thus, we resort to inferring user-intended objects by detecting visual saliences from RGB-D frames, which are then incorporated into a traditional real-time volumetric fusion pipeline. Experimental results on real-world scans demonstrate that our system is capable to effectively acquire detailed geometric information of salient objects in cluttered real-world scenes, and outperforms the status quo.

In summary, we present a real-time framework for acquiring 3D objects from complex real-world environments using hand-held consumer-level RGB-D scanning devices. Compared with existing works, our framework has several advantages: (i) geometric features of target objects are better preserved; (ii) users can conveniently tap to change target objects if multiple salient objects are present; (iii) tracking and registration will not be disturbed by changes in the background. To achieve them, we present a novel spatial-temporal visual saliency detection method and successfully incorporate visual saliency into real-time depth fusion.

2. Related Work

Our framework involves two key components: saliency detection and volumetric fusion. Thus, in this section, we give a brief review of the most related works in these two areas.

Saliency Detection. Visual saliency has long been a fundamental problem in neuroscience, psychology, and vision perception. Although there have not yet been consensus on how visual saliency should be defined and evaluated in the computer vision community, various saliency models have been proposed for different applications, for example, predicting human fixation or extracting salient regions [BCJL15]. In our framework, visual saliency is adopted as cues for identifying POI of users, and shares similar definition to the saliency recognized in image and video segmentation works [AEWS08, FXL17].

Classical saliency detection methods typically extract features at pixel level [CMH*15] or super-pixel level [JWY*13, QCB*15], which are then propagated into spatially-coherent meaningful regions based on different kinds of saliency metrics [PLX*14, GRB16]. Image segmentation techniques like GrabCut [RKB04] are often used at the final stage to decide whether a region is salient or not [FPCC16]. For RGBD images, the depth channel is handled isolately [GRB16], or used for extracting additonal features for comparison [PLX*14]. Recent deep learning methods have also addressed the problem [KWW16, HCH*16] but relatively heavy (cost hundreds of milliseconds per frame) for real-time applications. Spatio-temporal coherence has also been addressed when determining salient regions in video sequences [FWLF14, KKSK15, ZLR*13], where optical flow is incorporated in these methods for motion estimation. However, their methods are not suitable for identifying foreground objects of interests or determining which one is more important when multiple foreground objects exist. In fact, such methods require high-quality RGB sensors and good lighting conditions, while extra depth information from sensors used in our scenario offers stronger cues about focus and motions of objects than traditional 2D flow estimation. To the best of our knowledge, none of the previous video saliency detection methods have coupled depth fusion to consider such coherence in 3D space.

In our framework, saliency detection serves a very specific purpose, i.e. real-time depth fusion, which makes our saliency detection component addressed different from those typically tackled in computer vision in three aspects: (i) input RGB-D streams are normally captured around a target which the user intends to reconstruct, and thus salient regions are much clearer and more recognizable (ii) salient objects in the scene are strictly static rigid bodies, (iii) salient model must be computed very quickly to achieve real-time depth fusion. Despite various saliency detection algorithms studied in computer vision, none of them suits our application scenario. Thus, we introduce a novel spatio-temporal visual saliency detection method by customizing a salient model for single-frame RGB-D images based on [PLX*14], and integrate it into the frame-to-model registration process in the traditional depth fusion workflow. Experimental results have demonstrated that the proposed reconstruction-orientated saliency model and frame-to-model registration are mutually complementary in terms of precision.

Volumetric Fusion. Volumetric fusion is the most classic 3D reconstruction method studied in computer graphics [CL96]. Traditional volumetric fusion pipelines involve two key phases: tracking and integration, where ICP (Iterative Closest Point) [BM92] and SDF (Signed Distance Function) techniques are typically used in each phase respectively. In 2011, Iazadi et al. introduced *KinectFusion* [IKH*11] which demonstrated that real-time volumetric fusion systems could be achieved with the parallel processing abilities of modern graphical hardwares. Ever since, real-time 3D reconstruction systems based-on low-end depth sensors have become a research hotspot [CLH16].

Many subsequent works aimed to improve *KinectFusion* from various aspects. *Moving Volume KinectFusion* [RV12] and *Kintinuous* [WKF*12] eliminated the graphical memory limitation by straightforwardly shifting volumes maintained in memory according to camera trajectories. This idea was further extended by deploying better data structures and exploiting smarter memory-swapping strategies [NZIS13, KPR*15], which successfully improved the efficiency of graphical memory usage and has become a prevalent choice in practical volumetric fusion systems. With improvements of RGB cameras equipped by low-end RGB-D scanners, visual correspondences estimated from color images have also been considered to improve local registration precision [KSC13, WKJ*15] or to reduce drifting by loop-closure detection [CZK15, DNZ*16]. Thanks to advances in graphical hardwares, these time-consuming mechanisms used to appear in non-real time SLAM systems can now be integrated into real-time applications. Our framework follows this trend, i.e. we incorporate visual saliency into the classic volumetric fusion pipeline. Specifically, our application poses two major demands which have rarely been addressed before: (i) preserving detailed geometric information for target objects; (ii) reducing disturbance from non-important backgrounds.

