Joint Multiview Segmentation and Localization of RGB-D Images using Depth-Induced Silhouette Consistency

Chi Zhang[1,3*] Zhiwei Li[2] Rui Cai[2] Hongyang Chao[1,3] Yong Rui[2]

[1]Sun Yat-Sen University [2]Microsoft Research
[3]SYSU-CMU Shunde International Joint Research Institute, P.R. China

Abstract

In this paper, we propose an RGB-D camera localization approach which takes an effective geometry constraint, i.e. silhouette consistency, into consideration. Unlike existing approaches which usually assume the silhouettes are provided, we consider more practical scenarios and generate the silhouettes for multiple views on the fly. To obtain a set of accurate silhouettes, precise camera poses are required to propagate segmentation cues across views. To perform better localization, accurate silhouettes are needed to constrain camera poses. Therefore the two problems are intertwined with each other and require a joint treatment. Facilitated by the available depth, we introduce a simple but effective silhouette consistency energy term that binds traditional appearance-based multiview segmentation cost and RGB-D frame-to-frame matching cost together. Optimization of the problem w.r.t. binary segmentation masks and camera poses naturally fits in the graph cut minimization framework and the Gauss-Newton non-linear least-squares method respectively. Experiments show that the proposed approach achieves state-of-the-arts performance on both tasks of image segmentation and camera localization.

1. Introduction

3D scanning for objects is an important technique in computer vision with many applications. With the popularity of consumer-level depth cameras, even untrained users are able to scan objects at home. However, obtaining accurate camera poses is a major challenge for existing scanning systems. Typical RGB-D camera tracking systems leverage on either frame-to-frame matching [14] or frame-to-model matching [19] to localize cameras. In both cases drift is a common problem. For frame-to model tracking system such as KinectFusion [19] where online depth images are constantly integrated into a truncated signed distance function (TSDF) based volumetric representation [6], even small errors of camera poses will make the TSDF model blurry and consequently fine details are lost.

Loop closure detection and pose graph optimization are effective tools to address the above problem. Additional features, such as colors [14], local features [23], and occluding contours [28] have been considered in literature. However, another important type of constraints, i.e. geometric constraints, is overlooked by these approaches. Examples of the geometric constraints include epipolar tangency criterion [26] and silhouette consistency [12], which has been proved to be very helpful in camera calibration. In this work, we propose a method to incorporate geometric constraints in camera localization. Specifically, we jointly optimize the set of silhouettes and the camera poses, requiring that the silhouettes and the camera poses are consistent.

Existing related approaches [12, 2, 26] usually assume that a set of accurate silhouettes are provided. However in practice segmenting object in all viewpoints requires tedious user interactions. On the other hand, automatic multiview segmentation methods may fail in general cases, or the outputted silhouettes are not accurate enough to provide useful constraints for localization. To cope with this issue, our proposed method jointly perform a silhouette-consistent multiview segmentation on the fly while optimizing camera localization.

In multiview segmentation, accurate camera poses are required to propagate segmentation cues across views. In localization, accurate and silhouettes are needed to provide useful constraints to camera poses. Therefore the two problems are intertwined with each other and a joint treatment is preferred. Although both multiview segmentation and RGB-D camera localization have been intensively studied in literature, few approaches have modeled the two problems jointly.

To this aim, we describe an RGB-D object scanning pipeline consisting of two steps, i.e. an online keyframe collecting step and an offline joint optimization step. At the online step, a user walks around an object with a depth camera in hand. Meanwhile a realtime tracker, e.g. KinectFusion [19], evenly captures a set of keyframes covering the

*The author was partially supported by the NSF of China under Grant 61173081, the Guangdong Natural Science Foundation, China, under Grant S2011020001215, and the Guangzhou Science and Technology Program, China, under Grant 201510010165.

