HUMAN 3D RECONSTRUCTION AND MOTION CAPTURE USING A SINGLE FLYING CAMERA

by

WEI CHENG

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Master of Philosophy
in Electronic and Computer Science and Engineering

August 2018, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Wei CHENG

WEI CHENG

August 2018, Hong Kong

HUMAN 3D RECONSTRUCTION AND MOTION CAPTURE USING A SINGLE FLYING CAMERA

by

WEI CHENG

This is to certify that I have examined the above MPhil thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.

Prof. Ling SHI, Thesis Supervisor

Prof. Lu FANG, Thesis Co-Supervisor

Prof. Bertram SHI, Head of Department

Thesis Examination Committe

Prof. Jungwon SEO (Chairperson), Department of Electronic and Computer Engineering

Prof. Ling SHI (Supervisor), Department of Electronic and Computer Engineering

Prof. Lu FANG (Co-Supervisor), Department of Electronic and Computer Engineering

Prof. Jun ZHANG, Department of Electronic and Computer Engineering

Department of Electronic and Computer Engineering
1 August 2018

ACKNOWLEDGMENTS

I was grateful for receiving help of so many people during my master period. First of all, I have to give my sincere appreciation to my advisors, Professor Ling Shi, and Professor Lu Fang, for their years of mentoring, advice, and encouragement. I have learned a lot from them how to research and express my ideas. I thank Professor Yebin Liu from Tsinghua University, China for advising me on dynamic reconstruction.

I thank my colleagues in HKUST – Mr. Lan Xu, we collaborate several works together with excellent vibes, Mr. Xiaoyun Yuan, Mr. Mengqi Ji, Mr. Minjian Pang, Mr. Xiongfei Su and many others. They are good friends and gave me tons of help in daily life.

Lastly, I want to thank my family sincerely, for their care and companionship in my whole life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

HUMAN 3D RECONSTRUCTION AND MOTION CAPTURE USING A SINGLE FLYING CAMERA

by

WEI CHENG

Department of Electronic and Computer Science and Engineering

The Hong Kong University of Science and Technology

ABSTRACT

With the emergence of consumer RGB-D camera, interdisciplinary research in computer vision, graphics and robotics experienced huge growth in recent years. Aiming at the intelligent human body 3D reconstruction and motion capture, we adopt the aerial robot that employed with RGB-D camera as the flying camera, and present two novel works in this thesis: ***iHuman3D*** and ***FlyFusion*** for automated, adaptive, and real-time human body 3D reconstruction and motion capture.

Specifically, for static human full body 3D reconstruction, a real-time and active view planning system ***iHuman3D*** is proposed based on a highly efficient ray casting algorithm in GPU and a novel information gain formulation directly in Truncated Signed Distance Function (TSDF). Human body reconstruction module revises the traditional volumetric fusion pipeline with a compactly-designed non-rigid deformation for slight motion of the human target. Both the active view planning and human body reconstruction are unified in the same TSDF volume-based representation. On the other hand, for dynamic human motion capture, following the active reconstruction clue, a Geometry And Motion Energy (GAME) metric for guiding the viewpoint optimization in the volumetric space, proposed

FlyFusion succeeds to enable active viewpoint selection based on the immediate dynamic reconstruction geometry and predicted human motion. Quantitative and qualitative experiments are conducted to validate that the proposed systems effectively remove the constraints of fixed capture volume and extra manual labor, enabling real-time and intelligent human body 3D reconstruction and motion capture. Given above distinctiveness, we believe our work bridges the gap in robotics and human modelling, and will further promote both robotics, computer vision and graphics communities in interactive 3D human reconstruction.

CHAPTER 1

INTRODUCTION

Robust perception and understanding with humans present can enable numerous applications, such as human robot interaction, human motion analysis and recognition, computer games, and virtual reality. Benefit from the emergence of depth sensor that can capture depth and image data at video rate, above tasks based on depth sensor has received great attention in the fields of multimedia, computer vision, graphics and robotics etc. Among them, consumer-level depth camera like Kinect is widely used for 3D reconstruction due to the advantages of low-price, compact and portable [63, 43, 85].

On the other hand, increasingly mature robotics technology has taken a big step forward on robots' capability of executing numerous tasks boosted by merging visual perception techniques like visual inertial odometry (VIO) and simultaneous localization and mapping (SLAM). With orderly flexibility, stability controllability, aerial robots mounted with multiple sensors has been successfully applied in many human perception tasks including human tracking [52], gesture interaction [75]. Potential application scenarios of human perception using an aerial robot mounted with a depth camera which we call it flying camera[1] are explored in this thesis. Two scenarios are concerned, human body reconstruction and motion capture which both used to suffer from the constraints of human labor and expertise requirement [94], recording space restriction [87, 56], user comfortability [46] in conventional human centered systems. To overcome the above constraints and achieve automation and adaptiveness, human body reconstruction and motion capture are consider separately in this thesis.

In order to scan a complete human body, given the small Field of View (FOV) of consumer RGB-D cameras and human height, it needs to work around 2 meters away from the target. Due to the depth noise model which will be discussed in later chapters, extremely noisy geometric information can be captured in such distance. While fusion across

[1] The flying camera refers to an autonomous aerial robot that equipped with a consumer-grade low weight depth camera, serving to capture RGB-D video stream with acceptable quality.

multiple frame can be employed to improve the final smooth mesh [17], the result is still far from meticulous. Tong *et al.* [76] proposed a 3D full human body shapes scan system using multiple Kinects. However, all aforementioned approaches are still restricted by static camera array or human handhold cameras to follow the performer. Although autonomous view point selection using information gain metric is a well studied topic which is called Next Best View (NBV) in robotics community, the existing approaches still suffer from low processing speed [19, 40, 23] or naively defined [60]. An intelligent real-time human body 3d reconstruction scheme ***iHuman3D*** that uses a single flying camera is proposed to remove the constraints and labor in terms of the expert knowledge, target and capture volume.

Specifically, a human-centered volume in Truncated Signed Distance Function (TSDF) representation [18] is maintained similar in conventional real-time dense fusion scheme [63, 62, 32], which aligns and fuses the temporal information of input raw data from the depth sensor into a canonical human body model in real-time. Considering that non-rigid human target be hard to maintain a stationary pose and 3D geometry, a dynamic reconstruction model with embedded deformation node and warp field are used like in DynamicFusion [62]. Note that in the computer vision and computer graphics communities, consecutive works after DynamicFusion [62] tend to focus on embedding new information for the final reconstructed model, yet analysis on influence of the view points for the dynamic object still remain lack of investigation. Whereas, we consider active view planning by solving a next best view evaluation, in order to achieve adaptive and autonomous reconstruction. For consistency with our volumetric fusion pipeline, we proposed a volumetric occupancy probability model in TSDF, occupancy status is defined in term of TSDF value and weights, a further occupancy probability integration scheme is introduced. Based on the novel volumetric occupancy probability model, we adopt NBV evaluation by accumulating information gain (IG) [6, 40, 23, 19] along all the rays casted to the probability volume. It is worth noting that the proposed model is the first attempt to formulate the geometry information and occupancy probability in a uniform TSDF representation. Henceforth, a highly parallel and efficient algorithm to calculate the IG based on modern GPU hardware is introduced to enable the real-time performance of ***iHuman3D***. On the basis of information gain evaluation, smooth scan trajectory is generated by regularization on robot motion and trajectory smoothness.

For human motion capture, distinct from reconstruction task, view point optimization with dynamic human motion is the key problem to address, while it is very challenge in two folders: 1) No available existing views effectiveness evaluation metric on dynamic scene that the capturing target usually perform complicated and unpredictable motions. Meanwhile, the evaluation has to be solved at the video sampling rate within several milliseconds, for quick responds and prediction for fickle human motion. 2) As to view point selection criteria which is tightly dependent on immediate reconstruction output, robust dynamic reconstruction scheme is indispensable. Adversely, topology changes still remain arduous and fragile in most existing real-time motion capture systems. Moreover, the degradation in depth input may pare the endurance to motion extent and lead inevitable reconstruction crash.

To resolve above challenges, problem of active dynamic human motion capture based on a single flying camera is explored. The proposed system ***FlyFusion*** make the first pace forward by adaptive selecting the capture view of one flying camera targeting on real-time dynamic human motion capture. Specifically, different from NBV metric adopted in human body reconstruction which tries to maximize IG [6, 40, 23, 19] among all view candidates, ***FlyFusion*** defined a brand new Geometry And Motion Energy (GAME) metric in the volumetric space which by optimization simultaneously maximizes the raw data acquisition quality, target observation quality, geometry quality and target motion energy of the view candidate. To optimization GAME metric energy, a more meticulous hierarchical searching scheme is designed. On another hand, based on the dynamic reconstruction model employed in ***iHuman3D***, a more robust dynamic motion capture system adopting a novel topology compactness algorithm is proposed which effectively regularized the complex topology changes. Different from conventional dynamic scene reconstruction system, drifting analysis of working volume is originated which explicitly demonstrate the unnoticeable distinct between reconstruction volume and practical space. It's worth mentioning that, the above active view planning and robust dynamic motion capture module achieve mutual improvement: 1) active view point selection module optimizes the data acquisition, object observation, geometry and motion quality according to current reconstruction results via proposed GAME metric offering dynamic guidance of better view points; 2) dynamic motion capture module with high robustness to topological changes provide valid reconstruction results which provide evaluation assurance for active planning module. Note that even though the real-time reconstruction results may not guarantee the completeness of the surface as only the partial surface is captured, the results of GAME strategy evaluated and compared with other capture strategy show that its captured data is more meaningful for both online and offline dynamic scene reconstruction in term of accuracy.

Back to the fundamentals, although consumer-grade RGB-D cameras succeeded in multiple applications, they still surfer from many kinds for noises, like empty holes, unstable laterals, axial noises, and etc as discussed in [64, 59]. Formal literatures on depth image restoration only emphasis on approaches solving part of degradations, for example, inpainting [11, 93, 54], denoising [10, 53, 99] and refinement [49, 45, 96].

Given the above distinctiveness, proposed techniques in active human reconstruction and motion capture bridges the gap among bridges the gap among efficiency, accuracy and adaptability for human perception, and will further promote both computer vision and robotics communities in areas of human robot interaction, human motion analysis and recognition etc.

CHAPTER 2

RELATED WORKS

This chapter delivers an overview of related works to human reconstruction and motion capture. Specifically, thorough literature review on human model reconstruction, human motion capture, active view planning and depth image denoising are discussed in three separate sections.

2.1 Human Model Reconstruction

Acquiring 3D geometric content from real world is an essential task for many applications in robotics domain, computer vision and computer graphics communities. Detailed human models can be created using 3D scanning devices, such as structured light or laser scan. Allen *et al.* [2] employed 74 markers on target body and captured the location of landmarks from range scan, which is then used in mesh stitching and hole filling to reconstruct the complete human templates. However, such devices are too expensive and often require expert knowledge for the operation, meanwhile may bring discomfort to performer because of the placed makers.

The multi-view methods [21, 44] can get relative impressive results with less challenge on loop-closure issues. Hilton *et al.* [35] utilized the multiple camera from front, back and two side views, extracted silhouettes were then matched with predefine generic model to approximate the human template. Auvinet *et al.* [4] proposed a multiple depth camera system with adaptable calibration approach, then a human body volume reconstruction method was used based on visual hulls from multiple views. Some researchers [33, 94] utilized human handheld cameras to follow and reconstruct the human body, which have to rely on extra manual labor. Ye *et al.* [94] proposed a skeleton matching and camera pose estimation method and estimated the deformed human template. This kind of methods is usually computationally expensive, and mutual interferences among multiple active depth cameras who project IR patterns or illuminate the scene with phase modulated lights may bring serve degradation to raw depth data consequently as discussed in [59].

For autonomous human body scanning, Tong *et al.* [76] used multiple Kinects to scan 3D full human body shapes in a restricted capture volume, a non-rigid template-based registration and global alignment method was used to jointly align multiple scans. Commercial systems like *Artec Shapify* Booth [1] can automatically scan a human body inside the booth with whirling depth cameras. However, such methods suffer from the fixed capture volume constraint.

Some researchers have tried to use comsumer-grade depth sensors as 3D scanners for accurate real-time mapping of complex scenes [63, 43, 85, 83]. Newcombe *et al.* [63] proposed to use TSDF [18] as a volume representation, new geometric information were fused into the volume via update scheme. Camera poses were estimated via rigid Iterative Closest Point (ICP) [5] algorithm. These methods utilized the TSDF volume for both representing the geometry information and analyze the camera localization information. Based on that, Li *et al.* [51] proposed a self-portraits method which required target human to rotate himself with respect to the static depth camera. Such method requires self-rotation and trends to fail because target can barely keep stationary while self-ratating.

Most related automatically human reconstruction method is ***FlyCap*** proposed by [91], which employed a flying camera to automatically scan an A-posed human body with a spine-down trajectory. Depth stream was stored in the flying camera, and offline algorithm was adopted to reconstruct the final water-tight 3D mesh. Even though this method emancipated human labor, expert knowledge and restricted capture volume, the scan process is fixed with naive pre-define trajectory, and target requires to stand still with an A pose during scanning.

2.2 Human Motion Capture

Marker based motion capture [86] is a well developed technique which has been successfully applied into many field like avatar animation, motion analysis and virtual reality. Whereas, this kind of method requires the actor to wear maker suits like optical markers [80, 67], inertial measurement units (IMU) [90, 69, 81], or pressure sensors [98].

To emancipate the restriction of additional wearable devices and capture performer with realistic costume, markerless motion capture became a merging technique in last decades. Systems in early stage required the multi-view cameras with controlled chromakey backgrounds to implicitly [73] or explicitly [82, 22] reconstruct dense human motion by extracted human skeleton. Model systems using hundreds of cameras [15, 41] can extract extremely accurate skeletal motion or appearance of the human target. Whereas, high system setup complexity of aforementioned methods makes them hard to replicate. For example, most of the systems demand precise system or camera calibration, accurate segmentation of the actor from all views are also need. Moreover, for both marker based and camera array system, performers are required to stay inside the fixed capture volume indoor. On another hand, Wang *et al.* [84] and Hasler *et al.* [33] introduced a multiple handheld cameras system to capture the performer motion outdoor.

From the algorithm aspect, motion reconstruction can be classified into two main categories: discriminative approaches [71, 29, 9] and generative approaches [8, 28, 24]. The former takes advantage of data driven machine learning strategies to convert the reconstruction problem into a regression or pose classification problem, and is therefore suitable for human-computer interaction applications where real-time efficiency is more important than accuracy. In contrast, generative approaches such as [28], often rely on temporal information and solve a tracking problem. Many of these approaches parameterize the high dimensional human body by a low-dimensional skeleton embedded in the body model template. The motion reconstruction process is then formulated as a frame-by-frame optimization to deform the skeletal pose [73], the surface geometry [22, 31] or both of these together [82, 55, 33, 57, 94], even combined with shading based surface refinement algorithms [89, 88], to be consistent with the observed multi-view images. The generative strategy is the preferred choice when accurate results are desired. However, they share limitations such as the requirement of a pre-scanned model template and a skeletal-embedded and aligned initial pose, and they struggle to recover from tracking errors.

Recent studies have tried to solve the above limitations to make the motion reconstruction a coherent and fully automatic pipeline. Non-rigid surface registration methods [50, 74, 101] deform the model vertices instead of the skeletal structure, providing an appealing solution for general dynamic scene reconstruction without pre-embedding skeleton. For reconstruction of general dynamic scenes, early reconstruction methods rely on using high-end studio capture environment with tens or hundreds of video cameras [15, 22, 41]. Benefit from the emergence of consumer-level depth camera, a growing number of works strive for the convenient setup along with the real-time volumetric methods, from commodity multi-view [27, **?**, 26] to even lighter single-view solutions [62, 37]. Recently, DynamicFusion [62] was proposed to fuse the geometry information of a non-rigid scene with slow moving motions, which is real-time and totally automatic without the need for any pre-processing. Guo *et al.* [32] performed a high-quality fusion of both geometry and albedo in the same framework and thereby achieved impressive reconstruction results. Yu *et al.* [97] introduced the skeleton prior into the dynamic fusion pipeline to deal with fast human-central motions. However, our work aims at the general dynamic scenes reconstruction like human interacting with objects, with even some unpredictable new objects appearing in the scene. In such scenario, using prior or semantic knowledge like body or skeletal template [20, 28, 82, 8, 24, 55, 33] or the requirement of prescanned models [74, 28, 50, 95, 101, 31, 84] is not applicable. In addition, the main challenge for such non-rigid fusion based reconstruction methods is to handle topological changes. Guo *et al.* [32] adopted collision detection to address open-to-close changes of the topology. Dou *et al.* [27, 26] adopted a key volume strategy to resolve gerneral topology changes, which does not intrinsically address complex topological changes between key frames. Mira *et al.* [72] adopted a displacement vector field to deal with topological changes implicitly without explicitly modeling the topology changes.

Regardless the tremendous progress of dynamic reconstruction schemes, they remain constrained by a fixed and limited capture volume, or entailing extra manual labor to follow performers. Recently, Xu *et al.* [91] used multiple flying cameras to track the moving target. However, their system requires a pre-scanned template, and the reconstruction is accomplished offlinely. More importantly, none of existing dynamic reconstruction schemes pays attention to the adaptive selection of viewpoints during the reconstruction, which is vital since the quality of the reconstruction highly depends on the availability and quality of the input images.

2.3 Active View Planning

Estimation on view point selection problem is well studied in robotics society. NBV based active view planning problem determines new viewpoints for taking sensor measurements to maximize information collection from the current environment, which can date back several decades [16, 3]. Scott *et al.* [70] provided an overview of early approaches while Chen *et al.* [14] provided a survey of more recent work which placed extra emphasis on system setup. More compact literature [13] by Chen *et al.* thoroughly investigated recently techniques in active sensor planning.

Scott *et al.* [70] categorized the NBV algorithms into model-based and non-model-based methods. Model-based methods suppose an approximation of the scene is known as priori [68, 34] or *Google Earth* [42], these priori is hard to widely applicable in complicated practical world scenarios. Non-model based methods use relaxed assumptions about the scene, but require that the NBV must be estimated online based on the gathered data. Scott *et al.* [70] further classified the non-model based methods into volumetric and surface-based. In a surface-based approach, the boundaries of the surface are examined for evaluating the NBV [65, 12, 47]. Kriegel *et al.* [47] determined the viewpoints via a surface trend estimation, and elevated the algorithm [48] to a hybrid method based on contour and surface prediction. However, it is computationally expensive for more complex operations to the surface representation.

On the other hand, volumetric non-model based methods have become popular because they implicitly model the spatial information and facilitate simple visibility operations. NBVs are assessed by ray casting of a pin hole camera model into the partially reconstructed model from the view candidates and evaluated the traversed voxels. Modern volumetric representation *OctMap* [36] provides 3D occupancy grid mapping approach and data structure with a high storage and traversal efficiency. Vasquez-Gomez *et al.* [79] classified the voxels into five different categories by an occupancy and measurement metric. While Stefan *et al.* [40] and Monica *et al.* [60] proposed another definition which separate voxels into three basic types, occupied, free and known. Jonathan *et al.* [19] and Yamauchi *et al.* [92] made a step further counted the special frontier voxels, defined as unknown voxels which border free and occupied space.