3. Methodology

Our framework consists of two interdependent modules. The saliency detection module (Sec. 3.1) estimates salient image regions based on features extracted from each RGB-D frame and spatio-temporal information maintained by the volumetric fusion module (Sec. 3.2), which iteratively registers and fuses each incom-

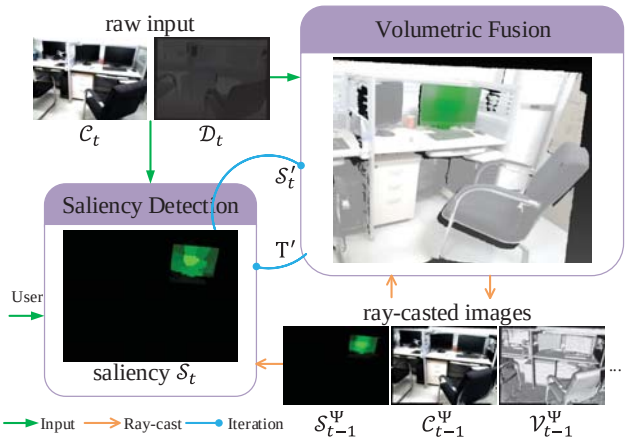


Figure 1: *Data flow in our framework.*

ing frame, with an aligned saliency map, into a global volumetric representation. Figure 1 illustrates the data flow in our framework. Before detailed discussions, we first give the mathematical notations as follows:

We define the space domain of an image as $\Omega \subset \mathbb{N}^2$ and each input frame at time t consists a depth map $\mathcal{D}_t : \Omega \to \mathbb{R}$ and an aligned color image $\mathcal{C}_t : \Omega \to \mathbb{N}^3$. For each frame, we aim to estimate its saliency map $\mathcal{S}_t : \Omega \to \mathbb{R}$ and camera pose $\mathbf{P}_t$, and fuse it into a maintained volumetric representation of reconstructed structures, which is defined as a set of voxels κ in a 3D space domain $\Psi \subset \mathbb{N}^3$. To incorporate saliency values into a volumetric fusion pipeline, we extend the traditional TSDF (truncated signed distance function) voxel stucture [IKH*11] with a float field storing its saliency $\kappa^S : \Psi \to \mathbb{R}$ (within [0,1]). The integration of saliency is carried out in a similar fashion to [IKH*11], where each voxel $v \in \Psi$ is updated with:

$$\kappa^S_t(v) = \frac{\kappa^S_{t-1}(v) * \kappa^W_{t-1}(v) + \mathcal{S}_t(p)}{\kappa^W_{t-1}(v) + 1}, \tag{1}$$

$$\kappa^W_t(v) = \min(\kappa^W_{t-1}(v) + 1, \sigma_M), \tag{2}$$

if v gets a corresponding pixel $p \in \Omega$ through ray-casting [IKH*11], we set $\sigma_M = 64$ and the voxel size to 5mm in our implementation and enable voxel hashing for unbounded reconstruction. Specifically besides a typical ray-casting, we generate ray-casted maps $\mathcal{S}^{\Psi}_{t-1}$ and $\mathcal{W}^{\Psi}_{t-1}$ as a saliency and a weight map, respectively. As for other notes, the induced vertex map from $\mathcal{D}_t$ is denoted as $\mathcal{V}_t$, and the ray-casted vertex map is denoted as $\mathcal{V}^{\Psi}_{t-1}$.

3.1. Saliency Detection

To implement a spatio-temporal video saliency with possible user interactions based on a previous image saliency method [PLX*14], we additionally take saliency results from previous frames and add interaction related contrastive features and propagation strategies. In general, we adopt a bottom-up two-level saliency extractor considering current frame, previous frames, camera trajectory, and potential user interactions. Specifically, we compute saliency values for each super-pixel based on multi-contextual contrast (local, global, pseudo-background, and the optional focus contrast), which are later propagated into larger salient image regions. The whole procedure is parallelized and accelerated by GPU to ensure real-time performance.

Feature Extraction. For each incoming pair of RGB and depth frame at time t, we first extract super-pixels via a GPU-accelerated SLIC algorithm [ASS*12] named gSLICr [RPR15] to construct nearly 200 super-pixels on each frame. Afterwards, we calculate the feature vector $\mathbf{S}_u = [\mathbf{S}^c_u, \mathbf{S}^n_u, \mathbf{S}^l_u, \mathbf{S}^d_u, \mathbf{S}^r_u]^T$ for each super-pixel $u \subset \Omega$, where $\mathbf{S}^c_u$ represents the centroid, $\mathbf{S}^n_u$ the PCA normal of the points in the super-pixel, $\mathbf{S}^l_u$ the average color in CIE-Lab color space, $\mathbf{S}^d_u$ the average depth, and $\mathbf{S}^r_u$ the number of pixels in u. All distance and location units are given in meters, and the three channels of CIE-Lab color are normalized in $[0,1]$.

Low-level Saliency Value. At time t, our low-level saliency value $\mathbf{S}_{u_t}$ of each super-pixel u_t is defined as the multiplication of the following three terms:

$$\mathbf{S}^L_{u_t} = C(\mathbf{S}_{u_t}) \times R(u_t, \mathcal{S}^{\Psi}_{t-1}) \times U(\mathbf{S}_{u_t}), \tag{3}$$

where $C(\mathbf{S}_{u_t})$ is a weighted combination of local, global, and pseudo-background contrasts defined in [PLX*14]. In our implementation, this term was calculated with the same parameter settings suggested in [PLX*14].