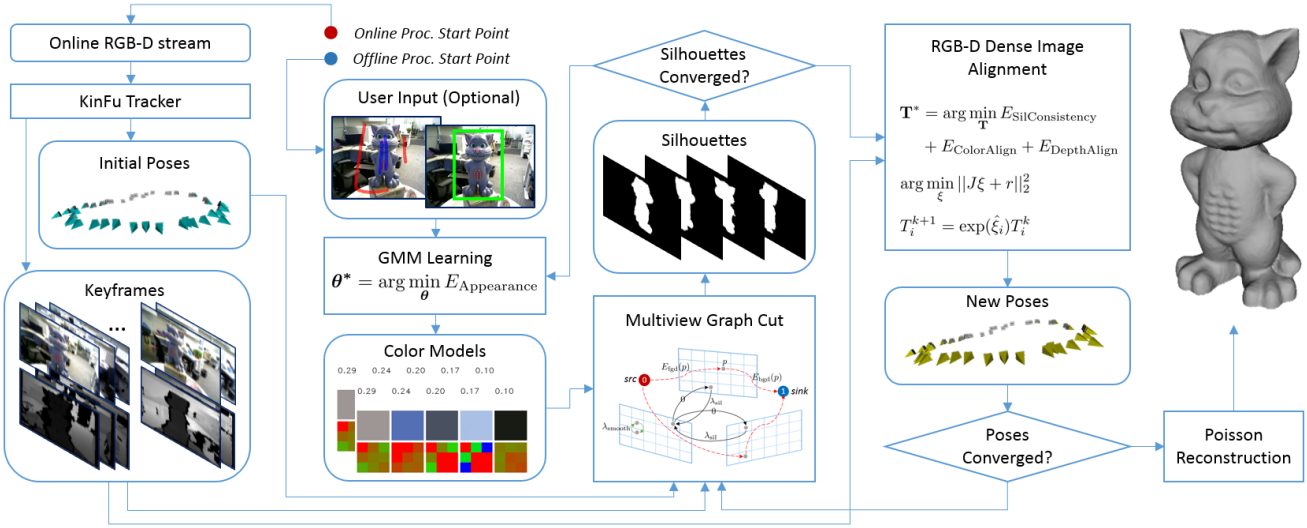

Figure 1. System Overview

object. At the offline step, the proposed approach is adopted to jointly estimate the segmentation masks and the poses of keyframes. For simplicity and without loss of generality, we assumed the color and depth frames are already aligned. Fig. 1 provides an overview of the system.

To model the two problem jointly, we introduce a novel silhouette consistency term to constrain both the segmentation masks and the camera poses. The silhouette consistency term has following merits:

1. It effectively penalizes inconsistent labeling between views, facilitated by the available depth.

2. It is sub-modular, enabling an efficient optimization.

Due to the silhouette consistency term, the joint segmentation and localization problem can be decoupled into two sub-problems and efficiently solved using off-the-shelf approaches, i.e. color model-based segmentation methods [20, 7, 8] and graph-based RGB-D SLAM model [14]. Optimization is iterated between estimating the binary foreground and backgorund labeling, whose objective is sub-modular and naturally fits in the graph cut framework [3, 20], and refining the camera poses, which naturally fits in the Gauss-Newton non-linear least-squares method.

2. Related Work

2.1. Multiview segmentation

Object segmentation from images has been studied intensively in literature and has been extended from monocular case [3, 20, 13, 7] to object co-segmentation [24, 25] and to multiview configuration [17, 8, 9]. Existing multiview segmentation methods can be broadly classified in two streams. The first stream focuses on a volumetric 3D reconstruction of the object, and then computes the segmentation as a byproduct by reprojecting the 3D model back to each view [15, 11, 22, 4]. These approaches suffer from the volume size limit and are not pixel-accurate for high-resolution inputs. The second stream works on image domain. Some solve a binary MRF for each view using the unary term to carry information propagated by other views [17, 9]. Some optimize an MRF containing all views and randomly generated 3D samples [8]. Our approach falls into this stream, and solves an MRF simultaneously for all views. But unlike existing methods, we explicitly model silhouette consistency w.r.t. camera poses.

2.2. Camera localization

RGB-D SLAM is also an intensively studied problem, in which typical systems usually consist of two steps, i.e. online tracking and offline optimization. For online tracking, many approaches have been developed, e.g. frame-to-model matching [19] and frame-to-frame matching [14]. Beyond depth, additional features such as colors [14], local features [23], undistorted depth [27] and contour information [28], have been explored. The work of Zhou et al. [28] is the most related one to ours, which explicitly takes contour information into consideration when matching a frame to a model in the KinectFusion [19] framework. However, this work is an online tracking method which does not consider all frames globally as in an offline optimization process. A major problem of online tracking is pose drift. Loop closure detection and pose graph optimization are effective tools to address the problem in an offline optimization process [14, 27, 18]. Our approach is a kind of offline optimization method. Different from existing techniques, we exploit another type of constraint, i.e. silhouette consistency, and

enforce it in a new way by a proposed silhouette consistency energy term facilitated by the available depth.

3. Formulation

Figure 1 shows the pipeline of the proposed approach which consists of an online data capture component and an offline optimization component. The offline optimization component consists of two modules, i.e. multiview segmentation and RGB-D image alignment. We first present the two modules independently, and then introduce a new silhouette consistency term which enables a joint optimization between them.

3.1. Notation

The input is a multiview sequence of keyframes containing N RGB-D images $\{I_i\}_{i=1}^N$, $\{D_i\}_{i=1}^N$, and initial poses $\{T_i\}_{i=1}^N$, which are usually obtained by a continuous tracker, e.g. KinectFusion [19]. The variables we want to optimize are

- $\{S_i\}_{i=1}^N$, the set of binary-valued silhouettes, where $S_i(p) = 1$ denotes p is a foreground pixel, and $S_i(p) = 0$ denotes background;
- $\{\theta_i^{\text{fgd}}\}_{i=1}^N$, $\{\theta_i^{\text{bgd}}\}_{i=1}^N$, the set of foreground and background color models;
- $\{T_i\}_{i=1}^N$, the set of camera poses that map local coordinates to world coordinates.