Recently, information theorem was introduced in active view planning method. To assess view quality, Information Gain (IG) formulated as information entropy discrepancy is used in several active view planning works. Stefan *et al.* [40] and Jeffrey *et al.* [23] proposed a set of information gain formulations and provided a comprehensive comparison among the volumetric IG metrics for active 3D reconstruction. Specially, Stefan *et al.* [40] conducted thorough evaluation on surface convergence and entropy reduction on five proposed IG formulations with three basic voxel types. While Jonathan *et al.* [19] proposed an adaptable and probabilistic NBV method without making any assumptions on the reconstructed object. A novel IG formulation based on special frontier voxels is proposed which encoded the strong prior that scan target is an object with closed surface.

While aforementioned methods trend to be time consuming and inappropriate for real-time systems, because of the high computation serial processing of ray casting which can be efficiently implemented on parallel computing devices such as GPUs. To exploit parallel computing capability of GPUs, most recent work [60] proposed a contour based method for NBV selection with KinectFusion scheme. Discrete classification among voxels was executed in TSDF representation, while no occupancy probability was used.

CHAPTER 3

ACTIVE HUMAN RECONSTRUCTION

In this section, we will first describe the overview of our ***iHuman3D*** system, followed by the elaboration of two major modules: the active view planning module and the human body reconstruction module, respectively.

3.1 System Overview

Recall that ***iHuman3D*** aims for intelligent human body reconstruction using a single aerial robot. As shown in Fig. 3.1, we adopt a compactly designed aerial robot, equipped with: *NUC* – a mini PC that acts as the brain with computation and control units, *Guidance* [**?**] – armed with an ultrasonic sensor and stereo cameras working as a navigation system, providing the pose estimation using its internal VO algorithm by fusing the IMU data, and *Xtion* – serving as the 3D sensor device to acquire the RGBD data (VGA resolution) of the scene. In particular, the aerial robot works around 2 meters away from the captured dynamic target. Such setting is the compromise between the field of view (FOV) and depth accuracy of the RGBD sensor.

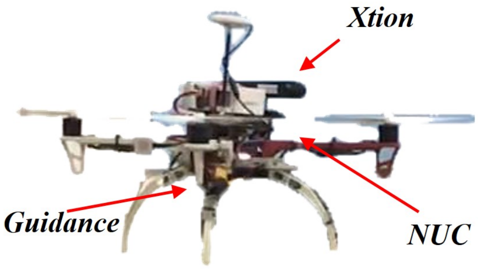


Figure 3.1. Flying camera using in both ***iHuman3D*** and ***FlyCap***

Fig. 3.2 gives a sketch of the working pipeline of ***iHuman3D***. The aerial robot works as a flying camera to capture the depth information in real-time. The captured RGB-D data is streamed via a wireless network connection to a desktop machine that runs our human body reconstruction and active view planning modules. To reduce the required bandwidth for real-time performance, we use data compression based on *zlib* for the depth stream. Then, the view planning result is streamed back to the aerial robot interface via the same network. Both the human body reconstruction and the view planning are performed in a highly parallel way on the modern GPU hardware to enable real-time performance.

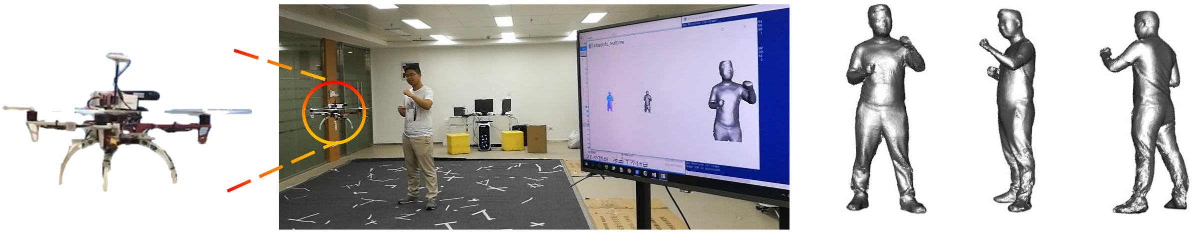


Figure 3.2. Realtime 3D human body reconstruction by a flying camera. Left: aerial robot mounted with a depth camera. Middle: live demo. Right: high quality realistic live mesh.

The system architecture of ***iHuman3D*** is illustrated in Fig. 3.3, which relates to: (i) human body reconstruction module, (ii) active view planning module, and (iii) flying camera module. The reconstruction module fuses the live depth input into the TSDF volumes, and meanwhile provides a real-time mesh visualization result. Based on the real-time live TSDF volume, the active view planning module examines the NBV from all the view candidates in parallel on the modern GPU hardware. For the stability of the whole system, the NBV results are transmitted back to the aerial robot in another fixed frame rate (10fps). In the flying camera module, we use the same robot interface to the hardware platform of the aerial robot as [91], which provides a depth stream with the corresponding camera location information in 30fps. Note that the whole system is synchronized with a common NTP server.

3.2 Active View Planning Module

3.2.1 Occupancy Probability in TSDF Volume

Aiming at realtime next-best-view selection, we follow the pioneer work [60] on information gain calculation in TSDF volume. As defined in [63], two components are stored

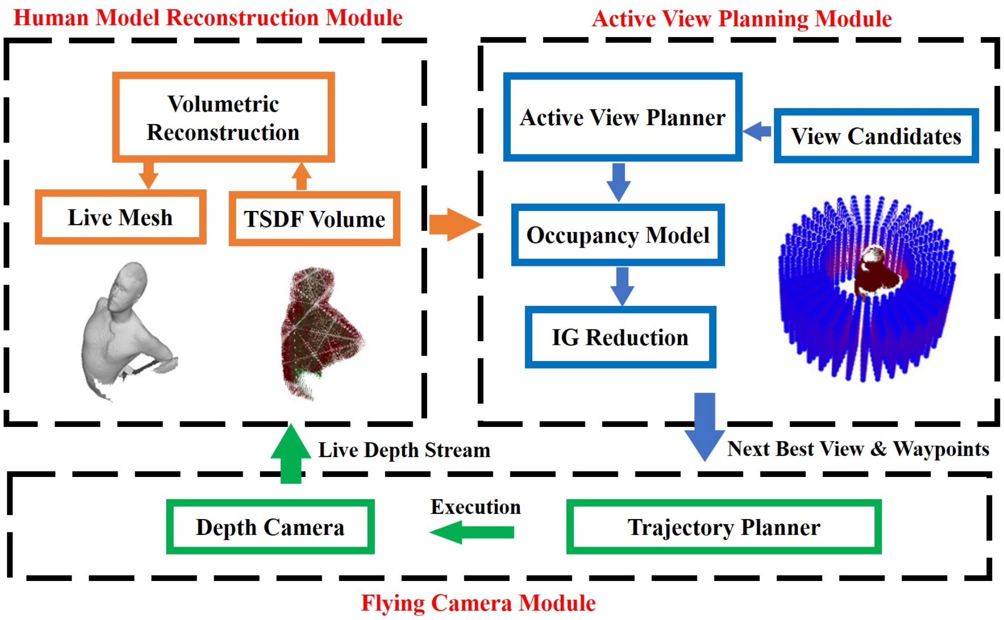


Figure 3.3. The architecture of *iHuman3D*. Three modules consist with *iHuman3D*, flying camera module captures human's depth stream and executes human scanning; human model reconstruction module absorbs depth stream, estimates the TSDF volume and reconstructs human mesh; active view planning model calculate the TSDF based information gain from view candidates and generates next-best-view and waypoints guiding flying camera's motion. Blue dots surrounding the TSDF are view candidates.

in TSDF which represents a fusion of the registered depth measurements from frames $1, \cdots, k$ for each voxel $\mathbf{p} \in \mathbb{R}^3$,

$$\mathbf{S}_k(\mathbf{p}) \mapsto [\mathbf{F}_k(\mathbf{p}), \mathbf{W}_k(\mathbf{p})], \tag{3.1}$$

where $\mathbf{F}_k(\mathbf{p})$ is the truncated distance value and $\mathbf{W}_k(\mathbf{p})$ indicates the measurement weight. For each voxel with a distance r from camera center along depth map ray, the distance from depth value is truncated with a range $\pm\mu$ centered at the measurement.

As illustrated in Fig. 3.4, in TSDF representation, a voxel with high positive value indicates it locates outside from the object surface with high probability to be free, whereas the negative volexs is on the opposite side. Similar as [60], we classify voxels into three sets: unknown set $\mathbb{U}_{1:k}$, occupied set $\mathbb{O}_{1:k}$ and empty set $\mathbb{E}_{1:k}$ according to $\mathbf{F}_k(\mathbf{p})$, $\mathbf{W}_k(\mathbf{p})$ as follows:

$$\begin{cases} \mathbf{W}_k(\mathbf{p}) = 0 & \to \text{unknow voxel}, \mathbf{p} \in \mathbb{U}_k \\ \mathbf{W}_k(\mathbf{p}) > 0, \mathbf{F}_k(\mathbf{p}) = 1 & \to \text{empty voxel}, \mathbf{p} \in \mathbb{E}_k \\ \mathbf{W}_k(\mathbf{p}) > 0, -1 \leqslant \mathbf{F}_k(\mathbf{p}) < 1 & \to \text{occupied voxel}, \mathbf{p} \in \mathbb{O}_k. \end{cases} \tag{3.2}$$

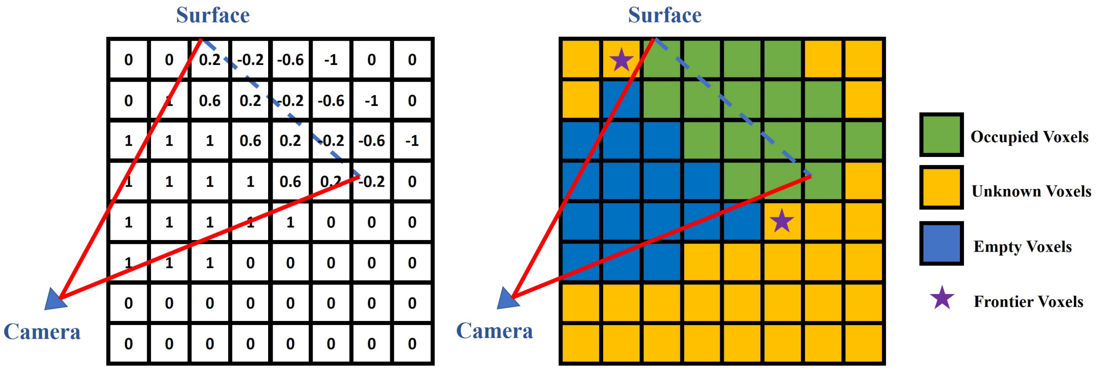
To model occupancy uncertainty for view information gain calculation, we adopt an occupancy probability model, where the occupancy grid mapping integration [61] is used instead of the TSDF update scheme in [63], i.e.,

$$\mathrm{P}(\mathbf{p}|\mathrm{D}_{1:k}) = [1 + \frac{1-\mathrm{P}(\mathbf{p}|\mathrm{D}_k)}{\mathrm{P}(\mathbf{p}|\mathrm{D}_k)}\frac{1-\mathrm{P}(\mathbf{p}|\mathrm{D}_{1:k-1})}{\mathrm{P}(\mathbf{p}|\mathrm{D}_{1:k-1})}\frac{1-\mathrm{P}(\mathbf{p})}{\mathrm{P}(\mathbf{p})}]^{-1}. \tag{3.3}$$

Here $\mathrm{P}(\mathbf{p}|\mathrm{D}_k)$ is the probability given current calibrated depth measurement D_k, $\mathrm{P}(\mathbf{p}|\mathrm{D}_{1:k})$ and $\mathrm{P}(\mathbf{p}|\mathrm{D}_{1:k-1})$ are integrated probability via all previous measurements in k and $k-1$ frame, $\mathrm{P}(\mathbf{p})$ is a prior probability. We assume that the occupancy of $\mathbf{p} \in \mathbb{O}_k$ in current measurement D_k is a normal distribution according to the new TSDF value $\mathbf{F}_{\mathrm{D}_k}(\mathbf{p})$.

$$\mathrm{P}(\mathbf{p}|\mathrm{D}_k) = exp(-\frac{\mathbf{F}_{\mathrm{D}_k}(\mathbf{p})^2}{2\sigma_1^2}). \tag{3.4}$$

Here we set $\sigma_1 = \mu/3$ to force occupancies distribute inside the truncate band mostly.



0	0	0.2	-0.2	-0.6	-1	0	0
0	1	0.6	0.2	-0.2	-0.6	-1	0
1	1	1	0.6	0.2	-0.2	-0.6	-1
1	1	1	1	0.6	0.2	-0.2	0
1	1	1	1	1	0	0	0
1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a)

(b)

Figure 3.4. TSDF representation. (a) Voxels are assigned with a truncated distance value along the camera casting ray. (b) Three basic voxel categories based on TSDF values and weights, occupied voxels (green), unknown voxels (yellow) and empty voxels (blue). Frontier voxels are unknown voxels whos neighbour contains both occupied voxels and empty voxels.

Similar to [36], under the assumption of an uniform prior $\mathrm{P}(\mathbf{p})$ and the usage of log-odds probability notation, Eqn. 3.3 can be simplified as:

$$\mathbf{L}(\mathbf{p}|\mathrm{D}_{1:k}) = \mathbf{L}(\mathbf{p}|\mathrm{D}_{1:k-1}) + \mathbf{L}(\mathbf{p}|\mathrm{D}_k). \tag{3.5}$$

3.2.2 View Information Gain

Given the basic voxel category in Fig. 3.4, similar to [60][19], the frontier voxels denoted as $\mathbf{f}_i \in \mathbb{F}_{1:k}$ are considered as the unknown voxels which border both empty voxels and occupied voxels. Note that these frontier voxels are near the boundary of the estimated human model, thus we assume the unknown voxels $\mathbf{p} \in \mathbb{U}_{1:k}$ near the frontier voxels may have a high probability to belong to the estimated human model. We then formulate the frontier information as:

$$Q(\mathbf{p}) = \max_{\mathbf{f}_i \in \mathbb{F}_{1:k}} exp(\frac{\|\mathbf{p} - \mathbf{f}_i\|_2^2}{-2\sigma_2^2}), \tag{3.6}$$

where σ_2 is set to be the same as the truncated band μ empirically.

The volumetric information from virtual view D_{k+1} is defined

$$\mathbf{I_p}(\mathbf{p}, \mathbf{r}) = \mathrm{Entropy}(\mathbf{p}) Q(\mathbf{p}) \prod_j^{m-1} [1 - P(\mathbf{p}_j | D_{1:k})], \tag{3.7}$$

where $\{\mathbf{p}_j, j = 0, ..., m-1\}$ are all voxels traversed along a ray $\mathbf{r}$ before hitting the voxel $\mathbf{p}$, and $\prod_j^{m-1} [1 - P(\mathbf{p}_j | D_{1:k})]$ indicates the visibility of $\mathbf{p}$. $\mathrm{Entropy}(\mathbf{p})$ is the entropy of $\mathbf{p}$ related to Q as follows:

$$\mathrm{Entropy}(\mathbf{p}) = -Q(\mathbf{p}) \ln Q(\mathbf{p}) - [1 - Q(\mathbf{p})] \ln [1 - Q(\mathbf{p})]. \tag{3.8}$$

Finally, the total view information of the virtual view D_{k+1} can be formulated as:

$$\mathbf{I_v} = \sum_l^L \sum_i^I \mathbf{I_p}(\mathbf{p}_{l,i}, \mathbf{r}_l), \tag{3.9}$$

where $\mathbf{r}_l$ is all possible casting ray of current view candidate $\mathbf{v}$ and $\mathbf{p}_{l,i}$ is all voxels casted through by a ray $\mathbf{r}_l$ before hitting on surface or volume boundaries. Focusing on the object-centric reconstruction tasks, we model the candidate view search space as a series of cylinder around the maintaining TSDF volume center, parameterized by $\mathbf{v} = (r, \theta, l)$ with all candidate views $\mathbb{V}$ pointing to object center.

For the evaluation of the information gain (IG) of all the candidate viewpoints through ray casting operation, we make use of the modern GPU hardware to achieve real-time implementation. The observation here is that all the candidate viewpoints and all the casted rays are independent to each other. So that different candidate viewpoints can be attached to different blocks in the GPU, while a thread in the block is related to a small batch of rays. In our setting, each $\mathrm{R}(\mathbf{v})$ is measured on 64 × 64 resolution, so each block in the GPU has 1024 threads and each thread casts 4 rays to the volume. After calculating the IG for such each 4 rays, the evaluation about all the candidate viewpoints is to perform intra-block sum reduction operation, which can be done efficiently on the GPU using the share memory and the warp reduction operation, as shown in the Algorithm 1.

Algorithm 1 Algorithm for IG reduction

```
Input: TSDF volume, $\mathbf{T}_{view}[n]$,
Output: $\mathbf{IG}_{view}[n]$
    Initialisation : prepare 4096 ray directions, $\mathbf{Ray}[4096]$.
    Block-wise LOOP Process in parallel
 1: for $i = 0$ to $n - 1$ do
 2:     $\mathbf{T}_{curr} = \mathbf{T}_{view}[i]$.
        Thread-wise LOOP Process in parallel
 3:     for $j = 0$ to 1023 do
 4:         allocate share memory $\mathbf{SEM}[32]$.
 5:         $\mathbf{IG}_{4ray} = 0$.
            LOOP Process of the ray-batch
 6:         for $rayIdx = 0$ to 3 do
 7:             $\mathbf{Ray}_{curr} = \mathbf{Ray}[j + 1024 \times rayIdx]$ .
 8:             Find $voxelIdx$ by using 3D Bresenham to rasterize $\mathbf{Ray}_{curr}$ and $\mathbf{T}_{curr}$
 9:             calcute $\mathbf{ig}$ in the $voxelIdx$.
10:             $\mathbf{IG}_{4ray} += \mathbf{I}_{\mathbf{p}}(\mathbf{p}, \mathbf{r})$.
11:         $warpid = tid >> 5$
12:         $laneid = tid \& 31$
            warp reduction for $\mathbf{IG}_{4ray}$
13:         $reducedValue = \mathbf{IG}_{4ray}$
14:         $\mathbf{SEM}[warpid] = warpReduct(reducedValue)$.
            share memory reduction for $\mathbf{IG}_{view}[i]$
15:         $reducedValue = \mathbf{SEM}[laneid]$
16:         $\mathbf{IG}_{view}[i] = warpReduct(reducedValue)$.
```

Benefit from the efficient parallel computation, the proposed method brings a huge lift of speed on NBV calculation. Moreover, it provides a general framework for real-time IG reduction, as long as the computational complexity of information-based function in step. 9 of Algorithm 1 equals to or is less than O(N).

3.2.3 Next Best View Scanning

We select the next-best-view by optimizing the following energy function,

$$\mathbf{v}^{\star} = \arg\max_{\mathbf{v}} \lambda_I \mathbf{I_v} + \lambda_C \mathbf{C_v} + \lambda_S \mathbf{S_v}, \tag{3.10}$$

where $\mathbf{I_v}$ is the view information calculated by Eqn. 3.9, $\mathbf{C_v}$ is the movement cost term which penalties views that need large robot movement from current position, $\mathbf{S_v}$ is the trajectory smoothness term which encourages the view points lying on current moving direction, and $\lambda_I, \lambda_C, \lambda_S$ are the corresponding coefficients.