$R(u_t, \mathcal{S}^{\Psi}_{t-1})$ is the *weighted* average saliency value from a subset of all pixel-wise correspondences as $\bigcup_{p^i_t \in u_t}(p^i_t, p^j_{t-1})$. In order to reduce the influence of dynamic regions or outliers, the weight $\mathbf{x}_i$ for each correspondence is calculated through their location as:

$$\mathbf{x}_i = \exp\left(-\frac{\left\|\mathcal{V}_t(p^i_t) - \mathcal{V}^{\Psi}_{t-1}(p^j_{t-1})\right\|^2}{\sigma^2_X}\right), \tag{4}$$

where $\sigma_X = 0.2$ in our implementation. In the frame-to-model registration spirit [IKH*11], here we use the saliency map $\mathcal{S}^{\Psi}_{t-1}$ ray-casted from fused volumetric data rather than directly adopt $\mathcal{S}_{t-1}$, as $\mathcal{S}^{\Psi}_{t-1}$ contains the saliency information of all previous frames in order to coherently focus on the target. For the first frame, $R(\cdot)$ is set to one.

$U(\mathbf{S}_{u_t})$ depicts the influence of user-specified POI hints. Specifically, we construct a focus contextual set which consists of super-pixels having its centroid $\mathbf{S}^c_{u_t}$ within the user-specified focal point $\mathbf{F}_c$ and radius $\mathbf{F}_r$, and then follow [PLX*14] to compute the feature contrast. If no specification has been given, our system will take the centroid of the super-pixels which get the highest saliency as $\mathbf{F}_c$, and the distance to the furthest pixels among those with saliency values higher than their average as $\mathbf{F}_r$ for subsequent inputs.

Propagation. To remove outliers and obtain more meaningful salient image regions, we need to group super-pixels into larger regions. To further reduce effects from unconcerned areas through user hints based on the strategy from [PLX*14], we similarly construct an undirected graph of super-pixels in $\mathcal{C}_t$ based on their adjacency in the image space, and to generate a set of salient seeds, each of which is then greedily propagated into a spanning tree. On account of speed, super-pixels with low-level saliency value $\mathbf{S}^L$ greater than the average are all considered as seeds by default and

the propagation of each seed is computed parallelly on GPU. However, if $\mathbf{F}_c$ and $\mathbf{F}_r$ have been specified, the following modifications would be adopted.

First, a super-pixel u with $\|\mathbf{S}_u^c - \mathbf{F}_c\| < \mathbf{F}_r$ and its empirical weighting $\mathbf{S}_u^M$ above their average will be chosen as seeds, which is calculated as:

$$\mathbf{S}_u^M = \exp\left(-\frac{\|\mathbf{S}_u^c - \mathbf{F}_c\|^2}{\mathbf{F}_r^2}\right) \times \mathbf{S}_u^L. \tag{5}$$

By multiplying this exponential term, we emphasize those seeds closer to the focal point for propagation, and also raise their saliency (see the comparison between $\mathbf{S}_u^L$ and $\mathbf{S}_u^M$ in Figure 2). Second, a

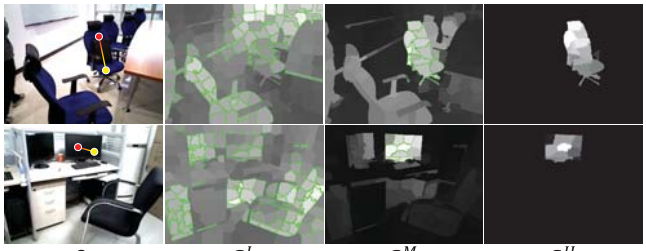


Figure 2: *Saliency at different levels. From left to right: user specification (red for focal point $\mathbf{F}_c$, spatial distance between red and yellow for radius $\mathbf{F}_r$), low-level, weighted low-level, and high-level saliency. Seeds are outlined with green borders. Since they are both the first frame of its video sequence, $\mathbf{S}^L$, $\mathbf{S}^M$ and $\mathbf{S}^H$ are all computed with $R(\cdot) = 1$, which means no influences from previous frames have been applied to them.*

new terminating condition is applied to the spanning tree algorithm, i.e. the edge $(\mathbf{S}_{u_i}, \mathbf{S}_{u_j})$ taking $\mathbf{S}_{u_o}$ as their seed will be added to the tree if

$$\left\|\mathbf{S}_{u_i}^c - \mathbf{S}_{u_o}^c\right\| < \mathbf{F}_r \cdot \mathbf{S}_{u_j}^L, \tag{6}$$

since the propagation through one seed should be limited in a certain extent with regard to the object size.

The final saliency value of $\mathbf{S}_u^H$ is then calculated as:

$$\mathbf{S}_u^H = Q(\mathbf{S}_u) \times \mathbf{S}_u^M, \tag{7}$$

where $Q(\cdot)$ is the normalized ($[0,1]$) frequency of $\mathbf{S}_u$ appeared in all the spanning trees. After propagation, the saliency map for current frame as $\mathcal{S}_t'$ is obtained and passed into the volumetric fusion step for tracking in the volumetric fusion module (Sec. 3.2).

Some results of our saliency detection module are illustrated in Figure 3, from which we can see that our method does produce more spatially and temporally consistent saliency maps compared with [PLX*14], and thus suits our application better. Please refer to our supplementary video for additional examples.

