Abstract

Since hand pose estimation from depth image is intensely studied, this project presents a sphere based hand pose estimation method and a corresponding cost function for evaluation. The sphere hand model has 26 degrees of freedom and consists of 48 tight spheres in different radius. The cost function with optimal cost value 2.50097 indicates the high accuracy of the hand model. For further evaluation, 4 different sets of parameters are chosen randomly to prove the accuracy and robustness of the sphere based hand pose model.

Contents

Chapter 1

Introduction

Hand pose estimation is one of the most important parts in hand tracking technique and has been intensely studied for decades. Nevertheless, hand pose capture from depth image data is still a great challenging due to high freedoms and constrained parameter spaces. In this project, a sphere based likelihood function is supposed to be implemented. In addition, evaluation of the implementation using a public data set is needed.

In spite of works in[OKA10], hand model is base on cylindrical fingers with high accuracy, but it requires an expensive GPU for model rendering and cost function evaluation. This project realizes a sphere based hand model and implements a sphere based likelihood function for evaluation, which is inspired by[QSW$^+$14]. The public data provided in[TSLP14] is used for the sphere hand model calibration and final implementation evaluation.

This essay introduces a simple sphere hand model based on the forward kinematic chain and depth image data. It approximates real hand using a set of spheres. The hand skeleton is based on the forward kinematic chain using DH parameters and KDL library. Moreover, the depth image data is first converted to 3D point cloud and used for calibration of the spheres' radius. Finally, cost function is simply formed by measuring euclidean distance between the model and the point cloud and cost value is used to evaluate the accuracy of the estimation approach. These simplifications can be proved to be fast and efficient in reaching global optima in hand pose estimation. More details are given in Section.

Chapter 2

Model and Cost function

2.1 Build sphere hand model

2.1.1 Introduction of sphere hand model method

To model hand kinematics, this essay adopt the widely used 26 degrees of freedom (DOF) hand motion model in [OKA10]: six DOFs for the global transformation and four DOFs per finger. With a given data set of hand, the mainly task base on this hand motion model is to calibrate the precise shape and size of the hand. The hand kinematics model is illustrated in Fig. 2.1.

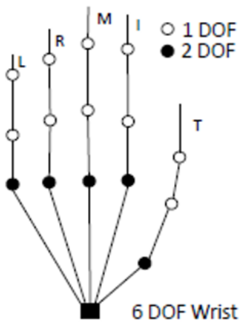
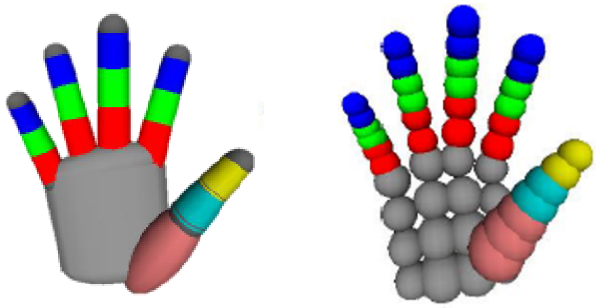


Figure 2.1: 26 DOFs kinematics hand motion model[OKA10]

These 26 motion parameters are denoted as Θ. According to the theory in[QSW$^+$14] and the example in[GPKT12], we can approximate the hand model using 48 spheres. This method simplifies real cylindrical fingers, illustrated in Fig. 2.2(a) into sphere joint chains, as illustrated in Fig. 2.2(b). The number of spheres for each part is manually specified: 6 for each finger (8 for the thumb) and 16 for the palm.



(a) cylindrical sphere hand model (b) Sphere hand model

The sphere model is denoted as $M(\Theta) = \{s_i\}_{i=1}^{48}$. Each sphere $s = \{c(\Theta), r\}$ has center $c(\Theta)$ and radius r. The center positions depend on the parameter Θ through forward kinematics. And spheres' radius can be manually fixed. In the sphere model, each sphere radius is large enough to overlap each other in order to full fill space between two neighboring joints and therefore contributes to a roughly approximation to a real finger.