The main challenge in human-centric scanning is that the human body may suffer from slightly non-rigid movement during scanning. As studied in [91], the non-rigid deformation of human body and the rigid motion of the aerial robot are coupled together. To turn the selected NBV in virtual view space in Eqn. 3.9 to the physical world space for aerial robot control, we initialize the camera pose in the volume space, denoted as $\mathbf{T}_{d2v}$, by performing the traditional Rigid-ICP algorithm with the TSDF volume and the live depth image. On the other hand, the camera pose in the world space, denoted as $\mathbf{T}_{d2w}$, can be directly retrieved from the onboard ***Guidance*** [100] module of the aerial robot. And then we can simply get the rigid transformation from the world space to the volume space, denoted as $\mathbf{T}_{w2v}$ as follows:

$$\mathbf{T}_{w2v} = \mathbf{T}_{d2v}(\mathbf{T}_{d2w})^{-1}. \tag{3.11}$$

To guide the movement of flying camera from current position to next-best-view spot, we utilize a quality-driven method to adaptively insert waypoints before reaching predicted spot. Here we consider different robot orientations on smooth trajectory generated via [30]. The angle formed between the camera ray's orientation and the surface normal is expected to be small, so as to guarantee sensing quality of depth camera. We define the quality of virtual depth image D generated by a yaw angle $\phi \in (-\pi/2, \pi/2)$ as:

$$N(D) = \sum_{l}^{L} < \bar{\mathbf{n}}_l, \bar{\mathbf{r}}_l >, \tag{3.12}$$

where $\bar{\mathbf{r}}_l$ is the unit vector with the opposite direction with pixel casting ray from camera principle point and $\bar{\mathbf{n}}_l$ is the unit surface normal. Note that we ignore the casting ray and normal pairs that have negative inner product. To obtain the optimal ϕ, we use the same reduction scheme in 3.2.2 and find the maximum quality view in 18 uniformly sampled candidates. The reconstruction ends when the highest information gain of all NBV candidates is smaller than a user-defined threshold.

3.3 Human Body Reconstruction Module

Our human body reconstruction module follows the conventional volumetric fusion pipeline [63], where the TSDF volume aligns the temporal information of the dense 3D data from depth camera and fuses a human body in real-time. On one hand, the TSDF volume is utilized by the active view planning module as described before. On the other hand, we use the Marching Cube algorithm to generate a mesh for visualization from the TSDF volume.

Moreover, with the observation that the human target always has slight motion during the reconstruction process, a light-weight non-rigid deformation method is adopted when integrating the new depth image into the TSDF volume. Similar to recent work [50, 91], we use the embedded deformation (ED) model to parameterize the non-rigid motion field. Given a reference mesh, the sparse ED nodes are uniformly sampled to cover the overall surface. Let $\mathbf{x}_i$ be the i-th ED node location, which is also associated with a set of parameters to represent the deformation around the ED node. Furthermore, neighboring ED nodes are connected together to form a digraph called ED graph, which is collectively represented by all the deformation parameters and ED node locations on it. Since each mesh vertex is “skinned” with its K neighboring ED nodes (with $K = 4$ in our system), the mesh can be deformed according the given parameters of an ED graph.

Aiming at compactly representing the static human model with slight non-rigid deformation, we use the rigid transformation (6-DOF) in the ED node and apply the Linear Blending Skinning (LBS) method for skinning. Thus, the full parameter set for the deformation is $G = \{\mathbf{T}_i\}$. For a particular mesh vertex $\mathbf{v}_j$, its new position is formulated as

$$\mathbf{v}_j' = \mathrm{ED}(\mathbf{v}_j; G) = \sum_{\mathbf{x}_i} w(\mathbf{v}_j, \mathbf{x}_i)\mathbf{T}_i\mathbf{v}_j, \tag{3.13}$$

where $w(\mathbf{v}_j, \mathbf{x}_i)$ measures the influence of the node $\mathbf{x}_i$ to the vertex $\mathbf{v}_j$. Please refer to

[91] for details about calculating w for all mesh vertices. Note that Eqn. (Eq.3.13) omits the conversion between the 3-vectors and their corresponding homogeneous 4-vectors (as needed for multiplications with $\mathbf{T}_i$) for simplicity of notation.

The data term is then designed to force vertices on the model to move to the corresponding depth point of the input depth data, especially along the norm direction, which can be considered as the first order approximation of the real surface geometry. As in [91], we find the dense depth correspondences between the model and the depth images via a projective lookup method, and discard those pairs with a highly distinct depth value (larger than 20 mm) or normal direction (larger than 20 degrees):

$$E_{\text{fit}}(G) = \sum_{j=1}^{\mathbf{C}} \|\mathbf{n}_{\mathbf{v}_j}^{T}(ED(\mathbf{v}_j; G) - \mathbf{c}_j)\|_2^2, \tag{3.14}$$

where $\mathbf{C}$ denotes all correspondent pairs between mesh vertices (denoted as $\mathbf{v}_j$) and depth points (denoted as $\mathbf{c}_j$) in the depth image captured by the aerial robot. Regarding the regular term that prevents unreasonable local deformation of the model, as we utilize 6-DOF rigid transformation instead of 12-DOF affine transformation, it is formulated as

$$E_{\text{reg}}(G) = \sum_{\mathbf{x}_j} \sum_{\mathbf{x}_i \in N(\mathbf{x}_j)} w(\mathbf{x}_j, \mathbf{x}_i) \|\mathbf{T}_i \mathbf{x}_j - \mathbf{T}_j \mathbf{x}_j\|_2^2, \tag{3.15}$$

where $w(\mathbf{x}_j, \mathbf{x}_i)$ defines the weight associated with the edge in the ED node graph.

Given the energy terms related to $\{\mathbf{T}_i\}$, we minimize them in an iterative closest point (ICP) framework, where dense pairs are updated by the projective lookup method. In each ICP iteration, the energy above can be rewritten as a sum of squares. In this form, the minimization problem can be seen as a standard sparse non-linear least-squares problem, which can be solved efficiently using the Gauss-Newton method. When performing Gauss-Newton optimization, we adopt the Taylor expansion of the exponential map around current estimated camera poses by introducing small Lie algebra parameters $\xi = (\nu^T, \omega^T)^T$. For compacting the non-rigid deformation to fit the slight motion assumption of our human body reconstruction module and to achieve real-time performance, the number of the ED nodes is restricted so that the ED graph is roughly covered the entire mesh. In our system, the number of al the ED nodes is around 100.

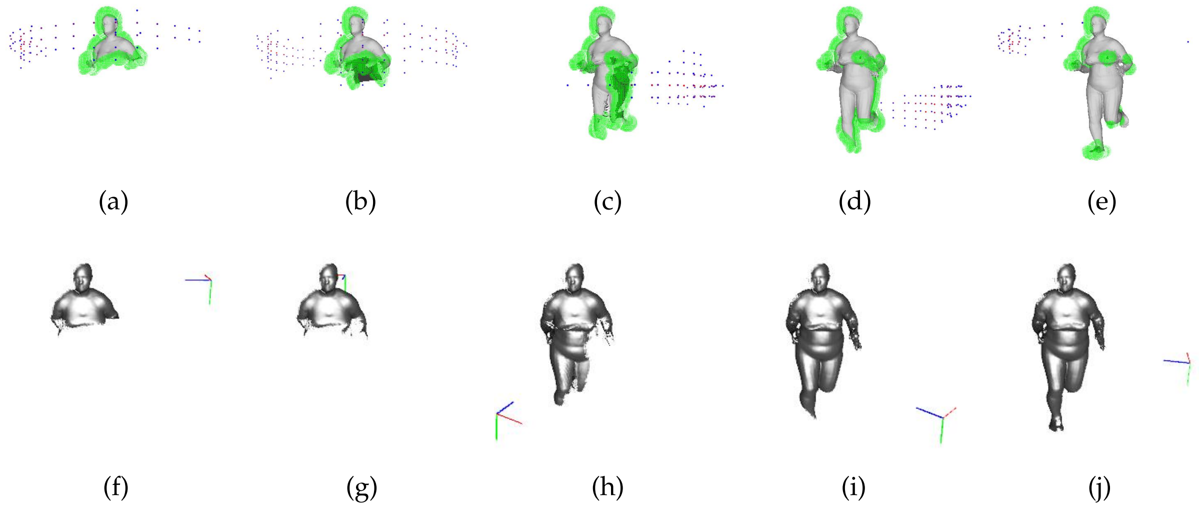


Figure 3.5. Based on partially reconstructed model(gray mesh), frontier information volume(green dots) is predicted. Top view candidates are colorized according to score. Red represents high scoring, while blue is relative low score view points. (a)-(e) represent $\{2, 3, 4, 6, 9\}\mathrm{th}$ NBV iteration respectively. (f)-(g) represent canonical model and camera pose in each NBV iteration.

3.4 Experimental Results

In this section, we first illustrate the computational efficiency of the proposed NBV method, and then experiments on the ***iHuman3D*** system using both the synthetic data and real-time human scanning data are conducted respectively. We highly recommend readers to refer supplementary and the video for more implementation details and more comprehensive results.

3.4.1 Computational Efficiency

For human-centric 3D reconstruction, we maintain a $2m \times 2m \times 2m$ TSDF volume with a $256 \times 256 \times 256$ voxel resolution. Empirically, the searching space of $\mathbf{v}$ is restricted by $\{\mathbf{v} | r \in [1, 1.5], \theta \in [-\pi, \pi), l \in [-1, 1]\}$. Jointly considering the depth measurement range and robot maneuverability, 4608 view candidates are uniformly sampled surrounding the volume center to sufficiently represent all possible views.

The efficiency of proposed next-best-view method is assessed. We implemented Algorithm 1 on NVIDIA GeForce GTX1080 using CUDA, and it takes about 30 ms to calculate the IGs from all the 4608 candidate viewpoints. As shown in Table 3.1, compared to 8s for 88 candidate viewpoints using Isler's methods [40], denoted as *Isler*, our method evaluates viewpoints on the order of 1.0×10^5 faster. The speed of evaluating viewpoints in our scheme is comparable with the *APORA* proposed in [19]. Note that the *APORA* still takes 12s to exhaustively evaluate all the candidate viewpoints during a NBV iteration, while the proposed Algorithm 1 enables real-time active view planning. Specifically, for each NBV iteration, our method achieves 200x speed up compared with *APORA* and *Isler*.

Table 3.1. Computational speed for evaluating candidate viewpoints

	Isler [40]	*APORA* [19]	***iHuman3D***
Average Number of Viewpoints	88	1.5×10^6	4608
Average Time	8 s	12 s	30 ms
Average Views/Second	11	1.3×10^5	1.5×10^5

3.4.2 Simulation Platform and Evaluation on Synthetic Data

To better assess ***iHuman3D*** system, we built a simulation platform to evaluate and visualize the next-best-view selection to improve the efficiency of our evaluations on the proposed active view planning and human body reconstruction algorithms. Based on our system architecture in Fig.3.3, we replaced the flying camera module with a simulation platform, which simulates the maneuverability of the physical aerial robot [30] and generates synthetic depth streams. We utilized the recent work SURREAL [77] which embedded Human3.6M [39] pose skeletons into various human SMPL [58] models and the render engine *Blender* to render synthetic depth streams with the ASUS Xtion camera intrinsic parameters.

To evaluate the proposed NBV guided human model reconstruction method, in Fig.3.5, we visualized the reconstructed mesh, camera position, frontier information volume and top view candidates together in the selected NBV iterations. Guided by the frontier information, ***iHuman3D*** automatically picks out the views which quickly fill the unobserved area, as shown in Fig.3.5 (c). Note that the robots motion regularization term and trajectory smoothness term ensure the picked view candidates to be close to current camera position and consistent to the direction of the moving robot, as shown in Fig.3.5 (b, c).

We further compare our method with the most related work ***FlyCap*** [91] qualitatively and quantitatively. ***FlyCap*** scans a human body for 3 circles with a fixed spiral-down trajectory, with an off-line human reconstruction algorithm. Focusing on evaluating different view planning method, for fair comparision, we use the same reconstruction module by feeding ***FlyCap*** a synthetic depth stream with the pre-defined spiral-down trajectory. The qualitative comparison of the reconstruction results is provided in Fig.3.6, which indicates the proposed method has a better overall quality especially in the tough cases emphasized by the red circles. Fig.3.7 provides the quantitative per-vertex reconstruction error compared to the ground-truth synthetic model. Our method achieves 14.86 mm average per-vetex error, compared to 20.25 mm of ***FlyCap***. To further analyze the accuracy improvement, we compare with the ***iHuman3D*** without the quality-driven waypoints selection, which achieves 18.13 mm average per-vetex error as shown in Fig. 3.7 (c). These results illustrate the superiority of our NBV guided method in terms of reconstruction.

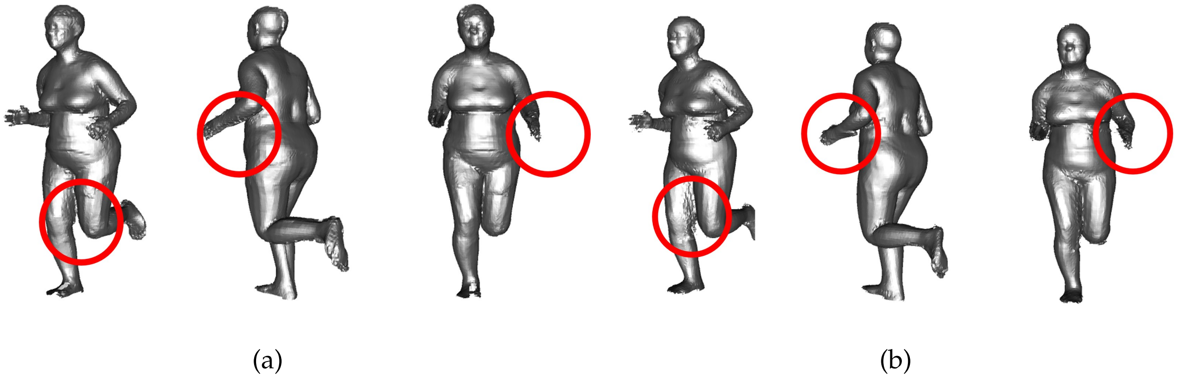


Figure 3.6. Quality comparison between (a) ***iHuman3D*** and (b) ***FlyCap***.

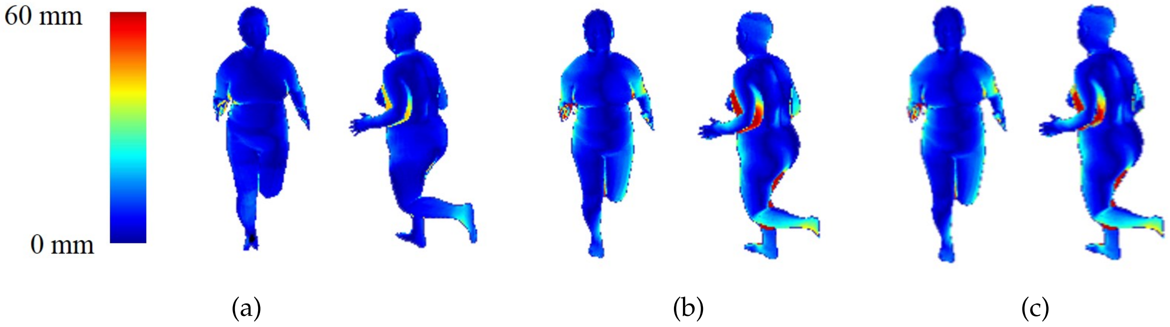


Figure 3.7. Error map compared to groundtruth. (a) ***iHuman3D*** (b) ***FlyCap*** (c) ***iHuman3D*** without quality-driven waypoints.

To evaluate the reconstruction efficiency, we compared the online generated trajectory of ***iHuman3D*** with the one of ***FlyCap***. As shown in Fig.3.8 (a), our method can finish the scanning task more quickly, leading to slower robot motion increase and superior robot efficiency. To evaluate reconstruction efficiency, the vertices convergence property is considered as shown in Fig.3.8 (b). It indicates that the proposed method can effectively guide human reconstruction. We argue that in conventional scanning methods, it is hard to model the observation overlaps between scan fragments, due to robot localization error and rigid ICP error caused by slight non-rigid movement of the human model. Whereas, in our ***iHuman3D*** with frontier information guided NBV selection, the robot adaptively moves to spots which can complete the model more efficiently.

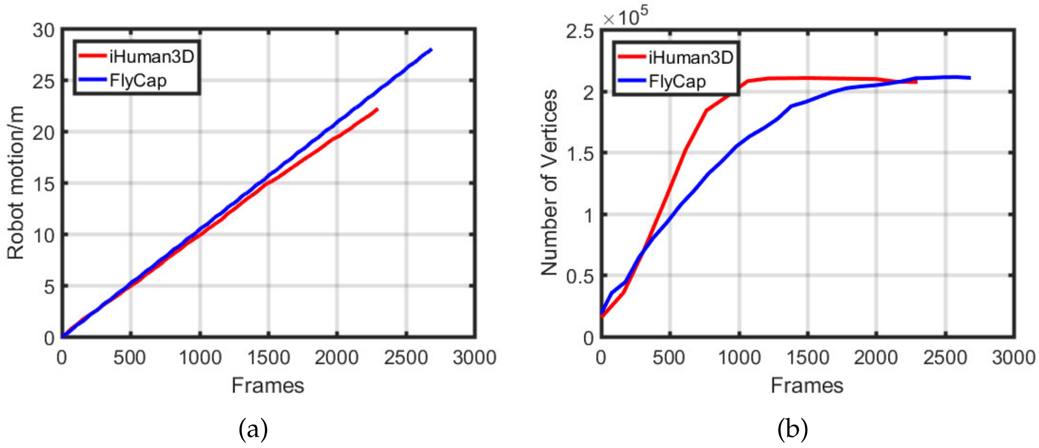


Figure 3.8. Efficiency evaluation. (a) Robot motion according to reconstruction frames. (b) Mesh vertices increase according to frames.

3.4.3 Real World Experiments

In this subsection, we evaluate the ***iHuman3D*** system in the practical scenarios as shown in Fig.3.9. As explained in Sec 3.1, a flying camera is used to scan a target human, while a desktop machine executes online reconstruction and active view planning. Given a manually selected initial pose, the flying camera will automatically scan the human model until the task is completed. The Flying camera and the desktop machine exchange data via wireless network, while the live output mesh and canonical model are displayed on a screen in real-time.

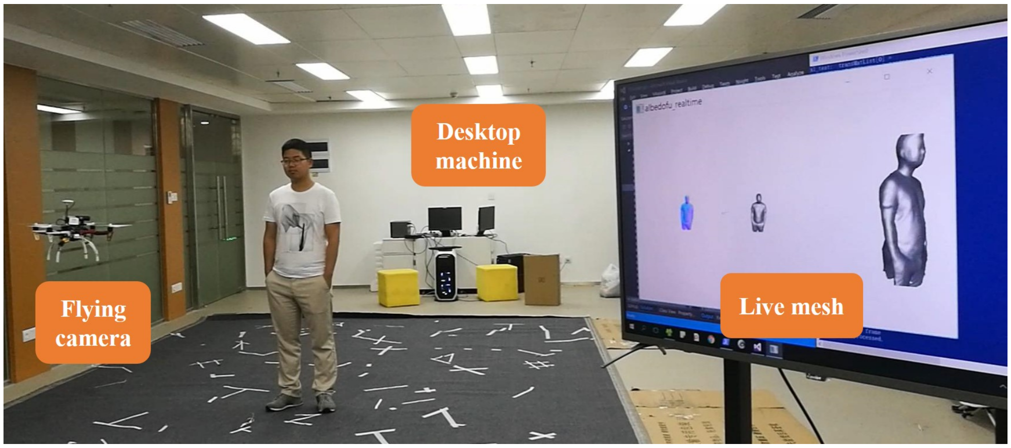


Figure 3.9. System setup in realtime human reconstruction.