For convenience we denote them collectively as $\mathbf{S}, \boldsymbol{\theta}, \mathbf{T}$ respectively. In the following, we will use i, j to index images, and p, q to index pixels.

3.2. Multiview Segmentation w.r.t. $\mathbf{S}, \boldsymbol{\theta}$

As a common objective in object segmentation, we want the binary labeling to agree with the foreground/background color models [20], which is enforced by the appearance energy

$$E_{\text{Appearence}}(\mathbf{S}, \boldsymbol{\theta}) = \sum_i \sum_{p \in \Omega_i} -\text{Prob}(I_i(p) \mid S_i(p), \theta_i^{\text{bgd}}, \theta_i^{\text{fgd}}) \tag{1}$$

where Ω_i denotes the set of pixels in the i-th image, and $\text{Prob}(I_i(p) \mid S_i(p), \theta_i^{\text{bgd}}, \theta_i^{\text{fgd}})$ denotes the probability that color $I_i(p)$ belongs to the foreground color model θ_i^{fgd} if $S_i(p) = 1$, or the probability that $I_i(p)$ belongs to θ_i^{bgd} if $S_i(p) = 0$. For each view, we train a Gaussian Mixture Model (GMM) for foreground and background respectively. Each GMM has five components in all experiments. Our experiments showed that in most indoor environment where RGB-D images are usually captured, the number components are good enough to model the color distributions.

The color models are efficiently learned from the initial set of silhouettes, which is obtained by projecting the visual hull induced by all image rectangles back to 2D. A user can additionally place a bounding box or draw scribbles to further constrain the problem. It is noted that the user only need to provide guidance in a few views, since our silhouette consistency term introduced in section 3.4 is able to effectively propagate these information across views. During the segmentation process, more guidance can be given in each iteration if the user is not satisfied with the results.

We also encourage the labeling to be smooth and aligned with image edges

$$E_{\text{Smooth}}(\mathbf{S}) = \sum_i \sum_{p,q \in \mathcal{N}_4} w_{pq} ||S_i(p) - S_i(q)||^2 \tag{2}$$

where $\mathcal{N}_4$ denotes a 4-neighborhood on image grid, $w_{pq} = \exp(-||I_i(p) - I_i(q)||/\gamma_1 - ||D_i(p) - D_i(q)||/\gamma_2)$ is a weight to encourage discontinuity on edges. For pixels without depth, w_{pq} only considers color.

3.3. RGB-D Image Alignment w.r.t. $\mathbf{T}$

We adopt the frame-to-frame matching approach proposed by Kerl et al. [14] to model both color and depth alignment error

$$E_{\text{ColorAlign}}(\mathbf{T}) = \sum_i \sum_{p \in \tilde{\Omega}_i} \sum_{j \in \mathcal{N}_i} ||I_i(p) - I_j(q)||^2 \tag{3}$$

$$E_{\text{DepthAlign}}(\mathbf{T}) = \sum_i \sum_{p \in \tilde{\Omega}_i} \sum_{j \in \mathcal{N}_i} ||D_i(p) - D_j(q)||^2 \tag{4}$$

where $\tilde{\Omega}_i$ is the set of pixels with valid depths, $\mathcal{N}_i$ is the set of neighboring cameras of the i-th keyframe in a pose graph, and

$$q = \pi_j(T_j^{-1} T_i \pi_i^{-1}(p, D_i(p))) \tag{5}$$

is pixel p's correspondence in image j, with π_j, π_i^{-1} being the corresponding projection and inverse-projection respectively. Note that the poses $\mathbf{T}$ are parametrized in $se(3)$ during optimization [1].

Before the global optimization, we need to construct a graph of keyframes, in which an edge connecting two keyframes means that the two frames view a large overlap of the common surface. Due to the trajectory drift problem, we need to carefully establish graph edges containing necessary loop-closures. Specifically, for each keyframe we collect candidates of neighbor frames by checking the angles of principal axis and distances of camera positions. If the two cameras meet a condition (e.g. angle $\leq 60^\circ$ and distance ≤ 0.5m), we further validate it by doing a dense alignment, i.e. minimizing Eq. (3) and (4) over a two-node graph. After the alignment, we count the number of