2.1.2 Implementation of Sphere hand model

To simplify the task, the implementation of sphere hand model is divided into two steps: Build forward kinematic hand skeleton using only 26 base spheres. And add the other spheres into the hand skeleton. This section will introduce these two steps respectively:

- Build forward kinematic hand skeleton.
  Regarding KDL chain library, this work builds forward kinematic chains using DH parameters. DH parameters are four parameters associated with a particular convention for attaching reference frames to the links of a spatial kinematic chain. It can describe a six freedoms chain in space with only four parameters. In my hand pose estimation method, each finger is considered as one forward kinematic chain, and each chain has three joints with four freedoms(base joint has two freedoms). Each chain is independent from the others, this character enables us to build each chain separately. At the end, all chains are combined together.

  The forward kinematic hand skeleton is illustrated in Fig. 2.2. After adding angel offset, we will roughly realize skeleton with different hand gestures. The kinematics hand motion skeleton is the basement of final spheres hand model. We can roughly see the hand gesture from the skeleton.

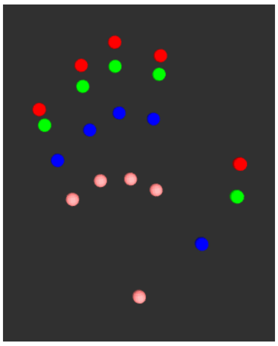


Figure 2.2: 26 DOFs kinematics hand motion skeleton

- Add the other spheres into the hand skeleton.
  In this essay, we regard spheres hand model as the approximation of the real hand. Nevertheless, using optimized personal hand model in[TSR⁺14] and sphere approximation technique in[WZS⁺06] should further improve the accuracy. In this work the spheres are evenly inserted between two joints. The 48 spheres hand model is illustrated in Fig. 2.3.

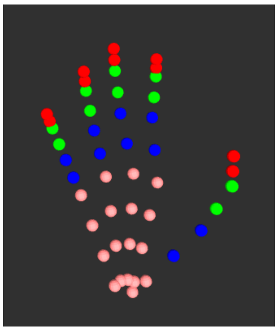


Figure 2.3: 48 spheres kinematics hand motion model

we can see the model is illustrated in a specific hand gesture. To make our model more like a real hand we can not leave space between spheres. That means the spheres should be more tighter. We can changing spheres' radius to achieve this goal.

Therefore, the next section is to calibrate sphere radius according to the 3D hand points cloud. This essay will first introduce how to get 3D hand points cloud from depth image and then go to the calibration work.

2.2 Hand segmentation and data set conversion

Another prepare work for this project is the depth image processing, which consists of hand segmentation and data set conversion:

- Hand segmentation.
  The original depth image with the whole human body includes too much information to proceed. To make the sphere based likelihood function evaluate more efficiently, hand segmentation is critical. With only hand segmentation information, we can calibrate parameters(sphere radius) of the sphere hand model faster. For this project, depth image from depth sensor is given as public data source, as illustrated in Fig. 2.4.

Figure 2.4: Depth image source

  In the public data sets, hand center and hand frame boundary coordinate in pixel scalar are also provided. we can scrap just one hand from the whole depth image.

- Data set conversion.
  In ROS environment, depth information can be easily converted into 3D coordinate information. The final results can be visualized in rviz. The 3D points cloud of hand segmentation is illustrated in Fig. 2.5.

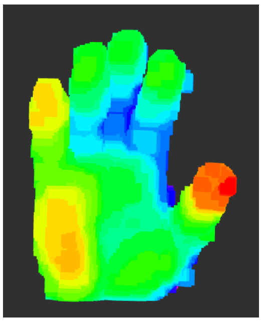


Figure 2.5: 3D hand points cloud

2.3 Determination of sphere hand model

The fist two sections have introduced prepare works for this project. As mentioned before, the radius of the sphere hand model need to be manually determined. Since the original sphere hand model is built based on the origin, which differs from the 3D points cloud, we need first to rotate the sphere hand model into 3D points cloud before setting radius values. The rotated result is illustrated in Fig. 2.6.