Two reconstructed human models of the practical scenarios is provided in Fig.3.10 rendered from different views, which obtain considerably high quality results, even with noisy depth input and limited volume resolution for real-time purpose. Note that the experiment is hard to be conducted without any external sensor like Vicon, since the aerial robot cannot receives GPS signals indoor and only the on-board visual odometry module helps for the robot localization. The initial location observations observed from the flying camera is poor and a naive 3D reconstruction might fail. Thanks to the proposed NBV guided human model reconstruction strategy, our method achieves relatively high accuracy and is robust to the noise of the robot localization module.

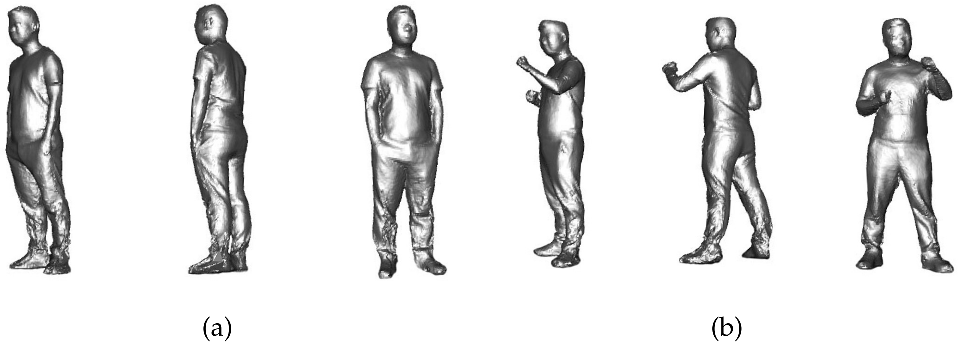


Figure 3.10. Two different human model reconstructed in realtime experiments, (a) Standing still and (b) Punching.

CHAPTER 4

ACTIVE HUMAN MOTION CAPTURE

4.1 Active View Analysis

In this section, a novel active view analysis metric for dynamic scene is introduced which can estimate better view selection to guide the dynamic reconstruction (Sec. 4.1.2). Based on this novel metric, we also describe a real-time active view planning approach for non-rigid scenes (Sec.4.1.3). Technically, we discretize the nearby space of current flying camera into a view space, where each sample can be the view candidate that the flying camera may reach in the next stage. To evaluate the effectiveness of each view candidate, we propose a novel Geometry And Motion Energy (GAME) metric, including three terms computed by raycasting the reconstructed volume with the dynamic scene. To enable the real-time view planning of the flying camera, we propose an efficient GPU-based hierarchical searching scheme to optimize the GAME metric in the entire view space.

4.1.1 Notation

We define the coordinate frame of the volume as $\mathbb{V}$ and the local coordinate frame of the flying camera as $\mathbb{B}$. Given a input depth image, a rigid transformation $\mathbf{T}_c$ is factored out from the non-rigid motions to transform a vertex or a voxel from $\mathbb{V}$ to $\mathbb{B}$. We discretize the nearby space of the flying camera to generate a view space $\mathcal{V}$. Considering the maneuverability of flying camera, each view candidate is parameterized by a 4-DOF vector $v = (x, y, z, \theta)$, where $\{x, y, z\}$ is the translation relative to the flying camera pose, θ is the pitch angle, and the roll and yaw angles are assumed to be zero. Thus, the camera pose of the view candidate $v \in \mathcal{V}$ is denoted as $\mathbf{T}_v$:

$$\mathbf{T}_v = \begin{bmatrix} \cos\theta & 0 & \sin\theta & x \\ 0 & 1 & 0 & y \\ -\sin\theta & 0 & \cos\theta & z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.1}$$

Additionally, we represent the set of rays casted from the view candidate ν into the volume space $\mathbb{V}$ as $\mathcal{R}_\nu$. We also define $\{\vec{x}_i | i = 0 \dots n\}$ as the set of traversed voxels coordinates along a cast ray in the data volume before hitting the surface, where $\vec{x}_n$ represents the hit voxel.

4.1.2 Geometry And Motion Energy (GAME) Metric

To analyze the effectiveness of a capture view for dynamic scene reconstruction, we propose a novel Geometry And Motion Energy (GAME) metric, which simultaneously considers the quality of captured depth, the central area size of observation, the geometry quality and motion energy for a view candidate, denoted as *depth term*, *center term* and *motion term*, respectively. Note that the GAME metric applies on the immediate reconstructed geometry stored as TSDF in the data volume, the motion field G, the camera pose in volume space $\mathbf{T}_c$, and a view candidate ν with camera pose $\mathbf{T}_\nu$, jointly.

Depth term: As the distance of the scene influences the captured depth accuracy directly, the proposed depth term aims to encourage the captured geometry to be nearby an optimal distance for dynamic reconstruction. Mathematically, the average distance of the scene is defined as:

$$d_{avg}(\nu) = \frac{1}{M} \sum_{\forall r \in \mathcal{R}_\nu} [\mathbf{T}_\nu \mathbf{T}_c \vec{x}_n]_z, \tag{4.2}$$

where $\vec{x}_n$ is the voxel coordinates in the surface hit by the ray r, $[.]_z$ returns the distance along the optical axis, and M is the number of all the valid rays hitting the surface. Then the depth term is formulated as:

$$E_{depth}(\nu) = \psi_{dist}(d_{avg}(\nu) - d_o), \tag{4.3}$$

where d_o is the reference of capture distance (e.g., $d_o = 1.0m$ for the ASUS Xtion depth camera used in our implementation) and ψ_{dist} serves as a penalty to avoid large distance error for safety issue of drone control:

$$\psi_{dist}(x) = \begin{cases} \frac{1}{1+\kappa x^2} & \text{if } |x| < d_{thres}, \\ -\infty & \text{if } |x| \geqslant d_{thres}, \end{cases} \tag{4.4}$$

where d_{thres} is the threshold allowing for certain error in distance control. We have $d_{thres} = 0.30m$ and the penalty scale $\kappa = 20$ in our implementation.

Center term: The proposed center term aims to ensure that the target is always in the center of the captured area, so as to maximize the valid data usage for dynamic reconstruction. To calculate the centrality of each ray, we define the projected pixel offset of the hit voxel $\vec{x}_n$ of the ray r in relative to the central pixel coordinates of the image as:

$$\begin{bmatrix} du \\ dv \end{bmatrix} = \pi(\mathbf{K}(\mathbf{T}_v \mathbf{T}_c \vec{x}_n)) - \begin{bmatrix} W/2 \\ H/2 \end{bmatrix}, \tag{4.5}$$

where $\mathbf{K}(\cdot)$ is the projection function of the given camera intrinsics, $\pi(\cdot)$ is the perspective division after projection, and (W, H) is the size of projected image. Then the centrality of the ray r is formulated as

$$\psi_{cen}(r) = \frac{1}{\lambda + du^2} + \frac{1}{\lambda + dv^2}, \tag{4.6}$$

where the damping factor $\lambda = 500$ in our implementation. Finally, we accumulate the centrality of all the valid rays to produce the center term as:

$$E_{center}(v) = \sum_{\forall r \in \mathcal{R}_v} \psi_{cen}(r). \tag{4.7}$$

Motion term: Apart from the distance and centrality to guarantee the capture of valid contents, another critical feature of dynamic scene reconstruction lies in its time-varying geometry and motion. The proposed motion term then measures the geometric quality and motion energy via accumulating the motion information of a valid casted ray r, to encourage the hit local geometry to face to the view candidate, which can be formulated as follows:

$$\psi_{mot_1}(r) = \|1 + \mathbf{n}_{x_n}^T \mathbf{d}(\mathbf{T}_c^{-1} \mathbf{t}_v, \vec{x}_n)\|_2^2, \tag{4.8}$$

where $\mathbf{n}_{x_n}$ is the normal of the hit voxel $\vec{x}_n$ extracted from the TSDF volume, $\mathbf{t}_v = [x, y, z, 1]^T$ is the translation part of $\mathbf{T}_v$, and $\mathbf{d}(\vec{v}, \vec{x}_n) = (\vec{v} - \vec{x}_n)/\|\vec{v} - \vec{x}_n\|$ is the direction vector in the volume space. Furthermore, we accumulate the motion information from the motion field by projecting the motion of ED nodes into current view candidate. The motion effectiveness for an ED node is represented by:

$$\psi_{mot_2}(\mathbf{x}_i) = \|\pi(\mathbf{T}_v \mathbf{T}_c (\mathbf{x}'_i - \mathbf{x}_i))\|_2^2, \tag{4.9}$$

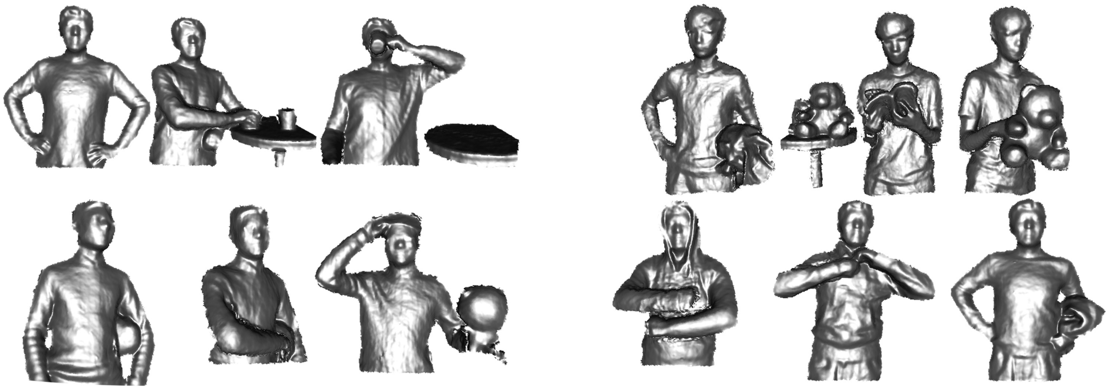


Figure 4.1. Reconstruction results of ***FlyFusion*** on *Drinking*, *Reading*, *Circling* and *Rolling fit* from the upper left to lower right.

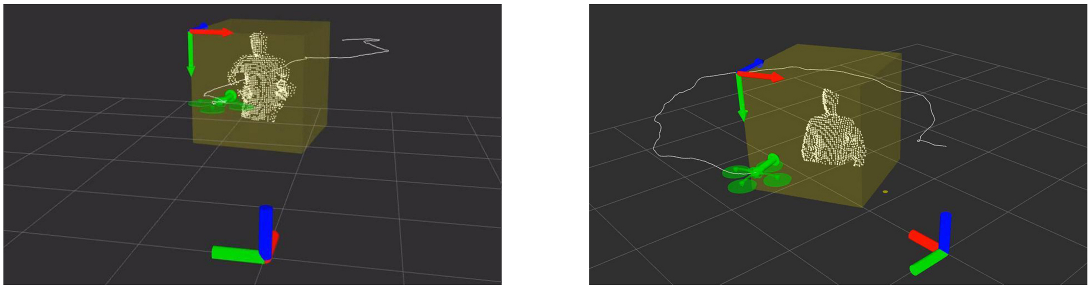


Figure 4.2. View planning examples in global view for the *Rolling fit* and *Circling* sequences respectively.

where $\mathbf{x}_i'$ is the warped node deformed by the motion field G. Our final motion term combines the above two kinds of motion effectiveness for current view candidate, i.e.,

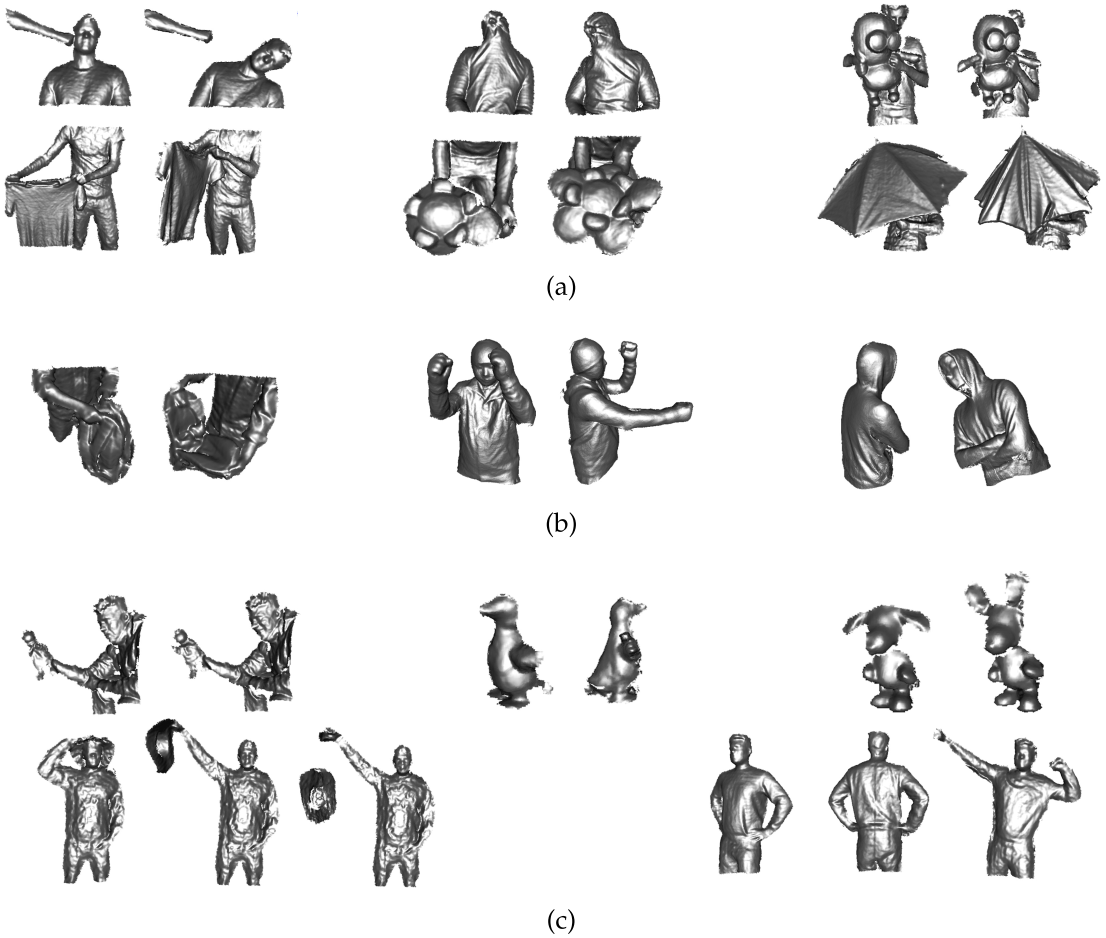
$$E_{\text{motion}}(v) = \sum_{\forall r \in \mathcal{R}_v} \psi_{\text{mot}_1}(r) + \sum_{\mathbf{x}_i} \psi_{\text{mot}_2}(x_i). \tag{4.10}$$

In conclusion, combing the above depth, center and motion terms to jointly evaluate the captured depth quality, observed central area size and motion quality, every view candidate v can be associated with the GAME metric:

$$E(v) = E_{\text{depth}}(v) + \lambda_{\text{cen}} E_{\text{center}}(v) + \lambda_{\text{mot}} E_{\text{motion}}(v). \tag{4.11}$$

4.1.3 Hierarchical Energy Minimization

In this subsection, we propose a active view planning method to illustrate that the GAME metric can be applied to guide the dynamic reconstruction interactively. Given the GAME



(a)

(b)

(c)

Figure 4.3. Evaluation on available single view RGB-D datasets. (a) Our results on the sequences *boxing*, *hoodie*, *minion*, *roll shirt*, *sun flower* and *umbrella* from VolumeDeform [72]. (b) Our results on the sequences *bag open*, *boxing* and *fast loop* from MonoFVV [32]. (c) Our results on the sequences *frog*, *duck*, *snoopy*, *hat* and *Alex* from KillingFusion [72].

metric explained in Sec. 4.1.2, the active view planning problem to select an optimal view candidate v^* can be modeled as:

$$v^* = \arg\max_{v \in \mathcal{V}} \omega(v)\mathrm{E}(v). \tag{4.12}$$

The weight above is formulated as $\omega(v) = exp(-\|\mathrm{d}_v\|^2/(2\sigma^2))$, where σ is a predefined parameter (0.5) and d_v denotes the Euclidean distance from the view candidate v to current localization of the flying platform. Unfortunately, Eqn. 4.12 is non-convex and even non-differentiable due to the ray casting operation. While using modern GPU hardware to traverse the view space in parallel seems applicable since the view candidates are independent for calculating the GAME metric, the tremendous computations required after dense discretization of the view space for stable control of the flying camera inhibit the optimization practically. Specifically, the neighboring 1.0^3 m^3 space of current flying camera space $\mathbb{B}$ is discretized into 64^3 view voxels along the three XYZ axes of $\mathbb{B}$. In each view voxel, the pitch angle θ is discretized into 32 discrete values uniformly ranging from -45 degrees to 45 degrees. Thus the size of whole view space $\mathcal{V}$ is $64^3 \times 32 = 2^{29}$, which is too large to traverse to find the optimal view.

To enable real-time active view planning, we develop a two-stage hierarchical scheme to search the view space $\mathcal{V}$ for the optimization of Eqn. 4.12 in several milliseconds.

Stage-I hierarchical search: As the translation (x, y, z) and the pitch angle θ of a view candidate v are naturally separated for the control of the flying camera, the traversing of $\mathcal{V}$ is then split into two subproblems of traversing two smaller subspace($\mathcal{V}_1$ and $\mathcal{V}_2$) to obtain the optimal rotation and translation iteratively. During each iteration, for the rotation, an arc acrossing current optimal view is built, with a radius of the optimal capture distance d_o and a angle of 90 degree respectively. The arc is discretized into 8 uniform locations with 32 uniform pitch angles towards the center to build a subspace of 256 views, denoted as $\mathcal{V}_1$. For the translation, the pitch angle θ is fixed and the neighboring 1.0^3 m^3 space of current optimal view is discretized into 64^3 regular view candidates to form a subspace, denoted as $\mathcal{V}_2$. To solve Eqn. 4.12, we iteratively traverse the two subspace $\mathcal{V}_1$ and $\mathcal{V}_2$ using GAME metric in parallel. The optimal view candidate is initialized with an identity pose and updated during each iteration.

Stage-II hierarchical search: Traversing the 64^3 subspace $\mathcal{V}_2$ is still too heavy for real-time active view planning. Since $\mathcal{V}_2$ is regular with a fixed pitch angle, we propose to use two cascade 8^3 subspace of $\mathcal{V}_2$ from coarse to fine, denoted as $\mathcal{V}_{21}$ and $\mathcal{V}_{22}$ respectively. The top three view candidates selected via GAME metric in the coarse pace $\mathcal{V}_{21}$ is further discretized in the fine space $\mathcal{V}_{22}$ to determine the final view candidate.