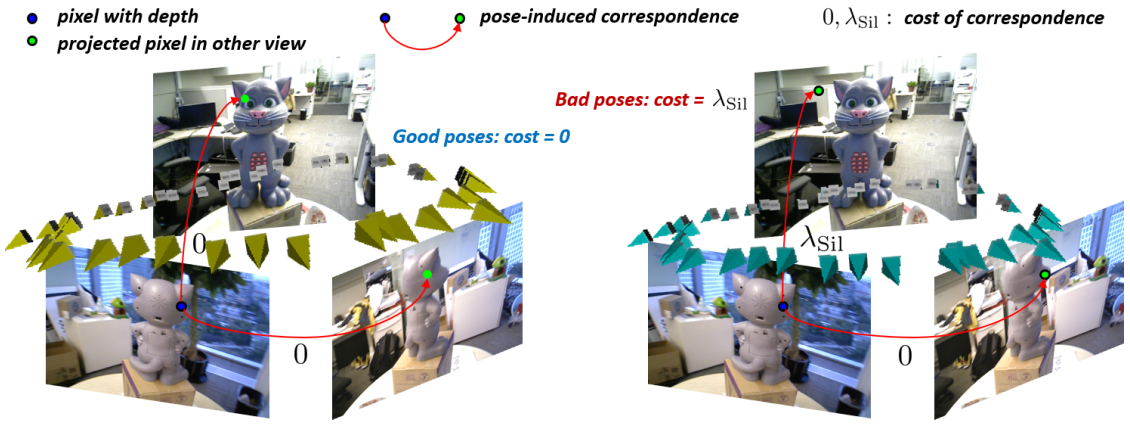

Figure 2. Silhouette consistency term constrains both near and far views. A pixel inside the silhouette of a back view is projected to a near (side) view and a far (front) view via depth and camera poses. The left figure shows a consistent case, while the right figure shows an inconsistent case which introduces a cost λ_{sil}.

matched pixels (e.g. difference of depths $\leq$ 0.5cm and angle of normals $\leq 15^\circ$). And if the ratio of matched pixels to total pixels with valid depth is above a threshold (e.g. 0.6), we establish an edge to connect them in the pose graph.

Since in object scanning, images are usually captured in an outside-in mode, the trajectory of cameras is not as complex as in large-scale scene modeling [14, 27], and seldom drifts significantly. The loop-closure detection in object scanning is less challenge than in large-scale scene modeling. The above simple strategy successfully detected necessary loop-closures in all our experiments.

3.4. Silhouette Consistency w.r.t. $\mathbf{S}, \mathbf{T}$

So far camera localization and multiview segmentation are modeled independently, but as we have argued a joint treatment would benefit them from each other. To this aim, we introduce the following silhouette consistency term

$$E_{\text{SilConsistency}}(\mathbf{S}, \mathbf{T}) = \sum_i \sum_{p \in \tilde{\Omega}_i} \sum_{j \neq i} S_i(p) \cdot ||S_i(p) - S_j(q)||^2 \tag{6}$$

Eq. (6) looks very similar to Eq. (3) and (4). Readers may understand as that a silhouette is only an additional channel beyond depth and color, which supplement object contour information to the optimization. However, it is significantly different from color and depth channels due to the following properties.

First, the silhouette consistency depends on both the segmentation $\mathbf{S}$ and the poses $\mathbf{T}$, therefore it connects and regularize both $\mathbf{S}$ and $\mathbf{T}$; Second, the subscript j in (6) ranges over all images instead of only the neighboring views as in (3)(4). This property is crucial and helps to prevent incremental pose drift during optimization. As shown in Fig. 2, even a back view of the object provides constraints (for both segmentation and localization) and hints (for segmentation) to a front view; Third, the penalty on a pixel p is active only when $S_i(p) = 1$, i.e. when p is a foreground pixel. This is coherent with the mathematical definition of silhouette consistency, i.e. any 3D point that lies on the object's surface must project inside all other silhouettes in 2D, while a 3D point that lies outside the surface could project to either inside or outside of other silhouettes.

3.5. Overall Energy

Putting all the pieces together, we obtain the overall objective function

$$\begin{aligned} E_{\text{All}}(\mathbf{S}, \boldsymbol{\theta}, \mathbf{T}) = & E_{\text{Appearence}}(\mathbf{S}, \boldsymbol{\theta}) + E_{\text{Smooth}}(\mathbf{S}) \\ & + E_{\text{DepthAlign}}(\mathbf{T}) + E_{\text{ColorAlign}}(\mathbf{T}) \\ & + E_{\text{SilConsitency}}(\mathbf{S}, \mathbf{T}) \end{aligned} \tag{7}$$

Without the silhouette consistency term (6), segmentation and localization would have become two independent problems as the binary masks and the camera poses would have nothing to interact on. The silhouette consistency term enables a joint formulation of the two problems, which provides constraints and hints for both tasks.