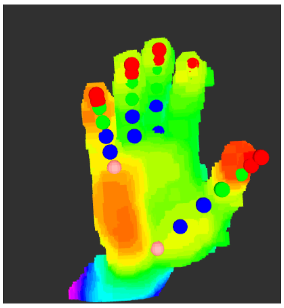
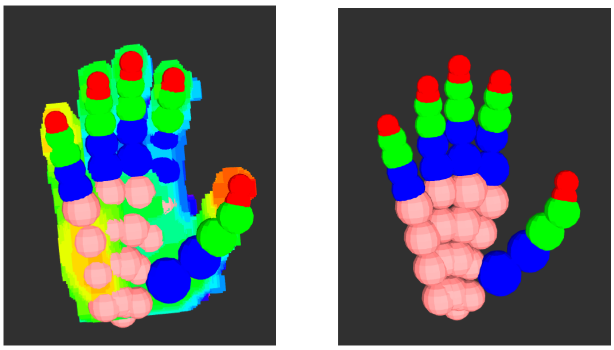


Figure 2.6: 3D hand points cloud

The next step is to estimate radius value of each spheres according to the 3D points cloud visualized in rviz. In my approach, every sphere is large enough to overlap with the neighboring spheres, which is more approximate to the real hand. The final result is illustrated in Fig. 2.7(a). Since the sphere hand model matches the 3D hand points cloud well, we can then extract the final sphere hand model, as illustrated in Fig. 2.7(b).



(a) Sphere radius calibration result

(b) Final 48 spheres kinematics hand motion model

2.4 Build fast cost function

Cost function is used to measure the discrepancy between the hand model and input depth, as well as hand model validity. Since the sphere model mentioned before is only a rough approximation to the 3D points cloud, the radius values still need to be calibrated by the cost function. In this project, cost function is simply defined as:

$$L = \sum_{p \epsilon sub(P)} D\left(p, s_{x(p)}\right)^2 \text{[QSW}^+\text{14]}. \tag{2.1}$$

Term $D(\cdot)$ matches points cloud P to the model M. To reduce the computational complexity, the term D is simply regarded as the euclidean distance from one point to its nearest sphere surface, which can be written as:

$$D(p, s) = abs\left(\|p - c\|_2 - r\right). \tag{2.2}$$

It has computational complexity $O(|M|\,|sub(P)|)$, as the nearest sphere for each point needs to be computed. Regardless of the collision effect between neighboring fingers, this cost function is simple and effective.

Chapter 3

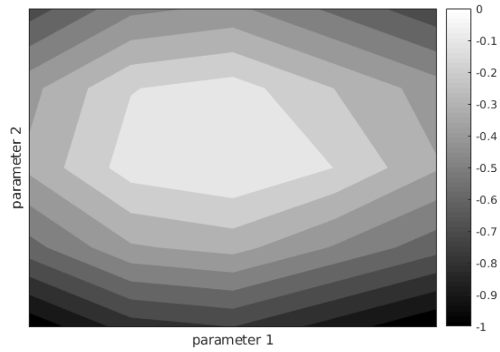
Evaluation

3.1 Evaluation of implementation

In this project, sphere based cost function is used for evaluating the sphere hand model. Lower cost value indicates higher accuracy in pose estimation. That means the optimal result is the minimum of the cost value. Therefor, I calibrate each sphere radius to the optimal value with the minimum cost. Before calibration, the cost value is 3.07591, and the final optimal cost value convergences to 2.50097. This result presents there is some acceptable deviation in the original sphere hand model. And after calibration, it has now high accuracy.