With the two-stage hierarchical search scheme, we traverse the subspace $\mathcal{V}_1$, $\mathcal{V}_{21}$ or $\mathcal{V}_{22}$ in parallel with modern GPU, by associating each view candidate with a CUDA block and performing ray casting for each view independently using 1024 CUDA threads. The final optimal view candidate v^* is obtained by the warp-reduction operation in the GPU. After obtaining v^*, we use the PID-based drone control strategy in [91] for the execution of the flying camera towards v^*.

4.2 Robust Dynamic Fusion

In this section, a robust dynamic fusion scheme is proposed under the single-view and template-less setting, which provides valid information of the scene for the active view analysis. Specifically, to enable real-time performance the reconstruction scheme adopts a temporal fusion strategy based on the the embedded deformation (ED) model and the key volume setting [27]. TSDF [18] is the underlying data structure and a motion field is estimated to align the key volume to the data frame (Sec. 4.2.1). Furthermore, a novel topology compactness strategy is proposed for explicitly regularizing the motion in the areas where topology changes occur (Sec. 4.2.2). We also analyze the "volume drifting" phenomenon to break the fixed capture volume constraint under the moving camera setting (Sec. 4.2.3).

4.2.1 Non-rigid Deformation

Similar to recent work [50, 62, 32], the deformation is represented by a motion field G, consisting of the rigid transformations $\{\mathbf{T}_i\}$ of sparsely sampled nodes $\{\mathbf{x}_i\}$ (ED nodes). The data term is formulated as the sum of point-to-plane distances:

$$E_{\text{data}}(G) = \sum_{(\mathbf{v}_j,\mathbf{c}_j)\in\mathcal{C}} \|\mathbf{n}_{\mathbf{c}_j}^T(\mathbf{v}'_j - \mathbf{c}_j)\|_2, \tag{4.13}$$

where $\mathbf{v}_j$ is a vertex sampled on the fused surface and non-rigidly transformed to $\mathbf{v}'_j$ by this Linear Blending Skinning (LBS) formula $\mathbf{v}'_j = \mathbf{T}_c \sum_{i\in\mathcal{N}(i)} \omega(\mathbf{v}_j, \mathbf{x}_i)\mathbf{T}_i\mathbf{v}_j$, where $\mathbf{T}_i$ is the $\mathbf{SE}(3)$ of each node; $\mathbf{T}_c$ is the rigid component of the entire scene; $\omega(\mathbf{v}_j, \mathbf{x}_i)$ is the skinning weight. Please refer to [62, 50] for more details. $\mathcal{C}$ denotes the set of correspondences found via a projective local search [62, 91].

To predict motions in invisible regions and prevent overfitting to the noise of the depth input, we adopt a locally as-rigid-as-possible motion regularization:

$$E_{\text{reg}}(G)=\sum_{\mathbf{x}_j}\sum_{\mathbf{x}_i\in N(\mathbf{x}_j)} w(\mathbf{x}_j,\mathbf{x}_i)\psi_{\text{reg}}(\|\mathbf{T}_i\mathbf{x}_j-\mathbf{T}_j\mathbf{x}_j\|_2^2), \tag{4.14}$$

where $w(\mathbf{x}_j, \mathbf{x}_i)$ defines the weight associated with the edge in the ED node graph, and ψ_{reg} is the the discontinuity preserving Huber penalty.

The above data term and regular term can be solved efficiently in realtime using the Levenberg-Marquardt (LM) method within the ICP framework. We use rigid ICP to estimate rigid component, followed by a non-rigid ICP to solve for local motions [91]. After estimating the local non-rigid motions and the rigid component, we use the strategy proposed in [27] to apply the local non-rigid motions to warp the key volume into the data volume and then blend current depth input into the data volume using the rigid component. The key volume is updated by non-rigidly integrating TDSF as in DynamicFusion [62]. When blending data volume or updating key volume, we discard the data of the voxels which are warped into invalid area in current depth input. In addition, the key volume is reset to the data volume periodically (every 50 frames in our implementation).

4.2.2 Topology Compactness

The key-volume strategy from [27] tackles topology change between key frames by degrading to the noisier depth input in regions of tracking failure, which does not address the problem intrinsically. For example (Fig. 4.4(a)), the performer's face is connected with the object in the canonical frame, thus the embedded graph connecting these two regions cannot generate correct deformation. Generally, topological changes can be categorized into the open-to-close surface merging and close-to-open surface splitting. While the former surface merging can be solved by handling voxel collision during data fusion [32], the latter surface splitting is still unresolved by previous works [32, 27]. Essentially, the smooth term in Eqn. 4.14 prevents two connected nodes from separation. Moreover, due to the LBS formula in Eqn. 4.13, a vertex in the splitting area is skinned to the separating nodes, producing inevitable artifacts (Fig. 4.4(b)). To handle the topology changes more robustly, especially between key frames, we propose a new strategy including the node-to-node and vertex-to-node compactness.

Node-to-node compactness: Apart from building a ED graph in the key volume in the Euclidean space as in [62, 27, 32] we warp the nodes to build a graph with the same topology in the data volume using the estimated local motions in Sec. 4.2.1. Since the edges of the deformation graph should only be close to the surface in both volumes, we rasterize each edge r into voxels $\mathcal{X}_r$ in both the key and data volumes to calculate its distance to the surface as follows:

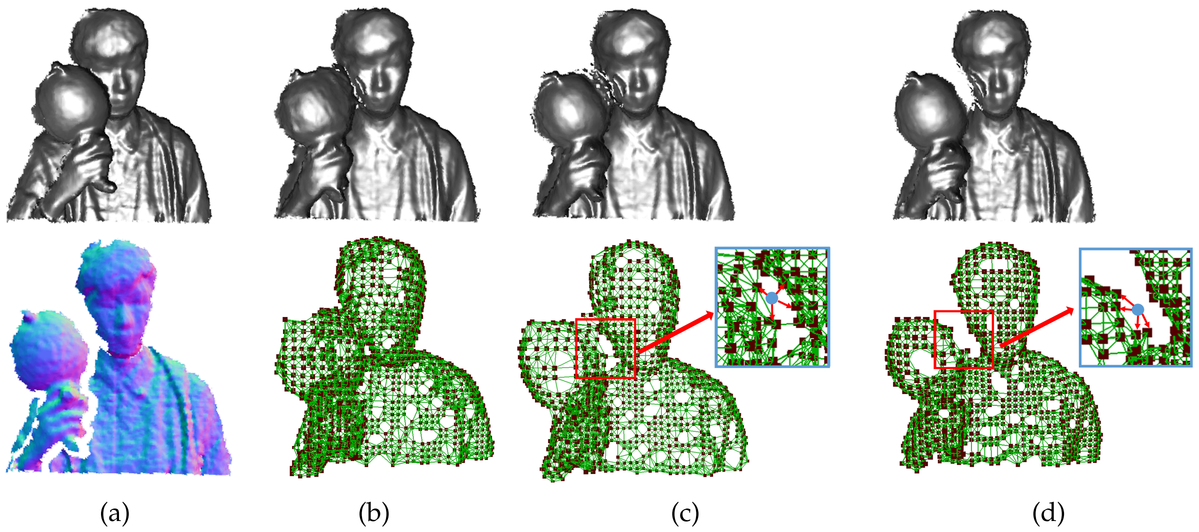


Figure 4.4. Topology compactness. (a) Current models in the key volume and the depth input. The object is moving off the face thus producing topological changes; (b) Using key-volumes [27] alone cannot disconnect the deformation graph thus the object is still connected to performer's face; (c) After the proposed node-to-node compactness a vertex (blue dot) is still skinned (red edges) by disconnected nodes; (d) After proposed vertex–to-node compactness the topological changes are successfully addressed.

$$\mathbf{dist}(r) = \sum_{\forall v \in \mathcal{X}_r} \mathbf{tsdf}(v) + \mathbf{I}(\mathbf{wgt}(v)), \tag{4.15}$$

where v is a rasterized voxel index, $\mathbf{tsdf}(\cdot)$ and $\mathbf{wgt}(\cdot)$ return the TSDF and weight values stored in the voxel, and $\mathbf{I}(x)$ is the indication function which is equivalent to 1 iff $x = 0$. All edges with distance larger than δ ($\delta = 6$ in our implementation) are discarded. We discard ED nodes with less than 2 connected edges, and build a new ED graph with more compact node-to-node connection (Fig. 4.4(c)).

Vertex-to-node compactness: A vertex on the surface should be skinned to a set of nodes which move consistently and form a connected local graph. Thus we propose to deform a vertex or voxel only when its supporting nodes are connected. Such vertex-to-node compactness prevents the noticeable artifacts of deformations and non-rigid fusion on regions where topology changes. As illustrated in Fig. 4.4(d), after applying the topology compactness, the influence of topology changes in the splitting surface are removed explicitly for robust dynamic reconstruction.

4.2.3 Handling Volume Drifting

To capture the motion of the dynamic subject in a larger space, we break the fixed capture volume constraint in former work [62, 27, 32] by enabling the TSDF volume to follow the subject. Under the moving camera setting, DynamicFusion [62] models the camera pose $\mathbf{T}_c$ from the fixed volume to the depth input by combining the estimated pose from both rigid ICP and the rigid component factored out from the non-rigid motion field. However, when the capture volume is movable, such camera pose suffers from accumulated drift error as analysed in [91]. Moreover, since the key volume is reset to the data volume periodically, the accumulated drift error is propagated into the non-rigid motion field. Thus the dynamic subject is warped towards the boundary area of the capture volume to reconstruct imcomplete and weird reconstruction results gradually, causing the "volume drifting" artifact, as shown in Fig. 4.5(a).

To avoid the volume drifting, we prohibit the drift error propagation during resetting the key volume. Specifically, every time the key volume is reset to the data volume, the virtual camera pose $\mathbf{T}_c$ from the volume to the depth input is also reset to guarantee that the dynamic subject is fully captured in the data volume. Note that $\mathbf{T}_i$ is the $\mathbf{SE}(3)$ of the ED node $\mathbf{x}_i$, and $\mathbf{T}_c^*$ denotes the updated virtual camera pose. We can formulate the traslation of the new virtual camera pose as follows:

$$\mathbf{t}_c^* = \mathbf{t}_c - \frac{1}{N}\sum_{\mathbf{x}_i} \mathbf{T}_i \mathbf{x}_i + \mathbf{t}_{center}, \tag{4.16}$$

where N is the number of all the ED nodes, $\mathbf{t}_c$ is the traslation of $\mathbf{T}_c$ and $\mathbf{t}_{center}$ is the centre of the capture volume in the coordinate frame of the volume, combining with an identity matrix $\mathbf{I}_c^* \in \mathbb{SO}_3$ as the rotaion of $\mathbf{T}_c^*$. Besides, the rigid transformation $\mathbf{T}_i$ in each ED node also need to be updated as

$$\mathbf{T}_i^* = \mathbf{T}_i \mathbf{T}_c (\mathbf{T}_c^*)^{-1}. \tag{4.17}$$

The updated motion field $\{\mathbf{T}_i^*\}$ and virtual camera pose $\mathbf{T}_c^*$ can keep the captured geometry stay in the central area of the volume during the tracking process, leading to more complete and reasonale reconstruction results, as shown in Fig. 4.5(a). Note that the above vitual camera pose from the capture volume to the depth input is not related to the physical localization of the flying plateform. The global localization of our FlyFusion system

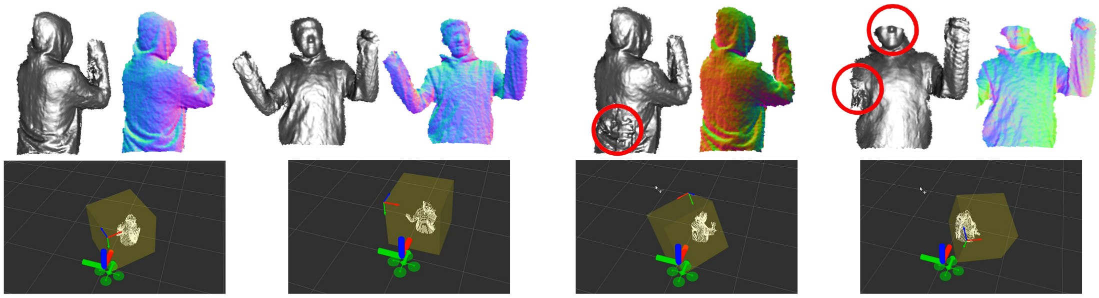

(a) (b)

Figure 4.5. The reconstructed geometry, normal map and the data volume visulization corresponding to the 150th and 300th frames of the "shaking" sequence a) with and b) without handling the volume drifting.

	No. Frames	Duration(s)	Mean speed(m/s)	Distance(m)
Back walk	900	31.584	0.153	4.581
Side walk	800	27.733	0.121	3.603
Panda bag	1100	36.714	0.133	4.716
Drinking	1200	43.288	0.106	4.546
Reading	1400	47.627	0.154	7.153
Circling	1200	41.721	0.194	8.168
Rolling fit	1100	38.172	0.167	6.861

Table 4.1. Statistics and basic metrics of the captured dataset in the experiments.

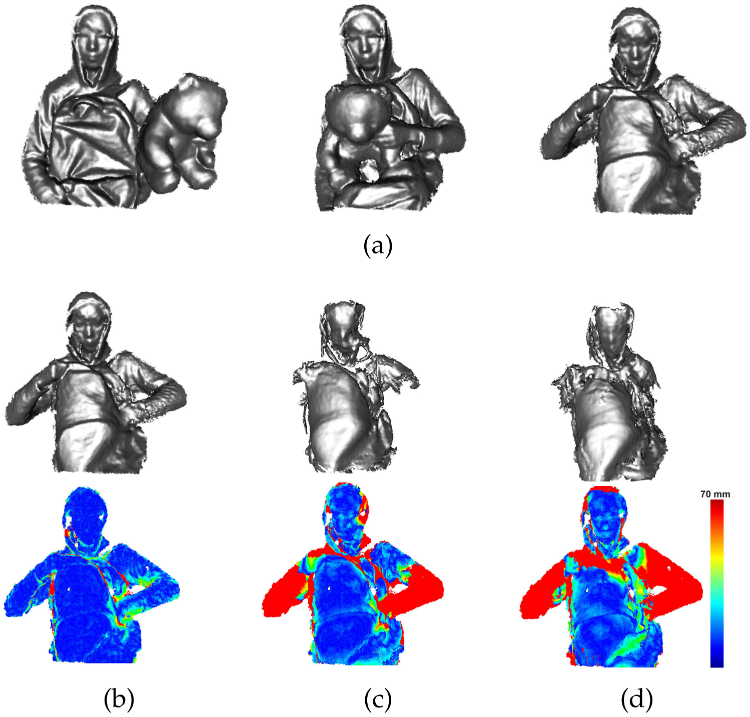
	Our Method	DynamicFusion	VolumeDeform
error (mm)	7.584	19.153	18.517

Table 4.2. Average numerical errors on the *packing doll* sequence, only for the visible surface regions.

relies on Guidance as the onboard navigation module.

4.3 Experimental Results

The proposed ***FlyFusion*** system is evaluated thoroughly in public datasets, captured dataset and synthetic data. Specifically, we record 7 test sequences consisting of over 8000 frames, as shown in Table 1. The reconstruction and active planning results are demonstrated in Fig. 4.1 and Fig. 4.2, which illustrate that our method supports various motions with different shapes and topologies. For more sequential results, we recommend viewing our accompanying video to more clearly visualize and understand the capabilities of our ap-


(a)

(b) (c) (d)

Figure 4.6. Evaluation of chaning topology on the *packing doll* sequence. (a) Geometry results of our method for 440th, 520 and 600th frames. (b, c, d) The results of our method, DynamicFusion and VolumeDeform, respectively. The 1st and 2nd rows present the geometry and error maps, respectively, for the 600th frame.

proach. For fair comparisons, in Sec. 4.3.1, we disable the viewpoint selection so as to particularly evaluate our proposed robust dynamic fusion scheme. Then in Sec. 4.3.2, we enable the viewpoint selection to evaluate the proposed GAME-based active view analysis strategy.

FlyFusion is implemented on a single NVIDIA GeForce GTX TITAN X GPU and a 3.2 GHz 4-core Xeon E3-1230 CPU with 16 GB of memory. The whole pipeline runs at 32 ms per frame, of which 15ms for the motion field optimization, 6 ms for the topology compactness and TSDF fusion, 8 ms for the GAME metric optimization, and 3 ms for all the other operations. All the TSDF volumes are set to be $1.0m^3$ with a voxel size of $4mm$. The parameters λ_{cen} and λ_{mot} for the center and motion term in the GAME metric are set to be 0.5 and 10.0 respectively based on both the balance of each term and the drone maneuverability.

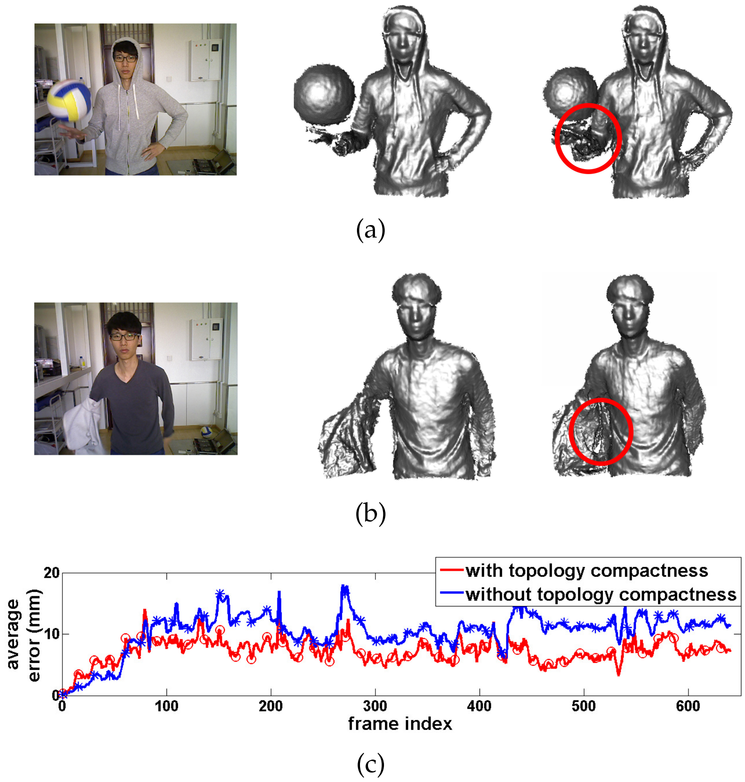


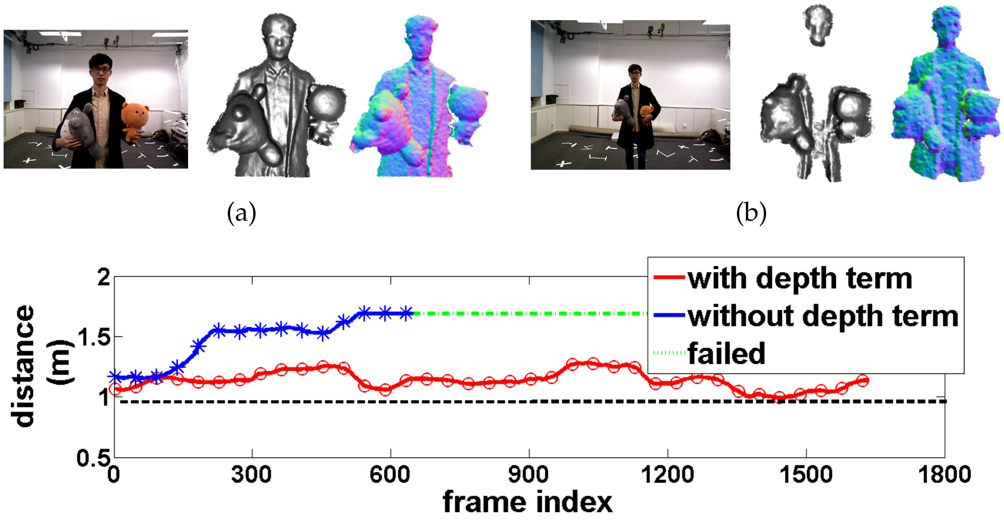
Figure 4.7. Evaluation of the proposed topology compactness strategy on the *clothing changes* sequence. (a) and (b) present the results for the 112th and 378th frames respectively, including color image, the geometry results with and without topology compactness. (c) The quantitative error curve of our method with and without topology compactness for the whole sequence.