4. Optimization

The objective function depends on both discrete and continuous sets of variables, whose minimization is challenging. Luckily the overall objective can be decomposed into two subproblems, namely segmentation and localization

$$E_{\text{Segmentation}} = E_{\text{Appearance}} + w_1 E_{\text{Smooth}} + \lambda_{\text{Sil}} E_{\text{SilConsistency}} \tag{8}$$

$$E_{\text{Localization}} = E_{\text{DepthAlign}} + w_2 E_{\text{ColorAlign}} + \lambda_{\text{Sil}} E_{\text{SilConsistency}} \tag{9}$$

which can be optimized using off-the-shelf methods, i.e. Graph Cut [16] and Gauss-Newton non-linear least-squares

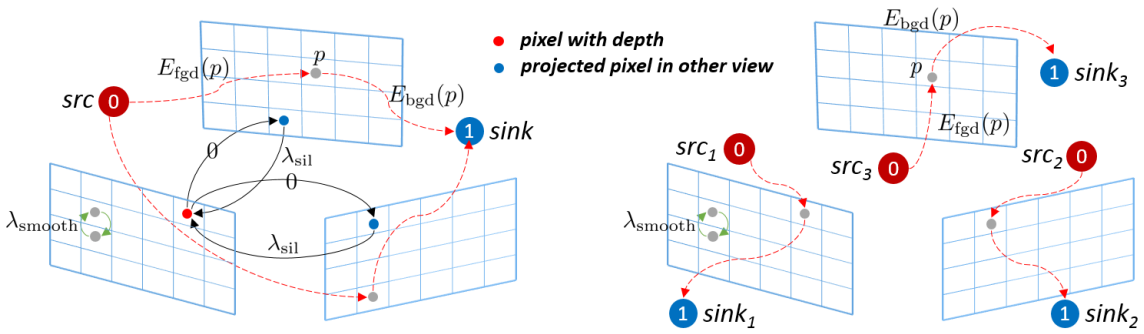

Figure 3. (Left) Segmentation Graph of our approach, labeling of all views are solved simultaneously. (Right) Segmentation graph of [20, 17, 9]. A single graph cut is run independently for each image. Possible multiview segmentation cues are preprocessed and encoded in unary terms.

Algorithm 1 Optimize E_{All}

Initialize $\mathbf{S}, \boldsymbol{\theta}$ by any monocular segmentation method.
Initialize $\mathbf{T}$ by minimizing $E_{\text{DepthAlign}} + E_{\text{ColorAlign}}$ using non-linear least squares [10].
repeat
  repeat
    Fix $\mathbf{T}$, $\mathbf{S}$, $\min_{\boldsymbol{\theta}} E_{\text{Segmentation}}$ by GMM EM learning.
    Fix $\mathbf{T}$, $\boldsymbol{\theta}$, $\min_{\mathbf{S}} E_{\text{Segmentation}}$ by Graph Cut [16].
  until converged
  Fix $\mathbf{S}, \boldsymbol{\theta}$, $\min_{\mathbf{T}} E_{\text{Localization}}$ by non-linear least squares.
until converged

method [10]. Minimization of the original problem is then reduced to solving the two subproblems iteratively. Alg. 1 provides an overview of the optimization. We set the coefficients $w_1 = 50$, $w_2 = 0.1$, and $\lambda_{\text{Sil}} = 0.1$ in all experiments.

4.1. Optimize Segmentation

We optimize the segmentation subproblem in a block gradient descent fashion, as shown in Alg. 1. When keeping the foreground masks $\mathbf{S}$ fixed, the parameters of each image's GMM color model can be re-estimated by EM algorithm.

When optimizing $E_{\text{Segmentation}}$ w.r.t. segmentation masks $\mathbf{S}$, it becomes a discrete optimization problem. Both $E_{\text{Appearance}}$ and E_{Smooth} are widely used submodular energies. And it is easy to check that the $E_{\text{SilConsistency}}$ is also submodular

$$E_{\text{SilConsistency}}(0,1) + E_{\text{SilConsistency}}(0,1) = 1 \quad (10)$$
$$> E_{\text{SilConsistency}}(0,0) + E_{\text{SilConsistency}}(1,1) = 0 \quad (11)$$

Therefore the subproblem can be efficiently solved by graph cut [16].

Fig. 3 compares the segmentation graph of our approach to some of the existing approaches [20, 17, 9]. Our silhouette consistency term acts as a kind of smoothness constraints to regularize labeling across images. In contrast to [20, 17, 9], our approach segments all images simultaneously by one graph cut. Labeling cues in one view are effectively propagated to all other views via known depth and camera poses. Hard-to-segment regions in one view, e.g. regions close to silhouette boundaries where depth usually misses, will get hints from other views, in which corresponding regions may have depth and be easy to segment.