3.2. Volumetric Fusion

Compared with the previous fusion pipelines (e.g., [IKH*11] and [KPR*15]), our framework involves two additional channels in the tracking and integration process: the calculated saliency

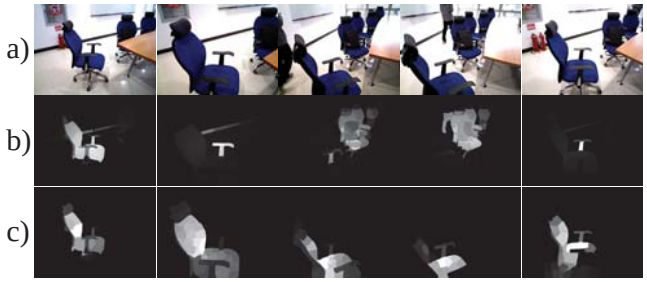


Figure 3: *Examples of coherent saliency results, each column of images are picked every 100 frames. a) input, b) [PLX*14], c) our results.*

map for current frame $\mathcal{S}_t$, and the ray-casted saliency map from the camera pose of last frame $\mathcal{S}_{t-1}'$. Specifically, in the tracking stage, both $\mathcal{S}_t'$ and the estimated pose $\mathbf{T}'$ are iteratively refined, and in the fusion stage, saliency information is integrated into volumes along with the geometric and color information.

Saliency-aware Pose Estimation. Our goal is to estimate the local transformation $\mathbf{T}$, which aligns the current frame to the previous one. In our implementation, $\mathbf{T}$ is obtained by iteratively solving the following minimization problem:

$$\arg\min_{\xi} E(\xi) = E_{wicp}(\xi) + \sigma_R E_{wrgb}(\xi), \tag{8}$$

where $\sigma_R = 0.1$ in our implementation. E_{wicp} and E_{wrgb} measure the saliency-aware geometric cost and the saliency-aware photometric costs, respectively. Following [WKJ*15], they are mainly defined based on the point-to-plane error in ICP registration E_{icp} and the intensity differences between RGB pixels E_{rgb}, respectively, which can both be formalized as least-squares problems after some derivations:

$$E_{icp} = \left\|\mathbf{J}_{icp}\xi + \mathbf{r}_{icp}\right\|^2, \tag{9}$$

$$E_{rgb} = \left\|\mathbf{J}_{rgb}\xi + \mathbf{r}_{rgb}\right\|^2. \tag{10}$$

We refer readers to [WKJ*15] for detailed definitions and derivations, here we focus on how saliency information is integrated into this cost function. Specifically, we define a saliency weight for each pixel-wise correspondence (p_t^i, p_{t-1}^j) as:

$$\mathbf{w}_i = \exp\left(\sigma_W \cdot \frac{\mathcal{S}_{t-1}^{\Psi}(p_{t-1}^j) \cdot \mathcal{W}_{t-1}^{\Psi}(p_{t-1}^j) + \mathcal{S}_t'(p_t^i)}{\mathcal{W}_{t-1}^{\Psi}(p_{t-1}^j) + 1}\right), \tag{11}$$

where $\sigma_W = 4$ in our implementation is the parameter adjusting the weight of salient regions through the tracking process. We follow the combination strategy in TSDF fusion to achieve a smooth tracking. Detailed evaluation of this parameter is given in Sec. 4.3.

We then organize all computed $\mathbf{w}_i$s of the current frame as a vector $\mathbf{W}$, based on which, the saliency-aware costs E_{wicp} and E_{wrgb} referred in equ 8 are finally defined as:

$$E_{wicp} = \left\|\mathbf{W} \cdot \mathbf{J}_{icp}\xi + \mathbf{W} \cdot \mathbf{r}_{icp}\right\|^2, \tag{12}$$

$$E_{wrgb} = \left\|\mathbf{W} \cdot \mathbf{J}_{rgb}\xi + \mathbf{W} \cdot \mathbf{r}_{rgb}\right\|^2. \tag{13}$$

After each iteration, the estimated camera pose is improved by ξ, which is mediately adopted to update the temporary saliency map $\mathcal{S}'_t$ to be referred in the next iteration. Following [WKJ*15], we also adopt a coarse-to-fine pyramid scheme when calculating equ 8, specifically, a 5×5 Gaussian kernel is applied to establish 3-level pyramids for each depth, color, and saliency maps. Such a weighting strategy encourages the fusion system to concentrate on the desired target and reduces effects from the distant background, which often presents more noise and distortion according to the distribution of sensor errors.

3.3. Interactive Focus Switching

Our system has further functionality allowing users to provide additional hints for focusing while reconstructing, since automatic inferred salient regions may be unexpected when multiple objects appear in front of the camera. In such a case, users can specify hints through a simple click-and-drag interaction, similar to the tap-to-focus operations in modern camera applications. A click on the input frame or ray-casted frame specifies a focal point $\mathbf{F}_c$ in the local or global coordinate. A drag from the focal point to a destination point tells a length $\mathbf{F}_r$ for the approximate size of the target. Focus switching on different objects can affect the tracking process and better capture geometric details of the target objects, as shown in Figure 4.