4.3.1 Evaluation for Non-rigid Reconstruction

In this subsection, we evaluate the robust dynamic reconstruction scheme in the ***FlyFusion*** system both qualitatively and quantitatively.

Public single-stream RGB-D datasets

We first test our robust dynamic fusion scheme on the public RGB-D datasets of several representative single-view non-rigid reconstruction methods, i.e., VolumeDeform [37], MonoFVV [32] and KillingFusion [72]. The reconstructed results of our scheme are shown in Fig .4.3 for a variety of dynamic scenes, including interactions with different objects (*sun flower*, *minion*, *umbrella* and *frog*), fast motion (*fast loop*, *boxing* and *snoopy*), motion with topology changes (*roll shirt*, *hat* and *Alex*), and motion with loop-closure (*fast loop*, *duck* and *Alex*), which further illustrates the generality of our method to support various motions, shapes and topologies.

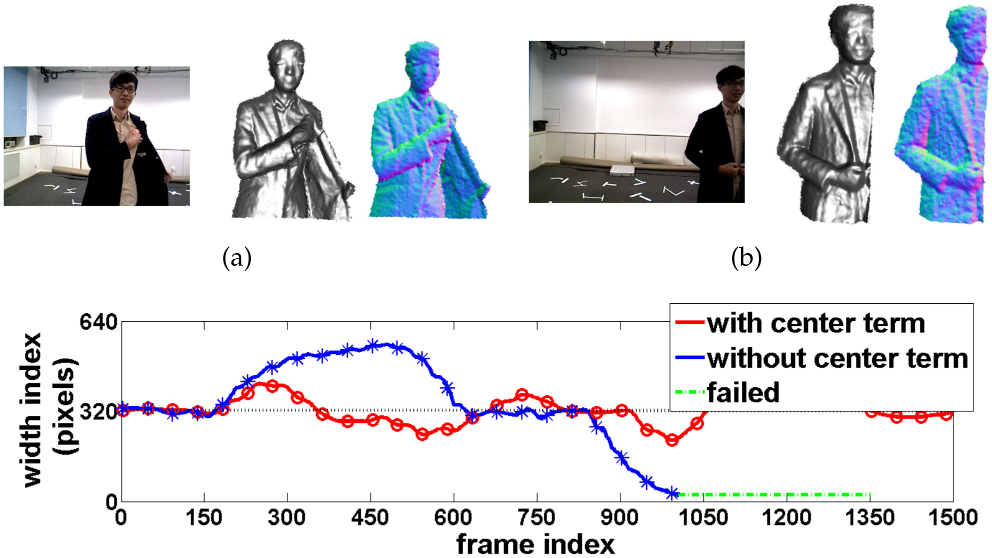


(a)

(b)

(c)

Figure 4.8. Evaluation of the depth term. (a, b) The results with and without the depth term respectively, including the color frame, the geometry result and the input depth image. (c) The quantitative curves of the average depth value of current depth input.

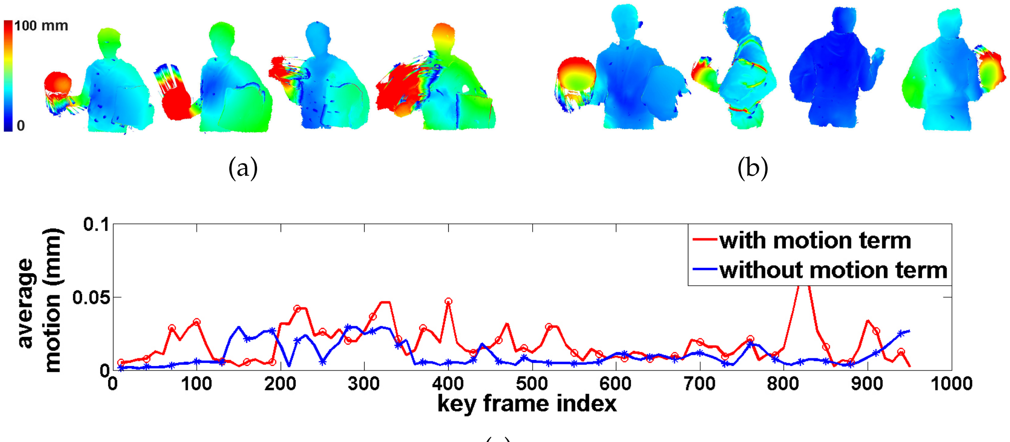


(a)

(b)

(c)

Figure 4.9. Evaluation of the center term. (a, b) The results with and without the center term respectively, including the color frame, the geometry result and the input depth image. (c) The quantitative curves of the average horizontal index of current depth input.

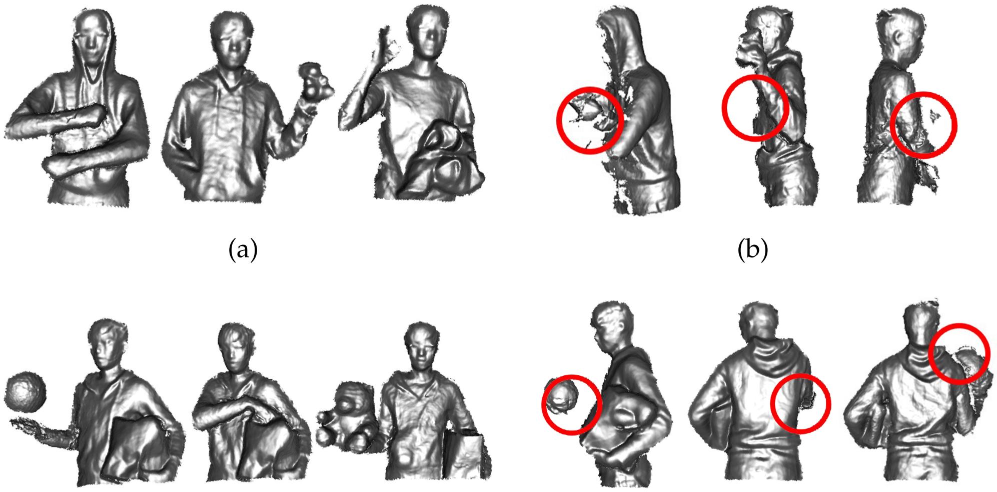


(a)

(b)

(c)

Figure 4.10. Quantitative evaluation of the motion term in the *panda bag* sequence. (a) and (b) are the motion maps of the 219th, 309th, 499th and 799th frames for our method with and without the motion term respectively. (c) The numerical motion curve.



(a)

(b)

(c)

(d)

Figure 4.11. Qualitative evaluation of the motion term. (a, b) The reconstructed geometry results with and without the motion term respectively for the *roll fit* sequence. (c, d) The reconstructed geometry results with and without the motion term respectively for the *panda bag* sequence.

Complex topology changes

Although the state-of-the-art shows outstanding performance on the public datasets, their performance degrades once large topology changes happen. We compare our method with VolumeDeform [37] and DynamicFusion [62] (our re-implementations) in terms of geometry by rendering a pre-reconstructed 3D motion to obtain ground-truth depth frames as input, for the sequence *packing doll*. As shown in Fig. 4.6, our scheme recovers the complex topology changes, while VolumeDeform [37] and DynamicFusion [62] suffer from severe artifacts. The average geometry errors for the entire sequence are also presented in Table. 2.

We further evaluate the proposed topology compactness method by disabling it and merely using the key-volume strategy just as Fusion4D [27]. The comparisons shown in Fig . 4.7(a) and Fig . 4.7(b) illustrate that with the proposed topology compactness strategy, the surface splitting topology changes between key frames are well modeled, leading to more accurate reconstructions. For quantitative comparison, we render the reconstructed scenes into a 2D depth map in the camera view, and compute its MAE (Mean Absolute Error) by taking the depth input as the reference. As expected, with the proposed topology compactness strategy, the MAE can be greatly reduced from 12.46 mm to **7.38 mm** in Fig. 4.7(c). Such MAE calculation is certainly not perfect, yet the reduction of MAE partially reveals the effectiveness of our dynamic fusion scheme for dealing with topology changes.

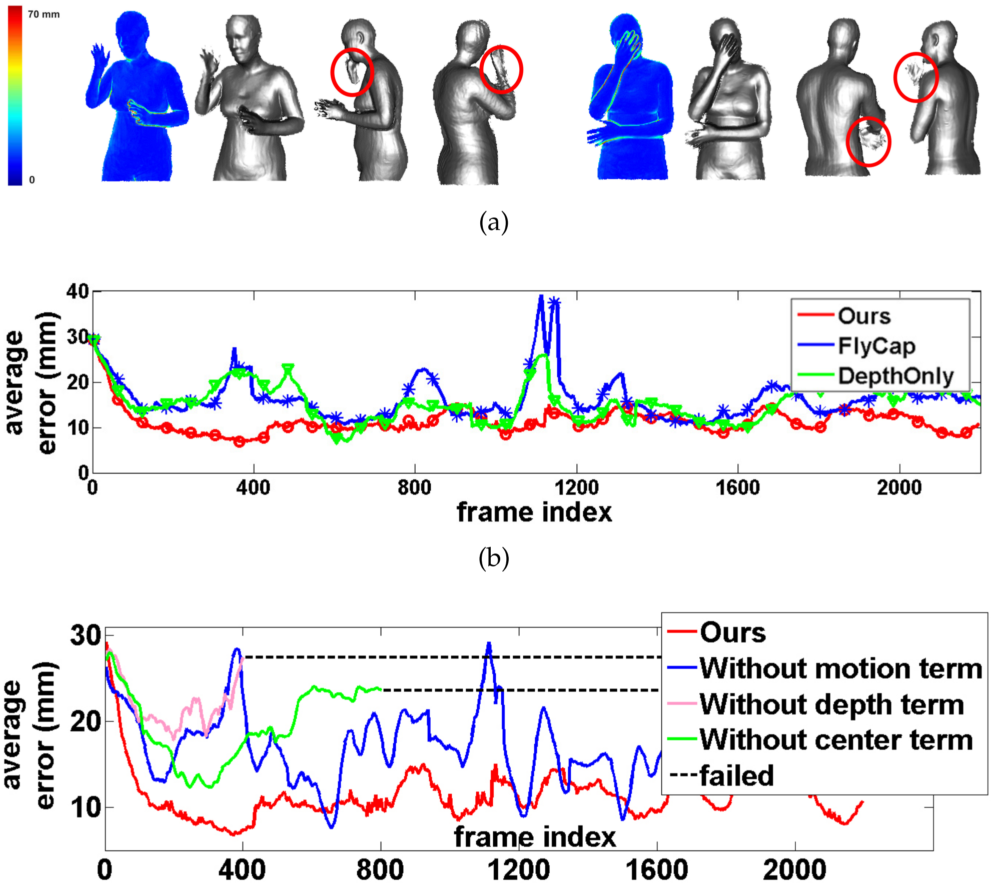
4.3.2 Evaluation for Active View Analysis

In this subsection, we evaluate the proposed GAME metric for active view analysis and prove the mutual gain between active view selection and dynamic reconstruction in both the real captured dataset and synthetic data.

Evaluation on Real Data

For conducting more comprehensive evaluations of the proposed GAME metric, we particularly perform an ablation study on the real captured dataset by disabling each term of the GAME-based optimization respectively.

In Fig. 4.8, we compare the results with and without the depth term in Eqn. 4.3, which illustrates that without depth term the captured depth quality and the reconstruction results deteriorate sharply. With depth term, the flying camera can maintain a relatively stable distance to guarantee the accurate depth measurement ($1m \pm 0.3m$ for ASUS Xtion). Fig. 4.9 further illustrates the effectness of the center term in Eqn. 4.7. By calculating the



(a)

(b)

(c)

Figure 4.12. Synthetic evaluation for non-rigid reconstruction on the *Phoning* and *Smoking* sequences. (a) Our error map, our result, the result of ***FlyCap*** and DepthOnly, respectively. (b) The numerical results of the average error. (c) The numerical curve of the ablation study of the GAME metric.

average horizontal pixel index of the valid pixels and comparing to the central index (320 for VGA input) in the captured depth images, our method with the center term successfully maintains the dynamic object in the central region, with the standard deviation **20.1 pixels**.

The quanlitative comparison of the motion term in Eqn. 4.10 is provided in Fig. 4.11, which shows that without the motion term the flying camera fails to capture the dynamic areas with large non-rigid motion and occlusion, causing severe artifacts in the final reconstructed results. In contrast, with the motion term, it successfully captures and reconstructs the dynamic object with complex motion and topology changes. For quantitative evaluation, we render the per-vertex motion field into 2D image in the camera view

	Error (mm) with motion term	Error (mm) without motion term
Directions	9.37	10.84
Discussion	14.48	15.90
Eating	10.26	16.69
Greeting	15.47	18.42
Phoning	8.06	14.28
Photo	10.58	14.57
Posing	9.21	13.35
Purchases	12.89	15.63
Sitting	13.13	12.93
SittingDown	14.72	17.52
Smoking	12.38	18.12
Waiting	8.67	9.94
WalkDog	10.15	13.52
Walking	8.94	12.89
WalkTogether	9.62	13.07

Table 4.3. Average geometry error with and without the motion term on the SURREAL dataset with motions from Human3.6M.

and then calculate the average per-pixel projected motion magnitude, which reveals the amount of motions in the captured images. As shown in Fig. 4.10, the motion term greatly improves the collection of valid motions from 8.71 mm to **18.14 mm** in average.

These evaluations aboce illustrate that our GAME-based active view analysis metric tends to select the viewpoints for capturing the high quality depth areas through the jointly optimization of the geometry, central location and motion. These areas own important non-rigid contents which is highly adaptive to the immediate dynamic reconstruction result, and can further guide the following deformation and reconstruction to improve the consequent dynamic reconstruction.

Evaluation on Synthetic Data

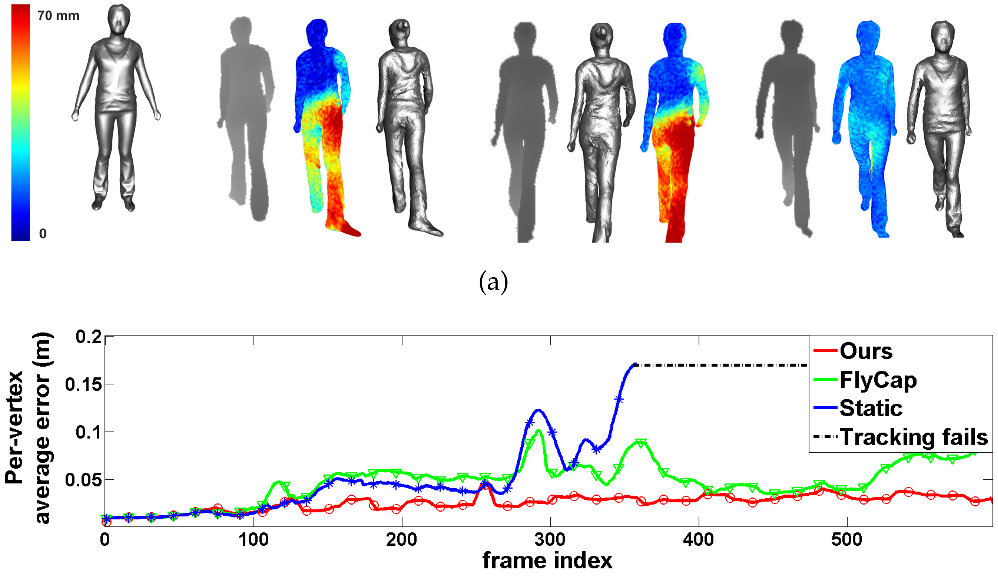
We testify the GAME-based view analysis for both the online and offline scene reconstruction using synthetic data, where the ground-truth geometry and motion are provided and the drone maneuverability is simulated via the *ROS* system.

For online non-rigid fusion, we utilize the SURREAL dataset with 15 various motions from Human3.6M [**?**] to generate synthetic depth images with ground-truth mesh in the view of the simulated drone. Our method is compared with the fixed view planning strategy in Xu *et al.* [91] (denoted as ***FlyCap***), which simply keeps a fixed distance and a fixed view angle based on current depth input. To further analyze the mutual effects between active view selection and dynamic reconstruction, we compare our method with simply applying the GAME metric to current depth input instead of the whole dynamic reconstructed result, by accumulating the per-pixel scene flow provided by SURREAL as the motion term, denoted as *DepthOnly*.

Fig. 4.12(a) shows that both ***FlyCap*** and DepthOnly suffer from artifacts in those moving areas while our scheme captures these motion successfully. The corresponding quantitative result is provided in Fig. 4.12(b) and we achieve **10.67 mm** average geometry error compared with 18.50 mm of ***FlyCap*** and 14.37 mm of DepthOnly, which illustrates that our active view analysis strategy not only outperforms previous fixed strategy but also benefits from the robust dynamic reconstruction which provides more reliable geometry and motion information. For thorough quantitative analysis, the ablation study of the GAME metric on the synthetic data is provided in Fig. 4.12(c). It is not surprising that the reconstruction results without depth term or center term deteriorate sharply due to limited captured depth quality, while our method with full GAME metric improves the dynamic reconstruction by selecting more important capturing views through jointly optimization of geometry, central location and motion.

To further illustrate the effectness of the motion term, an ablation study of the motion term on the whole SURREAL dataset with motions from Human3.6M is provided in Table. 3. The average numerical error for all the sequences of our method with the motion term is **11.19 mm**, compared with the 15.51 mm of the one without the motion term, which illustrates the effectiveness of our GAME metric and the motion term.

For off-line non-rigid tracking, we utilize a synthetic sequence with topology-coherent models as the ground truth by rendering the full-body depth maps in the virtual camera poses. We compare our method with ***FlyCap*** and the one with a static camera (denoted as *Static*) by performing non-rigid tracking using the recorded depth images of three methods, respectively. As shown in Fig. 4.13, the average per-vertex tracking error of our method is **25.62 mm** compared with the 47.87 mm of ***FlyCap*** and 67.13 mm for Static,



(a)

(b)

Figure 4.13. Synthetic evaluation for non-rigid tracking. The used template is at the top-left. The three triples correspond to the results of Static, FlyCap and our method. Each triple includes the depth input, the error map and the tracking result. The quantitative error curves are drawn at the bottom.

implying the effectiveness of our GAME-based active view planning strategy.