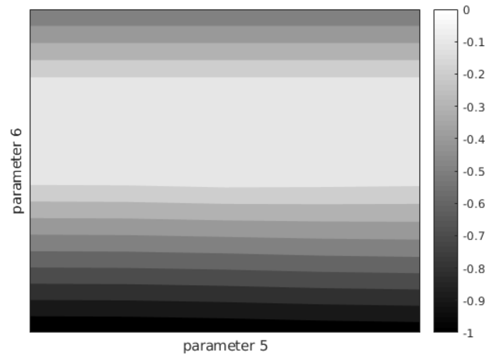
For further evaluation, this project choose randomly two independent initial joint angles and change the value of them(increase and decrease respectively in a determined gradient within 10 degree). Initial joint angle determines the hand model pose and it changes periodically in the value of π. This control variable method helps to figure out, whether the sphere hand model approach estimates given hand pose correctly. To make this evaluation result more powerful, four set of parameters are selected. Here is the situation where two initial joint angles are in one finger chain not taken into account, for they may not independent from each other.

As illustrated in Fig. 3.1, the cost value convergences to the optima value 2.50097 when both two coefficient are set in ground-truth value. In Fig. 3.1(a) and Fig. 3.1(b), the cost value diverges when any of the two coefficient deviates from the ground-truth value. In Fig. 3.1(c) and Fig. 3.1(d), the parameter 5 and parameter 7 seem have no effect with the cost value. Because different initial angle in different position have not the same influence on the cost value. If in the situation like in this project, when we change it in the same scalar, parameter which has small weight in cost function seem like having no contribution on the optimal cost value.

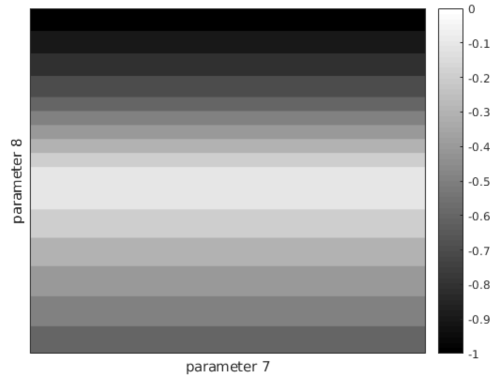


(a) Cost value according to parameter 1 and parameter 2

(b) Cost value according to parameter 3 and parameter 4



(c) Cost value according to parameter 5 and parameter 6



(d) Cost value according to parameter 7 and parameter 8

Figure 3.1: Cost value changing chart according to two random hand pose parameters

These figures prove that the sphere hand model presents the static hand gesture very well and also the cost function performances fast and effectively to evaluate the sphere based model. In the real experiment, there will be collisions between neighboring fingers and also between neighboring parameters on the same finger. Therefore the two random parameters are chosen from thumb finger and little finger separately. Further improvement of evaluation could be adding penalization norms to the cost function.

Chapter 4

Conclusion

This essay has implemented a sphere based hand pose estimation approach and has evaluated it using sphere based cost function. The tight sphere based hand pose model uses only 48 spheres to estimate hand pose. The small cost value of 2.50097 indicates the high accuracy of the method. The four charts of cost value show the robustness and high accuracy of the model. In conclusion, this simple estimation method and evaluation approach is useful for the future hand pose estimation or hand tracking works. Regarding the optimization method and the robustness of the cost function, penalization norms could be added in the cost function in the future.

List of Figures

Acronyms and Notations

HRC Human-Robot Collaboration

HRI Human-Robot Interaction

HRT Human-Robot Team

Bibliography

[GPKT12] Varun Ganapathi, Christian Plagemann, Daphne Koller, and Sebastian Thrun. Real-time human pose tracking from range data. In *European conference on computer vision*, pages 738–751. Springer, 2012.

[OKA10] Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Markerless and efficient 26-dof hand pose recovery. In *Asian Conference on Computer Vision*, pages 744–757. Springer, 2010.

[QSW+14] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1113, 2014.

[TSLP14] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):169, 2014.

[TSR+14] Jonathan Taylor, Richard Stebbing, Varun Ramakrishna, Cem Keskin, Jamie Shotton, Shahram Izadi, Aaron Hertzmann, and Andrew Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 644–651, 2014.

[WZS+06] Rui Wang, Kun Zhou, John Snyder, Xinguo Liu, Hujun Bao, Qunsheng Peng, and Baining Guo. Variational sphere set approximation for solid objects. *The Visual Computer*, 22(9-11):612–621, 2006.

License

Todo list