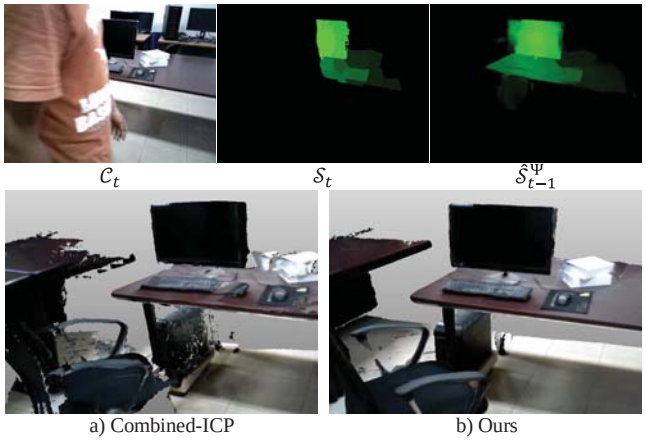
a) focus on cabinet b) focus on chair c) w/o saliency d) w/ saliency

Figure 4: *Examples of focus switching. Left: user specifications (top) and saliency (green) of reconstructed meshes (bottom). Right: quality of the target has been improved through focus switching.*

In many crowd or cluttered scenes, the target might be obstructed sometimes in the scanning process, e.g., due to a person walking across in front of the target (Figure 5). To reduce the impact of severe occlusion, we generated another ray-casted image as $\hat{\mathcal{S}}^{\Psi}_{t-1}$, whose rays only intersect with salient TSDF volumes. Comparisons between $\mathcal{S}_t$ and $\hat{\mathcal{S}}^{\Psi}_{t-1}$ enables our system to filter out the undesired regions before the tracking and fusion process.

4. Evaluation

We have evaluated our system in various aspects, including reconstruction quality, saliency coherence, and computational performance. Our testing data came from two sources: the large dataset of object scans constructed by Choi et al. [CZMK16] and a small dataset we built with 14 scenes scanned from 4 different places (i.e., a lobby, two seminar rooms, and an office) by an ASUS Xtion



a) Combined-ICP b) Ours

Figure 5: *An example of occlusion cases. Top: the color image and ray-casted images when the person was blocking the target. Bottom: final reconstructed models. Heavy occluded regions are filtered out through comparisons between $\mathcal{S}_t$ and $\hat{\mathcal{S}}^{\Psi}_{t-1}$ in our implementation.*

Pro Live sensor. The large dataset is mainly used for regression testing. The small dataset is established for evaluating focus switching and dynamic background objects during scanning, and thus either contain multiple targets or dynamic objects in the background (e.g., walking pedestrians, windblown curtains, etc.). We also made two 3D printed crafts with known geometry for quantitative evaluation.

4.1. Model Quality

Some representative results of both [CZMK16] and ours are shown in Figure 6, including various locations such as outdoors, office, and seminar room. We have compared our system with some of the state-of-art works, where KF, DV, CI, KT, E-F stand for *KinectFusion* [IKH*11], Dense RGB SLAM [SSC11], Combined-ICP [WJK*13], Kintinuous [WKJ*15], ElasticFusion [WLSM*15], respectively. Here we briefly classify these methods into three categories based on their cost functions used in pose estimation: KF only considers E_{icp}, DV only considers E_{rgb}, while CI, KT, and EF consider both E_{icp} and E_{rgb}. KT and EF are different from CI because they also involve the idea of loop-closure detection. All results are produced using the suggested default settings in their papers.

As illustrated in the top two rows of Figure 6, scans from [CZMK16] typically contain a single target in a static scene. Hence, all systems in our test produced similar results. However, once dynamic objects present in the background (Figure 6-sc1,sc2,sc3), the reconstruction processes of KF, DV, and CI are broken and severe drifting starts to appear. Although KT and EF can benefit from correctly detected loop closures, geometric details of their constructed target objects in large scenes are not well preserved with respect to ours (Figure 6-sc2,sc4). Generally, our method successfully protects local geometry by reducing registration errors of salient regions. Although misalignments may be amplified in other areas, such trade-off is reasonable and valuable in