CHAPTER 5

LIMITATION, DISCUSSION AND FURTHER WORK

The proposed *iHuman3D* still has limitations. The drone has to maintain in low-speed for the safety issue. Furthermore, the drone also lacks the ability of obstacle avoidance and cannot handle severe environments like fierce winds. A more compactly designed drone integrated with different kinds of sensors is needed to enhance the applicability. Our reconstruction method is restricted by the capabilities of the Xtion sensor, which has to work indoor and capture RGBD data with limited quality. We are looking forward to the binocular solution combining with the data driven learning technique to enhance the quality of the captured raw data. Moreover, we are hoping to eliminate the manually selected initial pose of the drone *iHuman3D* also needs to select to start scanning.

As the first step to explore the problem of active dynamic scene reconstruction using a flying camera, the proposed FlyFusion still has limitations. From the aspect of the drone, our method relies on the stability of the drone and cannot handle severe environments like fierce winds. The drone also has to maintain in a low-speed flight mode for the safety issue, considering about its size and the distance to the captured subject. Furthermore, the drone also lacks the ability of obstacle avoidance so as to explore different areas like separated rooms. A more compactly designed drone integrated with different kinds of sensors is needed to enhance the applicability. Another issue of our current work is the restriction of the capabilities of the Xtion sensor, which has to work indoor and capture RGBD data with limited quality. We are looking forward to the binocular solution combining with the data driven learning technique to enhance the quality of the captured raw data. Finally, in terms of dynamic reconstruction, current key-volume strategy with topology compactness has to re-initialize the motion field periodically, which hinders the acquisition of the topological consistent reconstruction results for post-processing. In the future, we are hoping to perform an off-line joint optimization of all the data captured from the online reconstruction step, to improve the final reconstruction results in terms of accuracy and topology consistency.

REFERENCES

[1] Artec 3D. Artec Shapify Booth. `https://www.shapify.me/`, Jan. 2016.

[2] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3):587–594, July 2003.

[3] John Aloimonos, Isaac Weiss, and Amit Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1(4):333–356, Jan 1988.

[4] Edouard Auvinet, Jean Meunier, and Franck Multon. Multiple depth cameras calibration and body volume reconstruction for gait analysis. In *Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*, pages 478–483. IEEE, 2012.

[5] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.

[6] F. Bissmarck, M. Svensson, and G. Tolt. Efficient algorithms for next best view evaluation. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5876–5883, Sept 2015.

[7] Blender Online Community. Blender - a 3d modelling and rendering package. http://www.blender.org, 2012.

[8] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *International Journal of Computer Vision*, 56(3):179–194, 2004.

[9] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1611.08050*, 2016.

[10] Avishek Chatterjee and Venu Madhav Govindu. Noise in structured-light stereo depth cameras: Modeling and its applications. *arXiv preprint arXiv:1505.01936*, 2015.

[11] Chongyu Chen, Jianfei Cai, Jianmin Zheng, Tat-Jen Cham, and Guangming Shi. A color-guided, region-adaptive and depth-selective unified framework for kinect depth recovery. In *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*, pages 007–012. IEEE, 2013.

[12] S. Y. Chen and Y. F. Li. Vision sensor planning for 3-d model acquisition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(5):894–904, Oct 2005.

[13] Shengyong Chen, YF Li, Wanliang Wang, and Jianwei Zhang. Active sensor planning for multiview vision tasks. 2008.

[14] Shengyong Chen, Youfu Li, and Ngai Ming Kwok. Active vision in robotic systems: A survey of recent developments. *Int. J. Rob. Res.*, 30(11):1343–1377, September 2011.

[15] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG)*, 34(4):69, 2015.

[16] C. Connolly. The determination of next best views. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 432–435, Mar 1985.

[17] Yan Cui and Didier Stricker. 3d shape scanning with a kinect. In *ACM SIGGRAPH 2011 Posters*, SIGGRAPH '11, pages 57:1–57:1, New York, NY, USA, 2011. ACM.

[18] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 303–312, New York, NY, USA, 1996. ACM.

[19] J. Daudelin and M. Campbell. An adaptable, probabilistic, next-best view algorithm for reconstruction of unknown 3-d objects. *IEEE Robotics and Automation Letters*, 2(3):1540–1547, July 2017.

[20] E. De Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graph. (Proc. of SIGGRAPH)*, 27(3), 2008.

[21] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graph.*, 27(3):98:1–98:10, August 2008.

[22] Edilson De Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *ACM Transactions on Graphics (TOG)*, 27(3):98, 2008.

[23] Jeffrey Delmerico, Stefan Isler, Reza Sabzevari, and Davide Scaramuzza. A comparison of volumetric information gain metrics for active 3d object reconstruction. *Autonomous Robots*, Apr 2017.

[24] Jonathan Deutscher, Andrew Blake, and Ian Reid. Articulated body motion capture by annealed particle filtering. In *Proc. of CVPR*, 2000.

[25] DJI. DJI A2. `https://www.dji.com/a2/`, 2016.

[26] Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, and Shahram Izadi. Motion2fusion: Real-time volumetric performance capture. *ACM Trans. Graph.*, 36(6):246:1–246:16, November 2017.

[27] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. Fusion4D: Real-time Performance Capture of Challenging Scenes. In *ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques*, 2016.

[28] J. Gall, C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, and H.P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *Proc. of CVPR*, 2009.

[29] Varun Ganapathi, Christian Plagemann, Daphne Koller, and Sebastian Thrun. Real time motion capture using a single time-of-flight camera. In *Proc. of CVPR*, 2010.

[30] Fei Gao and Shaojie Shen. Online quadrotor trajectory generation and autonomous navigation on point clouds. In *Safety, Security, and Rescue Robotics (SSRR), 2016 IEEE International Symposium on*, pages 139–146. IEEE, 2016.

[31] Kaiwen Guo, Feng Xu, Yangang Wang, Yebin Liu, and Qionghai Dai. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3083–3091, 2015.

[32] Kaiwen Guo, Feng Xu, Tao Yu, Xiaoyang Liu, Qionghai Dai, and Yebin Liu. Real-time geometry, albedo and motion reconstruction using a single rgbd camera. *ACM Transactions on Graphics (TOG)*, 2017.

[33] Nils Hasler, Bodo Rosenhahn, Thorsten Thormahlen, Michael Wand, Juergen Gall, and Hans-Peter Seidel. Markerless Motion Capture with Unsynchronized Moving Cameras. In *Proc. of CVPR*, 2009.

[34] Benjamin Hepp, Matthias Nießner, and Otmar Hilliges. Plan3d: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction. *arXiv preprint arXiv:1705.09314*, 2017.

[35] Adrian Hilton, Daniel Beresford, Thomas Gentils, Raymond Smith, Wei Sun, and John Illingworth. Whole-body modelling of people from multiview images to populate virtual worlds. *The Visual Computer*, 16(7):411–436, 2000.

[36] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[37] Matthias Innmann, Michael Zollhöfer, Matthias Nießner, Christian Theobalt, and Marc Stamminger. VolumeDeform: Real-time Volumetric Non-rigid Reconstruction. October 2016.

[38] Intel. NUC5i7RYH Mini PC. `http://www.intel.com/content/www/us/en/nuc/nuc-kit-nuc5i7ryh/`, 2016.

[39] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1325–1339, jul 2014.

[40] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza. An information gain formulation for active volumetric 3d reconstruction. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3477–3484, May 2016.

[41] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic Studio: A Massively Multiview System for Social Motion Capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015.

[42] Niels Joubert, Mike Roberts, Anh Truong, Floraine Berthouzoz, and Pat Hanrahan. An interactive tool for designing quadrotor camera shots. *ACM Trans. Graph.*, 34(6):238:1–238:11, October 2015.

[43] C. Kerl, J. Sturm, and D. Cremers. Dense Visual SLAM for RGB-D Cameras. In *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, 2013.

[44] Imran Khan. Robust sparse and dense nonrigid structure from motion. *IEEE Transactions on Multimedia*, 20(4):841–850, 2018.

[45] Kichang Kim, Akihiko Torii, and Masatoshi Okutomi. Joint estimation of depth, reflectance and illumination for depth refinement. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1–9, 2015.

[46] A Kolahi, Mo Hoviattalab, Tahmineh Rezaeian, M Alizadeh, M Bostan, and Hossein Mokhtarzadeh. Design of a marker-based human motion tracking system. *Biomedical Signal Processing and Control*, 2(1):59–67, 2007.

[47] Simon Kriegel, Tim Bodenmüller, Michael Suppa, and Gerd Hirzinger. A surface-based next-best-view approach for automated 3d model completion of unknown objects. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4869–4874. IEEE, 2011.

[48] Simon Kriegel, Christian Rink, Tim Bodenmüller, and Michael Suppa. Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects. *Journal of Real-Time Image Processing*, 10(4):611–631, Dec 2015.

[49] HyeokHyen Kwon, Yu-Wing Tai, and Stephen Lin. Data-driven depth map refinement via multi-scale sparse representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 159–167, 2015.

[50] Hao Li, Bart Adams, Leonidas J Guibas, and Mark Pauly. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia)*, 28(5):175, 2009.

[51] Hao Li, Etienne Vouga, Anton Gudym, Linjie Luo, Jonathan T Barron, and Gleb Gusev. 3d self-portraits. *ACM Transactions on Graphics (TOG)*, 32(6):187, 2013.

[52] Rui Li, Minjian Pang, Cong Zhao, Guyue Zhou, and Lu Fang. Monocular Long-Term Target Following on UAVs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 29–37, 2016.

[53] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Joint image filtering with deep convolutional networks. *arXiv preprint arXiv:1710.04200*, 2017.

[54] Shaoguo Liu, Ying Wang, Jue Wang, Haibo Wang, Jixia Zhang, and Chunhong Pan. Kinect depth restoration via energy minimization with tv 21 regularization. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 724–724. IEEE, 2013.

[55] Yebin Liu, Juergen Gall, Carsten Stoll, Qionghai Dai, Hans-Peter Seidel, and Christian Theobalt. Markerless motion capture of multiple characters using multiview image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2720–2735, 2013.

[56] Yebin Liu, Carsten Stoll, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1249–1256. IEEE, 2011.

[57] Yebin Liu, Carsten Stoll, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. Markerless motion capture of interacting characters using multi-view image segmentation. In *Proc. of CVPR*, 2011.

[58] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Trans. Graph. (Proc. of SIGGRAPH)*, 34(6):248, 2015.

[59] Tanwi Mallick, Partha Pratim Das, and Arun Kumar Majumdar. Characterizations of noise in kinect depth images: A review. *IEEE Sensors journal*, 14(6):1731–1740, 2014.

[60] Riccardo Monica and Jacopo Aleotti. Contour-based next-best view planning from point cloud segmentation of unknown objects. *Autonomous Robots*, 42(2):443–458, Feb 2018.

[61] Hans Moravec and Alberto Elfes. High resolution maps from wide angle sonar. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 116–121. IEEE, 1985.

[62] Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. DynamicFusion: Reconstruction and Tracking of Non-Rigid Scenes in Real-Time. June 2015.

[63] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *Proc. of ISMAR*, pages 127–136, 2011.

[64] Chuong V Nguyen, Shahram Izadi, and David Lovell. Modeling kinect sensor noise for improved 3d reconstruction and tracking. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 524–530. IEEE, 2012.

[65] Richard Pito. A solution to the next best view problem for automated surface acquisition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(10):1016–1030, October 1999.

[66] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[67] Ramesh Raskar, Hideaki Nii, Bert Dedecker, Yuki Hashimoto, Jay Summet, Dylan Moore, Yong Zhao, Jonathan Westhues, Paul Dietz, John Barnwell, et al. Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators. In *ACM Transactions on Graphics (TOG)*, volume 26, page 36. ACM, 2007.

[68] Mike Roberts, Debadeepta Dey, Anh Truong, Sudipta Sinha, Shital Shah, Ashish Kapoor, Pat Hanrahan, and Neel Joshi. Submodular trajectory optimization for aerial 3d scanning. In *International Conference on Computer Vision (ICCV) 2017*, 2017.

[69] Loren Arthur Schwarz, Diana Mateus, and Nassir Navab. Multiple-activity human body tracking in unconstrained environments. In *Articulated Motion and Deformable Objects*, pages 192–202. Springer, 2010.

[70] William R. Scott, Gerhard Roth, and Jean-François Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv.*, 35(1):64–96, March 2003.

[71] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time Human Pose Recognition in Parts from Single Depth Images. In *Proc. of CVPR*, 2011.

[72] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic. KillingFusion: Non-rigid 3D Reconstruction without Correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[73] Carsten Stoll, Nils Hasler, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. Fast Articulated Motion Tracking using a Sums of Gaussians Body Model. In *Proc. of ICCV*, 2011.

[74] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Transactions on Graphics (TOG)*, 26(3):80, 2007.

[75] Ting Sun, Shengyi Nie, Dit-Yan Yeung, and Shaojie Shen. Gesture-based piloting of an aerial robot using monocular vision. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 5913–5920. IEEE, 2017.

[76] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan. Scanning 3d full human bodies using kinects. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):643–650, April 2012.

[77] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from Synthetic Humans. In *Proc. of CVPR*, 2017.

[78] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 2017.

[79] J. Irving Vasquez-Gomez, L. Enrique Sucar, Rafael Murrieta-Cid, and Efrain Lopez-Damian. Volumetric next-best-view planning for 3d object reconstruction with positioning error. *International Journal of Advanced Robotic Systems*, 11(10):159, 2014.

[80] Vicon. Vicon systems. `http://www.vicon.com`, 2016.

[81] Daniel Vlasic, Rolf Adelsberger, Giovanni Vannucci, John Barnwell, Markus Gross, Wojciech Matusik, and Jovan Popović. Practical Motion Capture in Everyday Surroundings. *ACM Trans. Graph. (Proc. of SIGGRAPH)*, 26(3), 2007.

[82] Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popovic. Articulated mesh animation from multi-view silhouettes. *ACM Trans. Graph. (Proc. of SIGGRAPH)*, 27(3), 2008.

[83] Kangkan Wang, Guofeng Zhang, and Shihong Xia. Templateless non-rigid reconstruction and motion tracking with a single rgb-d camera. *IEEE Transactions on Image Processing*, 26(12):5966–5979, 2017.

[84] Yangang Wang, Yebin Liu, Xin Tong, Qionghai Dai, and Ping Tan. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. *IEEE Transctions on Visualization and Computer Graphics*, 2017.

[85] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger. ElasticFusion: Real-Time Dense SLAM and Light Source Estimation. *Intl. J. of Robotics Research, IJRR*, 2016.

[86] HJ Woltring. New possibilities for human motion studies by real-time light spot position measurement. *Biotelemetry*, 1(3):132, 1974.

[87] Chenglei Wu, Yebin Liu, Qionghai Dai, and Bennett Wilburn. Fusing multiview and photometric stereo for 3d reconstruction under uncalibrated illumination. *IEEE transactions on visualization and computer graphics*, 17(8):1082–1095, 2011.

[88] Chenglei Wu, Carsten Stoll, Levi Valgaerts, and Christian Theobalt. On-set performance capture of multiple actors with a stereo camera. *ACM Trans. Graph. (Proc. of SIGGRAPH)*, 32(6), 2013.

[89] Chenglei Wu, Kiran Varanasi, Yebin Liu, H-P Seidel, and Christian Theobalt. Shading-based dynamic shape refinement from multi-view video under general illumination. In *Proc. of ICCV*, 2011.

[90] Xsens. `http://www.xsens.com`, 2016.

[91] L. Xu, Y. Liu, W. Cheng, K. Guo, G. Zhou, Q. Dai, and L. Fang. Flycap: Markerless motion capture using multiple autonomous flying cameras. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2017.

[92] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146–151, Jul 1997.

[93] Jingyu Yang, Xinchen Ye, Kun Li, Chunping Hou, and Yao Wang. Color-guided depth recovery from rgb-d data using an adaptive autoregressive model. *IEEE transactions on image processing*, 23(8):3443–3458, 2014.

[94] Genzhi Ye, Yebin Liu, Nils Hasler, Xiangyang Ji, Qionghai Dai, and Christian Theobalt. Performance Capture of Interacting Characters with Handheld Kinects. In *Proc. of ECCV*, 2012.

[95] Mao Ye, Xianwang Wang, Ruigang Yang, Liu Ren, and Marc Pollefeys. Accurate 3D pose estimation from a single depth image. In *Proc. of ICCV*, 2011.

[96] Lap-Fai Yu, Sai-Kit Yeung, Yu-Wing Tai, and Stephen Lin. Shading-based shape refinement of rgb-d images. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1415–1422. IEEE, 2013.

[97] Tao Yu, Kaiwen Guo, Feng Xu, Yuan Dong, Zhaoqi Su, Jianhui Zhao, Jianguo Li, Qionghai Dai, and Yebin Liu. Bodyfusion: Real-time capture of human motion and surface geometry using a single depth camera. In *The IEEE International Conference on Computer Vision (ICCV)*. ACM, October 2017.

[98] Peizhao Zhang, Kristin Siu, Jianjie Zhang, C Karen Liu, and Jinxiang Chai. Leveraging depth cameras and wearable pressure sensors for full-body kinematics and dynamics capture. *ACM Transactions on Graphics (TOG)*, 33(6):221, 2014.

[99] Xin Zhang and Ruiyuan Wu. Fast depth image denoising and enhancement using a deep convolutional network. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 2499–2503. IEEE, 2016.

[100] Guyue Zhou, Lu Fang, Ketan Tang, Honghui Zhang, Kai Wang, and Kang Yang. Guidance: A visual sensing platform for robotic applications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 9–14, 2015.

[101] Michael Zollhöfer, Matthias Nießner, Shahram Izadi, Christoph Rehmann, Christopher Zach, Matthew Fisher, Chenglei Wu, Andrew Fitzgibbon, Charles Loop, Christian Theobalt, et al. Real-time Non-rigid Reconstruction using an RGB-D Camera. *ACM Transactions on Graphics (TOG)*, 33(4):156, 2014.

APPENDIX A

SYSTEM SETUP AND AERIAL ROBOT CONTROL

This appendix provides system setup of aerial robot system used in ***iHuman3D*** and ***FlyFusion*** and the aerial robot control with execution deviation.

A.1 System Setup

This section provides more details about the used hardware and software components in our ***iHuman3D*** and ***FlyFusion*** system. The following Fig. A.1 shows the platform of flying camera in the proposed systems and its three main layers.

- The bottom layer is the execute & sensor layer, which consists of the on-board sensors only accessed by the middle hardware abstraction layer (HAL).
- In the middle layer, we utilize two kinds of products from DJI. The ***DJI A2***[25] works as the flight controller, while ***DJI Guidance***[100] works as the navigation component. These two components connect each other with CAN ports, while they can communicate with the highest layer through the series port using DJI SDK.
- The top layer is the algorithm layer. It is programmable and related to all the on-board algorithms. The ***Intel NUC***[38] works as the on-board computing unit, and the ***ASUS Xtion*** acquires the RGBD data of the scene. Based on the NBV Commands from ourGAME-based active view planning module, the ***PID Controller*** calculates the PID parameters for UAV control, which is executed on the ***Intel NUC***.