4.2. Optimize Localization

With $\mathbf{S}, \boldsymbol{\theta}$ fixed, the localization subobjective $E_{\text{Localization}}$ is the sum of all quadratic terms with $\mathbf{T}$ and therefore can be effectively solved by the Gauss-Newton non-linear least-squares method. Specifically, we parameterize T_i by a 6-vector $\xi_i = (a_i, b_i, c_i, \alpha_i, \beta_i, \gamma_i)$ that represents an incremental transformation relative to the current T_i. Here (a_i, b_i, c_i) is a translation vector, and $(\alpha_i, \beta_i, \gamma_i)$ can be interpreted as angular velocity. Stacking all ξ_i together, we get a $6N$-dimensional variable ξ. To solve each iteration we calculate the linearized least-squares solution

$$\arg\min_{\xi} \|\mathbf{J}\xi + \mathbf{r}\|_2^2 \quad (12)$$

where $\mathbf{J}$ is the Jacobian and $\mathbf{r}$ is the residual vector. Both $\mathbf{J}$ and $\mathbf{r}$ are linear combinations of three terms computed from term (3), (4) and (6). Solving the linear equation yields an improved camera transformation

$$T_i^{k+1} = \exp(\hat{\xi}_i) T_i^k \quad (13)$$

Details of derivation on the Jacobian are provided in the supplementary material.

To prevent poses from trapping in bad local minimal and to improve optimization speed, we adopt a three level coarse-to-fine pyramid scheme. Blocks of the combined measurement Jacobian $\mathbf{J}_i$ and residual $\mathbf{r}_i$ can be computed in GPU, and reduce to a single $6N \times 6N$ linear equation. Then we solve it on CPU by the Cholesky decomposition.

5. Experiment

To evaluate the proposed approach, we collected ten RGB-D datasets with the ASUS Xtion sensor. Fig. 4

shows some sample color frames. The KinFu tracker [19] is used to continuously track the online stream, and a new keyframe is saved when its relative rotation or translation to the last keyframe is larger than 10° or 10cm. Each sequences consists of about 60 keyframes of 640×480 depth and color images, with their corresponding initial poses. The proposed approach takes about 200s to run on a regular PC and outputs refined camera poses, silhouettes, and a high-quality 3D mesh model. All keyframes are manually segmented to generating ground truth silhouettes. We set $\gamma_1 = 30$ and $\gamma_2 = 30$. Results are not very sensitive to these two parameters. To balance any two energy terms $\lambda_1 E_1 + \lambda_2 E_2 = \lambda_1 \sum_k e_1^k + \lambda_2 \sum_k e_2^k$ in Eq. (8) and (9), we determine the medians $e_1^{\text{med}}, e_2^{\text{med}}$ of $\{e_1^k\}$, $\{e_2^k\}$ respectively, and set $\lambda_1/\lambda_2 = e_2^{\text{med}}/e_1^{\text{med}}$. This strategy works well in practice.

5.1. Segmentation

An important feature of our object scanning system is it is able to generate a set of silhouettes on the fly. Given the keyframes as input, initial silhouettes are generated by projecting the commonly visible part of 3D space back to each image, which is the intersection of all viewing cones induced by the image rectangles. Here, we assume the object completely appears in all views. Although our system enables a user to provide bounding boxes or scribbles to guide the segmentation, we did not make use of the user input in experiments. The GMM color models are initialized from these initial masks.

Tbl. 1, provides a quantitative evaluation of the segmentation results among Grabcut [20], Djelouah'13 [8], Diebold'15 [7] and ours. Accuracy is measured by the percentage of mislabled pixels compared to hand-labeled ground truths. Grabcut performs inferior compared to all methods since multiview geometric cues are not explored. In some cases, masks in Djelouah'13 appear to be inflating since the 3D consistency enforced by its sparse 3D samples does not penalize background pixels being labeling foreground. To achieve accurate results, Diebold'15 needs about 5.3 scribbles for each image in average since it cannot leverage on results/guidance of other views. Our results outperform the others, since the silhouette consistent term makes substantial use of the available depth and enforce the consistency among multiple views explicitly.

5.2. Localization

Directly evaluating the accuracy of camera poses is a challenging task since ground truth poses are difficult to obtain in general. Instead, we evaluate poses by two indirect measures, i.e. the calibration ratio [2] and accuracy of the reconstructed 3D model. Calibration ratio is based on the observation that given a set of perfect silhouettes and perfect camera poses, the viewing ray of every foreground pixel should intersect with the silhouette-induced viewing cones of all the other views in a common intersection. And the ratio for image i is defined as

$$C_i = \frac{1}{|\mathcal{M}_i|\,(N-1)} \sum_{p \in \mathcal{M}_i} \Phi(r_p) \tag{14}$$

where $\mathcal{M}_i$ is the set of foreground pixels of image i, N is the number of cameras. r_p is the induced viewing ray of pixel p, and $\Phi(r_p)$ is the maximum number of cameras whose viewing cones induced by their own silhouettes have at least one common interval along r_p. Therefore, if both camera poses and silhouettes are perfect, calibration ratio is equal to one, otherwise, it will be less than one. Since camera poses and silhouettes are the only two reasons that affect the calibration ratio, if we fix the silhouettes to be the manually labeled ground truth, calibration ratio is a good measure of the accuracies of camera poses.