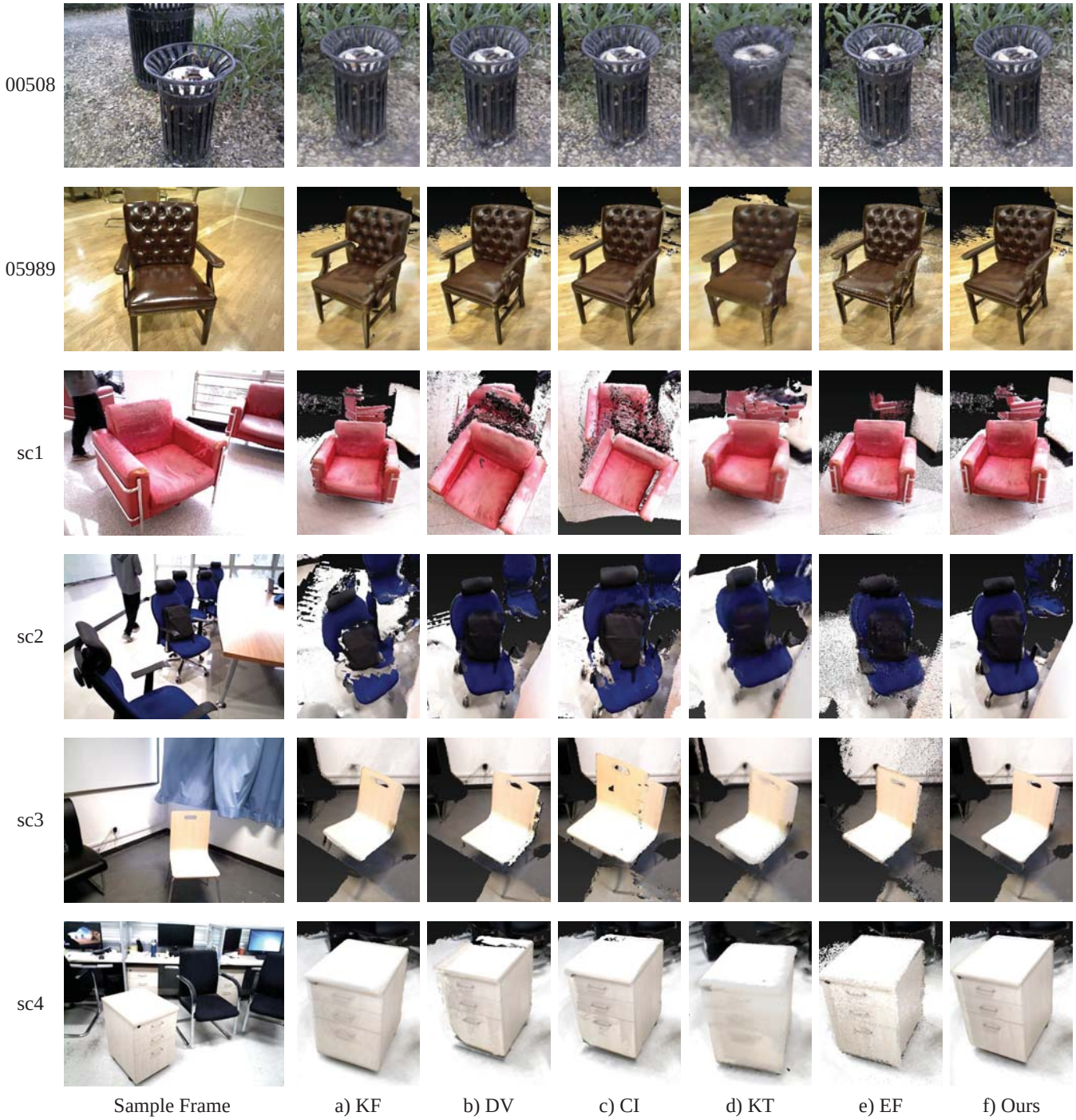


Figure 6: *Reconstruction of some test scenes. Top two rows from [CZMK16] with their ID on the left and the rest from our dataset. a) KinectFusion [IKH*11]. b) Dense RGB SLAM [SSC11]. c) Combined-ICP [WJK*13] d) Kintinuous [WKJ*15]. e) ElasticFusion [WLSM*15]. f) Our results.*

our application scenario. Since ground truth geometry of models in [CZMK16] are unknown, for quantitative evaluation, we scanned two 3D printed crafts shown in Figure 7, twice of each in both static and dynamic scenes separately. Both Combined-ICP [WJK*13] and our method were performed for each scan. The reconstructed results were manually segmented and registered to their ground truth mesh models, with their average point-to-model distances reported in Table 1. As a result, models reconstructed from our

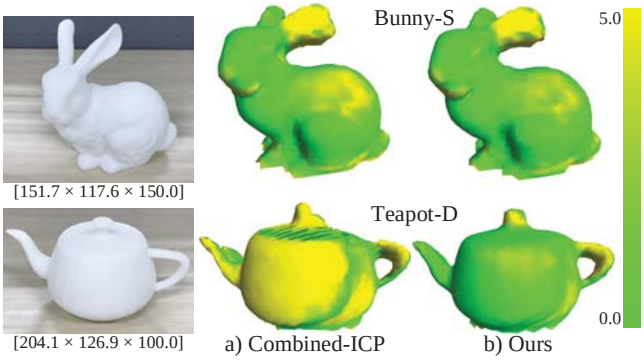


Figure 7: *Quantitative evaluation. Left: 3D printed models with known geometry for our experiment. Right: Visualization of the point-to-model distance for some reconstructed models. All given in millimeters.*

pipeline outperform those from Combined-ICP, especially in dynamic cases where the fusion process is influenced by dynamic backgrounds.

	Bunny-S	Teapot-S	Bunny-D	Teapot-D
Combined-ICP	2.81	2.11	3.41	3.99
Ours	**2.48**	**1.72**	**2.61**	**2.01**

Table 1: *Average point-to-model distance of test cases, while S for static cases and D for dynamic cases. All given in millimeters.*

4.2. Running-time

We deployed our system on a desktop with Intel Core i7 CPU, 32GB memory, and GeForce GTX 980Ti graphical card (2,816 CUDA cores). Average time costs for each step of our pipeline are reported in Table 2 with comparison to Combined-ICP. The saliency detection step includes super-pixel construction, feature extraction and two-level saliency calculation. Statistic results show that both the weighted tracking and the saliency integration bring insignificant burdens to our system. Finally, the overall speed meets the real-time requirement (less than 33ms per frame for 30Hz streams).