The SDK supported by DJI is utilized to communicate between the highest programmable layer and the middle HAL layer. All the velocities including the three axises velocities and the three angular velocities (roll, pitch and yaw) can be set from ***Intel NUC*** to ***DJI***

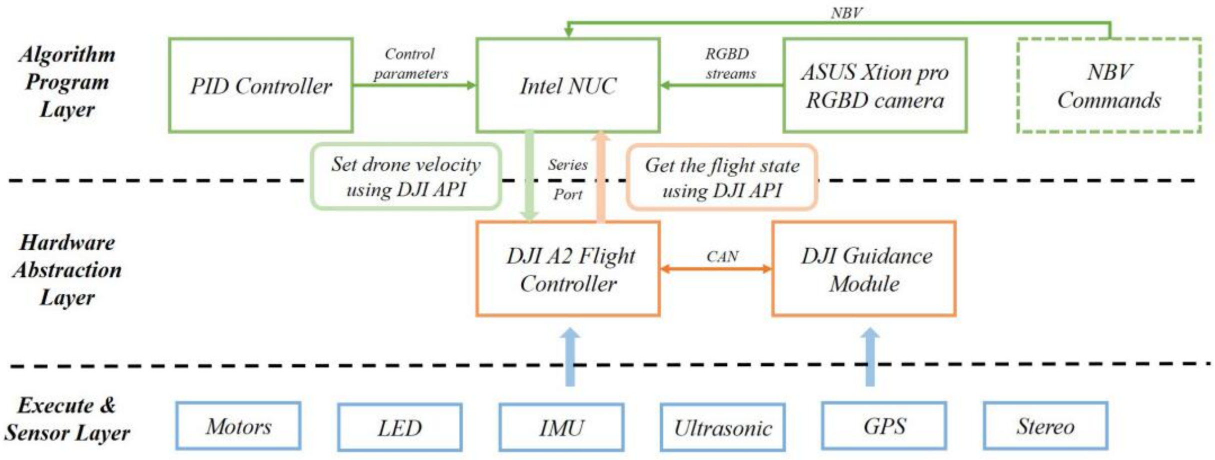


Figure A.1. The layered illustration of the aerial robot platform in ***iHuman3D*** and ***FlyFusion*** system.

A2. Without velocity setting, the drone will stay hovering. On the other hand, we can collect the flight states from ***DJI A2*** to ***Intel NUC***, including all the sensor data, the current altitude, the current velocities of the drone, etc.

Based on the acquired states of the flying camera and current next best view (NBV) command, we utilize the PID controllers run on the ***Intel NUC*** to directly tune the three linear velocities (along three axises) and the three angular velocities of the aerial robot platform, without an additional feedback controller for trajectory tracking on the aerial robot platform. More details about the PID controllers for the task of active view planning are provided in the following supplemental material.

A.2 Flying Camera Control

This section elaborates the control for the execution of the flying camera towards current desired view point. Recall that the NBV command includes the 3-DOF vector $\mathbf{v} = (r, \phi, l)$, and a robot yaw angle ϕ representing the desired view location along the three XYZ axises and the yaw angle in volume coordinate frame of the flying camera. By utilizing the transform matrix $\mathbf{T}_{v2w} = {\mathbf{T}_{w2v}}^{-1}$, we can transform the desired volume view to world coordinate $\mathbf{W}$, The PID controllers are utilized to turn the desired view $\mathbf{W}^*$ into the desired linear and angular velocities. Firstly, current corresponding state of the flying camera

$\mathbf{W}_{cur} = (x_{cur}, y_{cur}, z_{cur})$ is extracted from DJI A2. Then the error between the measurement and the desired values is formulated as:

$$\mathrm{Err} = (\mathrm{Err}_x, \mathrm{Err}_y, \mathrm{Err}_z, \mathrm{Err}_\phi) = \mathbf{W}_{cur} - \mathbf{W}^*, \tag{A.1}$$

which is applied to the PID controllers to tune the velocities of the flying camera. For proposed systems, two different PID controllers are applied to the linear and angular velocities respectively.

A.2.1 PID controller for the linear velocities

The target of this controller is to ensure that the flying camera reaches the desired location stably by tuning the three linear velocities along the XYZ axes. Taking the Z axis for example, we use the following formulation to calculate the desired linear velocity V_z along the Z axis of the camera coordinate system:

$$V_z = K_p \times \mathrm{Err}_z - K_d \times \frac{\partial \mathrm{Err}_z}{\partial t}, \tag{A.2}$$

where the K_p is the proportional coefficient, the K_d is the derivative coefficient. During all the experiments of ***iHuman3D*** and ***FlyFusion*** system, K_p is 2.6 and K_d is 0.12 for the Z axis

The calculation of the linear velocities of the X and Y axes is similar to Eqn. A.2, only with different proportional and derivative coefficients according to the maneuverability of the flying camera. For the linear velocity V_x along the X axis, the proportional and derivative coefficients are set to be 1.5 and 0.1, respectively. For the linear velocity V_y along the Y axis, the proportional and derivative coefficients are set to be 1.0 and 0.02, respectively. The final velocity is alsotruncated to be no larger than 1.0 m/s for safety guarantee.

A.2.2 PID controller for the angular velocities

The target of this controller is to ensure that the flying camera maintains a specific view angle in the desired location. Recall that the roll and yaw angular velocity are set to be zero, and thus only the pitch angular velocity is controlled by this PID controller. Based on the error Err_ϕ calculated from Eqn. A.1, we use the following formulation to tune the pitch angular velocity:

$$W_{\mathsf{pitch}} = \mathsf{K}_\mathsf{p} \times \mathsf{Err}_\phi - \mathsf{K}_\mathsf{d} \times \frac{\partial \mathsf{Err}_\phi}{\partial \mathsf{t}}. \tag{A.3}$$

To improve the stability of the flying camera during tuning the pitch angular velocity, the proportional and derivative coefficients are piecewisely defined as follows:

$$(\mathsf{K}_\mathsf{p}, \mathsf{K}_\mathsf{d}) = \begin{cases} (2.4, 0.2) & \mathsf{if}\ \mathbf{abs}(\mathsf{Err}_\phi) \geqslant 25^o \\ (1.2, 0.05) & \quad \mathsf{otherwise} \end{cases} \tag{A.4}$$

In addition, the final angular velocity is also truncated to be no larger than 90 degrees/s for safety guarantee.

APPENDIX B

HUMAN PERCEPTION SIMULATION AND VISUALIZATION PLATFORM

This appendix chapter presents human perception simulation an visualization platform. As key components of proposed ***iHuman3D*** and ***FlyFusion***'s work flow shown in Fig.B.1, simulation platform plays a alternative role of sensing and execution module like flying camera described thoroughly in Appx.A. Visualization platform obtains information from both active reconstruction and execution modules and displays them in a uniform global space. Both simulation and visualization platform provide furtherance to algorithm evaluation in ***iHuman3D*** and ***FlyFusion*** systems. Specified issues are discussed in this chapter.

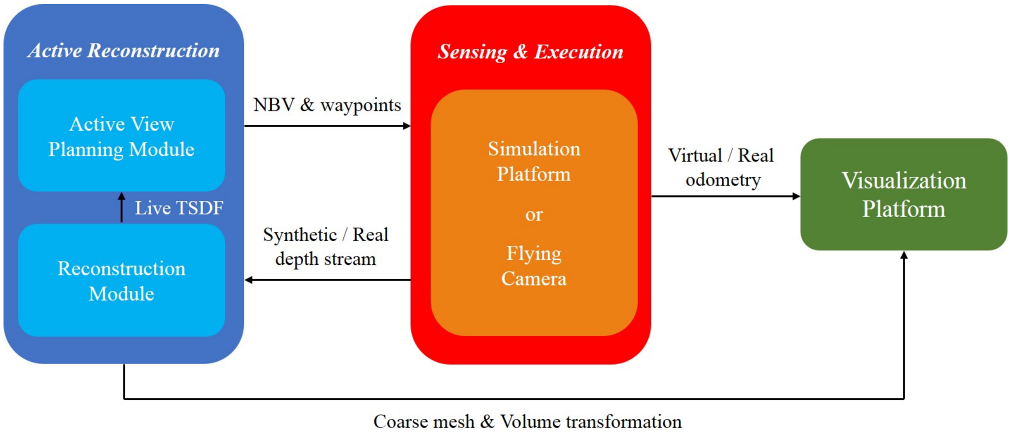


Figure B.1. Work flow of proposed ***iHuman3D*** and ***FlyFusion***.

B.1 Simulation Platform on Human Perception

To evaluate algorithms in practical system is inefficient and sometimes dangerous especially for unpredictable human-centered motions, constructing a simulation platform is advantageous and may accelerate algorithm assessment. This section provides details of the novel and powerful simulation platform used in proposed systems, which is in applicable of synthetic simulation on ***general*** human perception tasks.

Following the recent pioneer work SURREAL [78], to simulate human body morphology, realistic SMPL [58] models are adopted which contain considerable gender, height and shape variation. A 20 bone skeleton is embedded into the SMPL model, human motions are further driven by skeleton data from Human3.6M [39] dataset captured from professional performer with large pose variations in daily activities. Pose blend-shape and shape blend-shape are applied to SMPL model to generate instant body shape in each skeleton frame.

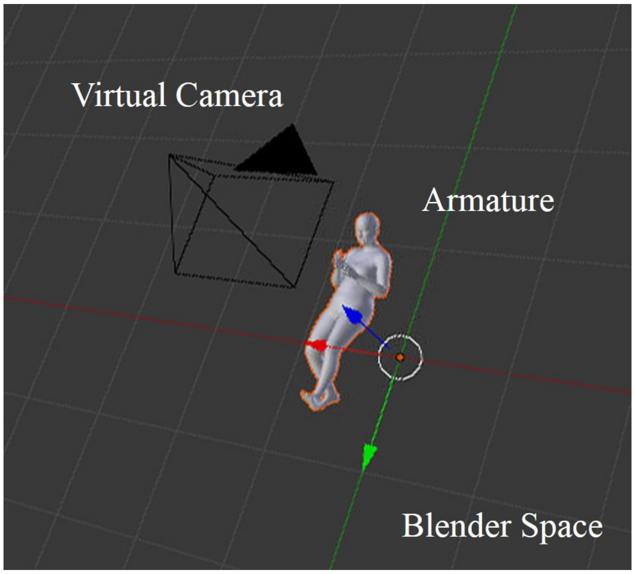


Figure B.2. Sythentic simulation space built on *Blender*. Amature action sequences based on SMPL model were adopted to simulate actual human motion, depth images are rendered by a virtual camera given by the active view planner. *Blender* space lies in special right-handed coordinate system, with (x, y, z) correspond to $(\mathtt{red}, \mathtt{green}, \mathtt{blue})$ arrows.

As illustrated in Fig. B.2, *Blender* [7] was used for rendering after body shape sequence animation. Depth images are rendered by a virtual *Blender* camera from poses given by active view planning module. To better simulate the characteristics of practical depth camera *Asus Xtion Pro*, both intrinsic parameters and frame rate of the *Blender* camera are forced to accord with practical camera's. It's worth noting that, *Z-pass*, the depth rendering technique in *Blender* evaluates absolute distance between vertices and camera center rather distance along z axis as in conventional depth image, so that post-processing is needed to obtain proper depth iamge sequences.

As to aerial robot simulation, maneuverability of the practical aerial robot is simulated by a speed control method proposed by *Gao et al.* [30]. Practical world space is defined identical to visualization *Rviz* space in simulation which will be discussed in Appx. B.2.

Depth camera is the key component which bridges reconstruction volume, *Blender* spaces and visualization *Rviz* space illustrated in Fig. B.3. Similar to the transformation relationship described in Sec. 3.2.3, the transformation from *Blender* to volume $\mathbf{T}_{b2v}$ is defined:

$$\mathbf{T}_{b2v} = \mathbf{T}_{d2v}(\mathbf{T}_{d2b})^{-1}. \tag{B.1}$$

where $\mathbf{T}_{d2v}$ is obtained from rigid-ICP algorithm, and $\mathbf{T}_{d2b}$ is the transformation from virtual camera to *Blender* spaces which can be easily obtained by the maintained transformation in visualization space described in Appx. B.2.

When next consecutive view point is ready which corresponds to the transformation $\mathbf{T}'_{d2v}$, the next virtual camera pose $\mathbf{T}'_{d2b}$ can be solved as

$$\mathbf{T}'_{d2b} = (\mathbf{T}_{b2v})^{-1}\mathbf{T}'_{d2v}. \tag{B.2}$$

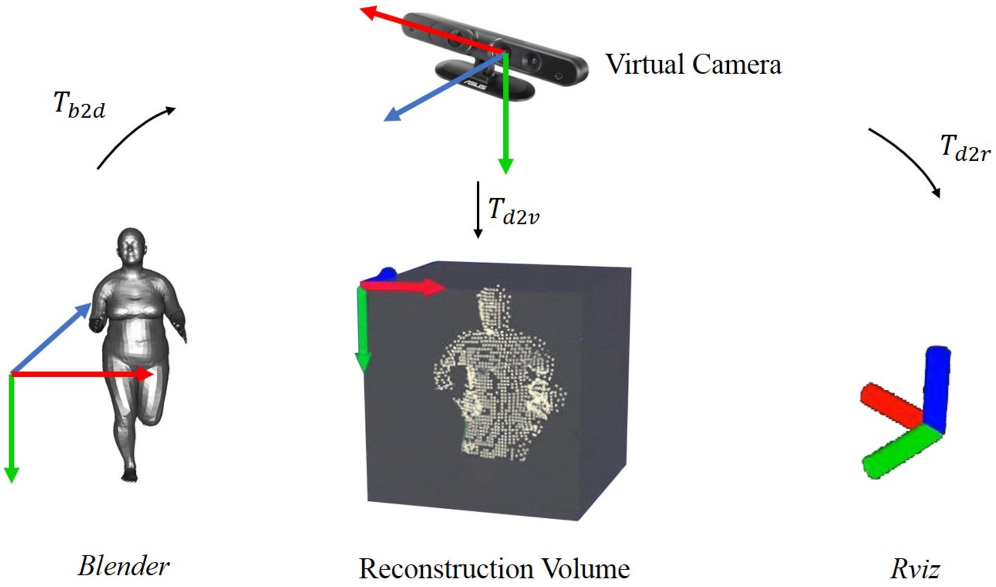


Figure B.3. Coordinate system in proposed work flow. All coordinate systems in ***iHuman3D*** and ***FlyFusion*** Camera-Based Coordinate System (CCS) expect for *Blender*. Axes (x, y, z) correspond to $(\mathtt{red}, \mathtt{green}, \mathtt{blue})$ arrows or bars. Transformation across spaces are connected via depth camera.

Also shown in Fig. B.3, the coordinates system in *Blender* obey a special right-handed coordinate whose x, z axes are opposite to common Camera-Based Coordinate System

(CCS)'s x, z axes. Note that all other modules including visualization platform in proposed systems listed in Fig. B.3 adopt the conventional CCS. So there exists such transformation between *Blender* to traditional computer vision system, we further specify it as the transformation between *Blender* depth camera to (conventional) depth camera system $\mathbf{T}_{\mathrm{bd2d}}$:

$$\mathbf{T}_{\mathrm{bd2d}} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{B.3}$$

Hence the transformation from *Blender* virtual camera to *Blender* space itself $\mathbf{T}_{\mathrm{bd2b}}$ can be written as

$$\mathbf{T}_{\mathrm{bd2b}} = \mathbf{T}_{\mathrm{d2b}}\mathbf{T}_{\mathrm{bd2d}} \tag{B.4}$$

B.2 Human Robot Visualization

As demonstrated in Fig. B.3, multiple components such as depth camera, reconstruction volume, real/virtual world space and etc. exist in proposed ***iHuman3D*** and ***FlyFusion*** system. Critical necessary of global assessment on reconstruction and motion capture in global coordinate system was raise. To address such demand, a human robot visualization platform is developed to transfer all components into a uniform coordinate system for further evaluation or potential applications, eg. free view point video. Moreover, volume drifting evaluation of dynamic scene in real 3D space is initially proposed discussed in Sec. thanks to the convenience of display in uniform space.

Robot Operate System (ROS) [66] provide abundant packages with functionalities on all-around aspects of robot manipulation, processing and visualization. Illustrated in Fig. B.4, we visualize above components in *Rviz*, a package in ROS. Specifically, flying camera, camera trajectory, live coarse reconstruction mesh and volume are displayed in the same space.

In simulation setup, *Blender* space and *Rviz* are rigidly assign to a constant transformation $\mathbf{T}_{\mathrm{r2b}}$ with the same origin which is identical to Eqn. B.3. Virtual odometry is generated by speed integration in the same rate in practical system.

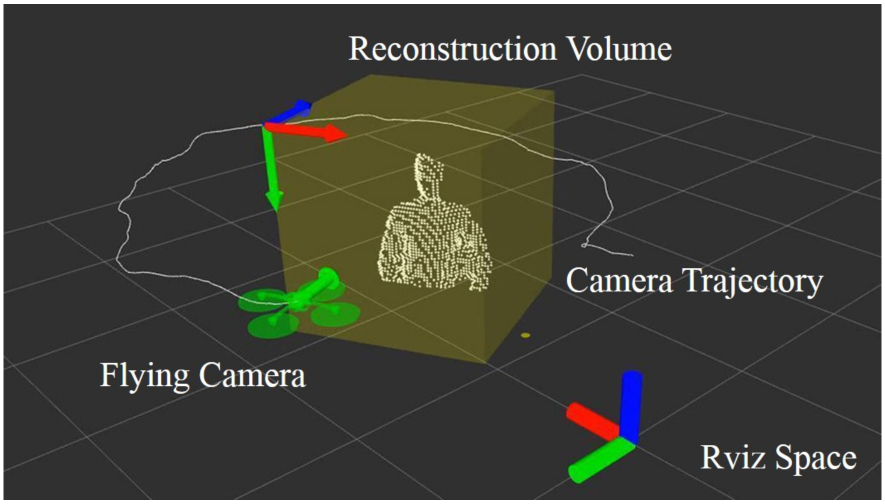


Figure B.4. Global visulazation space built on *Rviz*. *Rviz* space with (x, y, z) correspond to $(\mathtt{red}, \mathtt{green}, \mathtt{blue})$ bars. Reconstruction volume is shown in transparent yellow which lie in CCS, with (x, y, z) correspond to $(\mathtt{red}, \mathtt{green}, \mathtt{blue})$ arrows.

Specially, in practical setting, there still remain a certain gap between *Rviz* and real world North East Down (NED) coordinate system. To align both coordinates, we record the odometry $\mathbf{T}^0_{w2d}$ at initial moment when active view planning task begins. Hence fixed transformation between NED and visualization space $\mathbf{T}_{w2r}$ is denoted as

$$\mathbf{T}_{w2r} = \mathbf{T}^0_{d2r}\mathbf{T}^0_{w2d} \tag{B.5}$$

where $\mathbf{T}^0_{d2r}$ is a transformation contains only z axis translation which is the relative height to the ground h_0, specifically:

$$\mathbf{T}^0_{d2r} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{B.6}$$

Such transformation aligns practical working room to the *Rviz* with setting initial camera orientation as identity and force the initial fly camera pose as origin with a actual height drift. Note that although live quaternion represents a transformation from NED to body coordinate by definition, we ignore the tiny translation between aerial robot center to mounted RGB-D camera by directly treat it as $\mathbf{T}_{w2d}$.

Further, all components are mapped in *Rviz* via flying/virtual camera system by transformation relationship listed in Fig. B.3.