Tbl. 2 shows the averaged calibration ratios over all cameras in each iteration. The Calibration ratios steadily increase along iteration, which indicates that the poses are becoming more and more accurate. Calibration ratios converges after four iterations in almost all cases we tested. Fig. 7 provides a visual comparison of the reconstructed models. Without the silhouette consistent energy, our approach reduce to the RGB-D alignment approach presented in Section 3.3, which is itself a typical offline optimization method for improving camera poses [14]. There for we use it as a baseline. As shown in the figure, models generated by our joint optimization preserve more fine details, such as the keyboard on the belly of Tomcat. Beyond the usage in calibration, silhouettes can help depth integration [5] and mesh optimization [21] to preserve fine structures. Further discussion on this direction is beyond scope of this work.

We scanned the Tomcat and MusicBox by a commercial high-quality 3D scanner[1] whose precision is about 1mm in general. Fig. 6 shows model errors of Kinfu, RGB-D alignment only and the joint optimization. As shown in the figure, our model obtains significantly lower errors. Since Kinfu has no offline optimization, drift of camera poses significant hurts model qualities.

6. Conclusion

We have presented an RGB-D camera localization approach that effectively exploits the silhouette constraints. Unlike existing silhouette-based calibration approaches which usually assume accurate silhouettes are provided, our system is able to generate object silhouettes on the fly during optimization, making its usage very practical. Experiments demonstrated large improvements on both tasks of object segmentation and camera localization.

[1] Artec3D, http://www.artec3d.com

Figure 4. Sample keyframes of our collected datasets.

	Tomcat	Musicbox	Donkey	Dragon	Chair	Horse	Plant	Gundam	Bag	Bag2
GrabCut [20]	5.31	4.78	5.63	4.24	0.98	5.20	5.98	9.31	3.38	6.17
Djelouah'13 [8]	1.54	1.46	1.63	1.27	0.26	1.50	4.20	3.00	0.85	1.92
Diebold'15 [7]	**0.35**	0.37	0.31	0.29	0.14	0.63	2.73	0.52	**0.20**	0.39
Ours	**0.35**	**0.35**	**0.28**	**0.20**	**0.13**	**0.41**	**2.03**	**0.48**	0.23	**0.31**

Table 1. Comparison of error rates of generated silhouettes.

(a) Color image (b) Grabcut [20] (c) Djelouah'13 [8] (d) Diebold'15 [7] (e) Ours

Figure 5. Examples of generated silhouettes.

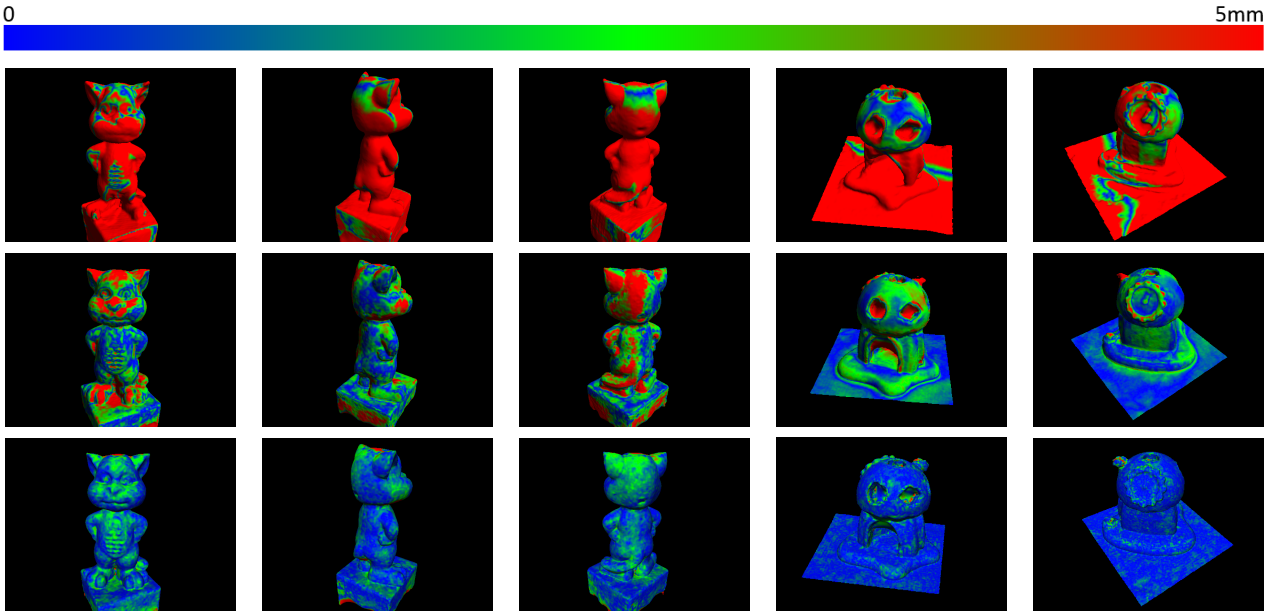

Figure 6. Quantitative evaluation of model error. Row 1-3 show results of Kinfu [19], RGB-D alignment only (i.e. our approach with silhouette consistency term disabled) and joint optimization, respectively.