	Saliency	Tracking	Fusion	Total
Combined-ICP	-	11.23	1.40	12.63
Ours	12.87	12.09	1.50	26.46

Table 2: *Average computational performance of the test scenes. All timings are given in milliseconds.*

4.3. Parameter Study

The key parameter of our framework is the weight value σ_W in Equation 11, which controls the influence of saliency to volumetric fusion. Intuitively, high values of σ_W will significantly increase the importance of salient regions while decrease non-salient regions. If we set $\sigma_W = 0$, our framework is degenerated to Combined-ICP [WJK*13]. However, the quality of the reconstructed models is not monotonically increasing with σ_W, because the convergence speed of the Combined-ICP algorithm will be slowed down if the salient regions are planar surfaces and/or with featureless textures. In such cases, registration errors will be raised if we do not increase the number of iterations when solving Equation 8. In our experiments (see Figure 8), setting σ_W to around 4 generally produces plausible results for all testing data.

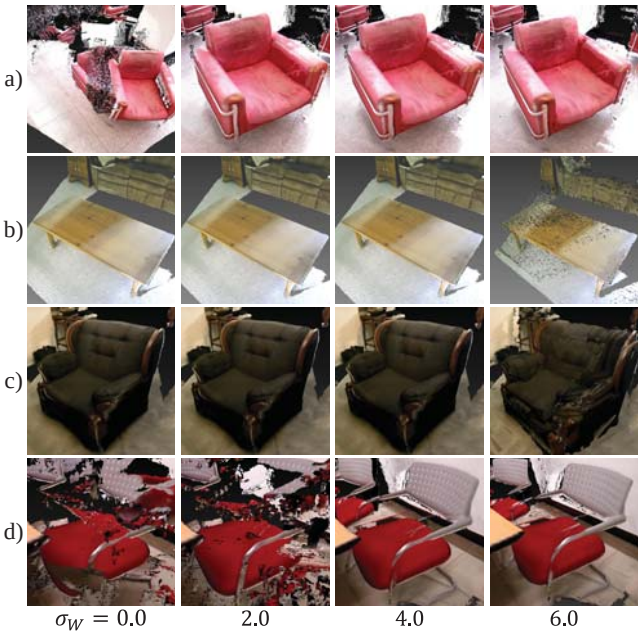


Figure 8: *Impacts of σ_W. a) and d) contains a walking pedestrian in the background. Also notice that σ_W is in an exponential term.*

4.4. Limitations

Our framework has two main limitations:(i) Our system is not suitable for full scene reconstruction applications. Since we put more emphasis on protecting local geometry of target objects, misalignments are forced into background areas. Thus, the quality of non-target objects in the reconstructed results is typically poorer compared with existing methods. (ii) Our system is not friendly to unpredictable users. Apparently, our system benefits from visual saliency detection to understand user intentions. However, if such intention were vague or obscure, our system might get confused and thus fail to produce high-fidelity results.

5. Conclusions and Future works

We have presented a real-time system for acquiring 3D objects with high fidelity using hand-held consumer-level RGB-D scanning devices. A novel spatio-temporal visual saliency detection method is incorporated into the traditional real-time volumetric

fusion pipeline, which successfully emphasizes the important targets and eliminates disturbance of non-important objects. We also present a simple user interface for focus changing. In the future, our system can be ported to mobile devices for free reconstruction, and the saliency information can be stored in the output mesh model to guide the object segmentation process.

Acknowledgements

We thank the reviewers for constructive comments. This work was supported by the Natural Science Foundation of China (Project Number 61521002) and Research Grant of Beijing Higher Institution Engineering Research Center. Hongbo Fu was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 113513).

References

[AEWS08] ACHANTA R., ESTRADA F., WILS P., SÜSSTRUNK S.: Salient region detection and segmentation. In *International Conference on Computer Vision Systems* (2008), pp. 66–75. 2

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SÜSSTRUNK S.: Slic superpixels compared to state-of-the-art superpixel methods. *IEEE TPAMI 34*, 11 (2012), 2274–2282. 3

[BCJL15] BORJI A., CHENG M.-M., JIANG H., LI J.: Salient object detection: A benchmark. *IEEE Transactions on Image Processing 24*, 12 (2015), 5706–5722. 2

[BM92] BESL P. J., MCKAY N. D.: A method for registration of 3-d shapes. *IEEE TPAMI 14*, 2 (1992), 239–256. 2

[CL96] CURLESS B., LEVOY M.: A volumetric method for building complex models from range images. In *SIGGRAPH* (1996), pp. 303–312. 2

[CLH16] CHEN K., LAI Y.-K., HU S.-M.: 3d indoor scene modeling from rgb-d data: a survey. *Computational Visual Media 1*, 4 (2016), 267–278. 2

[CMH*15] CHENG M.-M., MITRA N. J., HUANG X., TORR P. H., HU S.-M.: Global contrast based salient region detection. *IEEE TPAMI 37*, 3 (2015), 569–582. 2

[CZK15] CHOI S., ZHOU Q.-Y., KOLTUN V.: Robust reconstruction of indoor scenes. In *CVPR* (2015), pp. 5556–5565. 2

[CZMK16] CHOI S., ZHOU Q.-Y., MILLER S., KOLTUN V.: A large dataset of object scans. *arXiv preprint arXiv:1602.02481* (2016). 1, 5, 6

[DNZ*16] DAI A., NIESSNER M., ZOLLHÖFER M., IZADI S., THEOBALT C.: Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. *arXiv preprint arXiv:1604.01093* (2016). 1, 2