	Tomcat	Musicbox	Donkey	Dragon	Chair	Horse	Plant	Gundam	Bag	Bag2
Initial	0.75	0.79	0.66	0.78	0.56	0.63	0.55	0.51	0.66	0.77
Iter 1	0.83	0.84	0.82	0.88	0.74	0.81	0.69	0.87	0.89	0.93
Iter 2	0.89	0.91	0.90	0.90	0.89	0.94	0.77	0.90	0.93	0.94
Iter 3	0.97	0.94	0.92	0.92	0.95	0.98	0.88	0.92	0.95	0.96

Table 2. Averaged calibration ratios increase with iterations.

Figure 7. Visual comparison of generated meshes. Row 2-4 are meshes generated by Kinfu [19], RGB-D alignment only (i.e. our approach with silhouette consistency term disabled) and joint optimization. Row 5-7 show close-up views of the respective models.

References

[1] J.-L. Blanco. A tutorial on se (3) transformation parameterizations and on-manifold optimization. *University of Malaga, Tech. Rep*, 2010. 3

[2] E. Boyer. On using silhouettes for camera calibration. In *Computer Vision–ACCV 2006*, pages 1–10. Springer, 2006. 1, 6

[3] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112. IEEE, 2001. 2

[4] N. D. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Automatic 3d object segmentation in multiple views using volumetric graph-cuts. *Image and Vision Computing*, 28(1):14–25, 2010. 2

[5] D. Cremers and K. Kolev. Multiview stereo and silhouette consistency via convex functionals over convex domains. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(6):1161–1174, 2011. 6

[6] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, pages 303–312, 1996. 1

[7] J. Diebold, N. Demmel, C. Hazırbaş, M. Moeller, and D. Cremers. Interactive multi-label segmentation of rgb-d images. In *Scale Space and Variational Methods in Computer Vision*, pages 294–306. Springer, 2015. 2, 6, 7

[8] A. Djelouah, J.-S. Franco, E. Boyer, F. Le Clerc, and P. Perez. Multi-view object segmentation in space and time. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2640–2647. IEEE, 2013. 2, 6, 7

[9] A. Djelouah, J.-S. Franco, E. Boyer, F. Le Clerc, and P. Perez. Sparse multi-view consistency for object segmentation. 2015. 2, 5

[10] J. Engel, T. Schöps, and D. Cremers. Lsd-slam: Large-scale direct monocular slam. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014. 5

[11] J.-S. Franco and E. Boyer. Fusion of multiview silhouette cues using a space occupancy grid. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1747–1753. IEEE, 2005. 2

[12] C. Hernández, F. Schmitt, and R. Cipolla. Silhouette coherence for camera calibration under circular motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(2):343–349, 2007. 1

[13] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988. 2

[14] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for RGB-D cameras. In *2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 6-10, 2013*, pages 3748–3754, 2013. 1, 2, 3, 4, 6

[15] K. Kolev, T. Brox, and D. Cremers. Fast joint estimation of silhouettes and dense 3d geometry from multiple images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):493–505, 2012. 2

[16] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147–159, 2004. 4, 5

[17] A. Kowdle, S. N. Sinha, and R. Szeliski. Multiple view object cosegmentation using appearance and stereo cues. In *Computer Vision–ECCV 2012*, pages 789–803. Springer, 2012. 2, 5

[18] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3607–3613, Shanghai, China, May 2011. 2

[19] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011, Basel, Switzerland, October 26-29, 2011*, pages 127–136, 2011. 1, 2, 3, 6, 7, 8

[20] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004. 2, 3, 5, 6, 7

[21] S. N. Sinha and M. Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *10th IEEE International Conference on Computer Vision (ICCV 2005), 17-20 October 2005, Beijing, China*, pages 349–356, 2005. 6

[22] D. Snow, P. Viola, and R. Zabih. Exact voxel occupancy with graph cuts. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 345–352. IEEE, 2000. 2

[23] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012. 1, 2

[24] S. Vicente, V. Kolmogorov, and C. Rother. Cosegmentation revisited: Models and optimization. In *Computer Vision–ECCV 2010*, pages 465–479. Springer, 2010. 2

[25] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2217–2224. IEEE, 2011. 2

[26] K.-Y. K. Wong and R. Cipolla. Reconstruction of sculpture from its profiles with unknown camera positions. *Image Processing, IEEE Transactions on*, 13(3):381–389, 2004. 1

[27] Q. Zhou and V. Koltun. Simultaneous localization and calibration: Self-calibration of consumer depth cameras. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 454–460, 2014. 2, 4

[28] Q. Zhou and V. Koltun. Depth camera tracking with contour cues. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 632–638, 2015. 1, 2