[FPCC16] FENG J., PRICE B., COHEN S., CHANG S.-F.: Interactive segmentation on rgbd images via cue selection. In *CVPR* (2016), pp. 156–164. 2

[FWLF14] FANG Y., WANG Z., LIN W., FANG Z.: Video saliency incorporating spatiotemporal cues and uncertainty weighting. *IEEE Transactions on Image Processing 23*, 9 (2014), 3910–3921. 2

[FXL17] FU H., XU D., LIN S.: Object-based multiple foreground segmentation in rgbd video. *IEEE Transactions on Image Processing 26*, 3 (2017), 1418–1427. 2

[GRB16] GUO J., REN T., BEI J.: Salient object detection for rgb-d image via saliency evolution. In *IEEE International Conference on Multimedia and Expo* (2016), pp. 1–6. 2

[HCH*16] HOU Q., CHENG M.-M., HU X.-W., BORJI A., TU Z., TORR P.: Deeply supervised salient object detection with short connections. *arXiv preprint arXiv:1611.04849* (2016). 2

[IKH*11] IZADI S., KIM D., HILLIGES O., MOLYNEAUX D., NEWCOMBE R., KOHLI P., SHOTTON J., HODGES S., FREEMAN D., DAVISON A., FITZGIBBON A.: Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *UIST* (2011), pp. 559–568. 1, 2, 3, 4, 5, 6

[JWY*13] JIANG H., WANG J., YUAN Z., WU Y., ZHENG N., LI S.: Salient object detection: A discriminative regional feature integration approach. In *CVPR* (2013), pp. 2083–2090. 2

[KKSK15] KIM H., KIM Y., SIM J.-Y., KIM C.-S.: Spatiotemporal saliency detection for video sequences based on random walk with restart. *IEEE Transactions on Image Processing 24*, 8 (2015), 2552–2564. 2

[KPR*15] KÄHLER O., PRISACARIU V. A., REN C. Y., SUN X., TORR P., MURRAY D.: Very high frame rate volumetric integration of depth images on mobile devices. *IEEE TVCG 21*, 11 (2015), 1241–1250. 1, 2, 4

[KSC13] KERL C., STURM J., CREMERS D.: Robust odometry estimation for rgb-d cameras. In *IEEE ICRA* (2013), pp. 3748–3754. 2

[KWW16] KUEN J., WANG Z., WANG G.: Recurrent attentional networks for saliency detection. In *CVPR* (2016), pp. 3668–3677. 2

[NZIS13] NIESSNER M., ZOLLHÖFER M., IZADI S., STAMMINGER M.: Real-time 3d reconstruction at scale using voxel hashing. *ACM TOG 32*, 6 (2013), 169. 1, 2

[PLX*14] PENG H., LI B., XIONG W., HU W., JI R.: Rgbd salient object detection: a benchmark and algorithms. In *ECCV* (2014), Springer, pp. 92–109. 2, 3, 4

[QCB*15] QI W., CHENG M.-M., BORJI A., LU H., BAI L.-F.: Saliencyrank: Two-stage manifold ranking for salient object detection. *Computational Visual Media 1*, 4 (2015), 309–320. 2

[RKB04] ROTHER C., KOLMOGOROV V., BLAKE A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM TOG* (2004), vol. 23, pp. 309–314. 2

[RPR15] REN C. Y., PRISACARIU V. A., REID I. D.: gslicr: Slic superpixels at over 250hz. *arXiv preprint arXiv:1509.04232* (2015). 3

[RV12] ROTH H., VONA M.: Moving volume kinectfusion. In *BMVC* (2012), vol. 20, pp. 1–11. 1, 2

[SSC11] STEINBRÜCKER F., STURM J., CREMERS D.: Real-time visual odometry from dense rgb-d images. In *IEEE ICCV Workshops* (2011), pp. 719–722. 5, 6

[WJK*13] WHELAN T., JOHANNSSON H., KAESS M., LEONARD J. J., MCDONALD J.: Robust real-time visual odometry for dense rgb-d mapping. In *IEEE ICRA* (2013), pp. 5724–5731. 5, 6, 7

[WKF*12] WHELAN T., KAESS M., FALLON M., JOHANNSSON H., LEONARD J., MCDONALD J.: Kintinuous: Spatially extended kinectfusion. In *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras* (2012). 1, 2

[WKJ*15] WHELAN T., KAESS M., JOHANNSSON H., FALLON M., LEONARD J. J., MCDONALD J.: Real-time large scale dense RGB-D SLAM with volumetric fusion. *International Journal of Robotics Research 34*, 4-5 (2015), 598–626. 2, 4, 5, 6

[WLSM*15] WHELAN T., LEUTENEGGER S., SALAS-MORENO R. F., GLOCKER B., DAVISON A. J.: Elasticfusion: Dense slam without a pose graph. In *Robotics: Science and Systems* (2015). 5, 6

[ZK13] ZHOU Q.-Y., KOLTUN V.: Dense scene reconstruction with points of interest. *ACM TOG 32*, 4 (2013), 112. 1

[ZLR*13] ZHONG S.-H., LIU Y., REN F., ZHANG J., REN T.: Video saliency detection via dynamic consistent spatio-temporal attention modelling. In *AAAI* (2013). 2