Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Stairs Reconstruction with 3D Point Cloud for Gait Generation of Lower Limb Exoskeleton Robot

Yachun Feng[1,2], Linqing Xia[1,2], Yong He[1,2], Can Wang[1,2], Zefeng Yan[1,2] and Xinyu Wu[1,2,3,*]
1. Guangdong Provincial Key Lab of Robotics and Intelligent System, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
2. CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institutes of Advanced Technology
3. Department of Mechanical and Automation Engineering,The Chinese University of Hong Kong
Email: (yc.feng, lq.xia, yong.he, can.wang zf.yan & xy.wu)@siat.ac.cn

***Abstract*— With the aging of the population, a considerable number of elderly persons suffer from walking disability. Lower limb exoskeleton robots have been proved to be effective in rehabilitation training and walking assistance. Though significant advancement has been made in the last decade, lacking the ability of adapting to complex terrain is one of the main challenges that restrict the wide application of exoskeleton robots. Thus we focus on the environmental perception and corresponding gait pattern generation of lower limb assistant exoskeleton. In this paper, we introduce a method to generate gait of exoskeleton walking on a stair, which applies the L-0 minimization-based acquisition and reconstruction of stairs information through 3D Point Cloud. The results of experiments show that our method can precisely measure the physical dimension information of stairs.**

***Index Terms*— Going upstairs, L-0 smooth, region growing, gait generation.**

I. Introduction

According to National Bureau of Statistics of China [1], the number of elderly persons over 65 has exceeded 166 million in 2018. The changing trend of the data display that it will continue to grow by more than 8 million annually. Some senior citizens lose the ability in autonomic walking because of disease and decline of muscle. Many commercial and research institutions have proposed the various types of lower-limb exoskeleton over the past few decades to enable these elderly persons to regain the ability of dependent life [2] - [7]. One of the effects of muscle atrophy is that activities as simple as uphill walking or stairs climbing become more difficult for senior citizens, even putting them at risk for falls. The invention of exoskeleton filled those gaps exactly up. Because of the close interaction between the exoskeleton and the wearer, it is necessary to develop a precise control strategy of the exoskeleton movement.

As a branch of the multi-degree-of-freedom robot, the first step of exoskeleton control is gait planning algorithm. Several classical gait trajectory planning methods have been studied in bipedal robots, such as the method based on three-dimensional inverted linear pendulum mode [8], the method based on cart-table model [9] and the ZMP preview control method [10]. Yong He et al. applied these algorithms in their developed Auto-LEE exoskeleton successfully [11]. Paper [12] proposed an individualized gait pattern generation method through a Gaussian process regression of gait data from a large number of healthy individuals can yet be regarded as a good way to do it, but the ground in the real environment is complex and changeable, therefore, this method must make its choice between the economy and security. Cyberdyne Ltd. [13] has developed surface EMG signals-based gait planning and control algorithm for the exoskeleton; the additional sensor system increases the inconvenience and financial costs of exoskeleton. How to acquire the ground information and convert it into the gait information of the exoskeleton is a research hotspot in this field.

Point cloud signals usually contain more shape information than other media such as pictures, it is also the most commonly used technique in spatial structure reconstruction. [14] and [15] have presented the method of identification and reconstruction of the indoor environment such as doors, walls and stairs with a point cloud camera. [16] proposed a method that uses the stereo vision and a point-cloud matching algorithm to accomplish modeling of terrain and planning of the robots movement.

Information about the point cloud are contained in the position and normal vector of each point that makes up it, it is very computationally expensive to do denoising and extract useful information of the point cloud. In [17], the authors proposed a segmentation algorithm of 3D Unorganized Point Cloud with region growing and [18] present an anisotropic point cloud denoising method using L-0 minimization.

In this paper, we introduce a method to generate Gait of exoskeleton on a stair, which applies the L-0 minimization-based acquisition and reconstruction of stairs information through 3D Point Cloud. After getting the stair's point cloud information through the depth camera, the point cloud will be

*This work is supported by the Shenzhen Robotics Research Center Project, Guangdong frontier and key technological innovation project (NO. 2017B090910013), the National Natural Science Foundation of China (No. U1613219) and the Shenzhen Overseas Innovation and Entrepreneurship Research Program, China (No. KQJSCX20170731164301774).

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

segmented into different regions according to spatial position and normal vector of points by region growing algorithm, then the noise in each point cloud segmentation will be removed by the denoising algorithm based on L-0 Minimization. By processing the point cloud with this method, the geometric information of the stairs can be obtained and the gait planning for the exoskeleton can be carried out finally through the gait generation algorithm.

This paper is organized as follows. Section II shows the region growing method, the smooth L-0 norm minimization and the gait planning strategy of our lower limb exoskeleton. Section III details the experimental setting and the results analysis. Finally, the conclusions of this work are presented in section IV.

II. Method

In this section, we discuss our geometric size measurement algorithm about stairs and gait planning strategy in detail.

A. Region growing

In the application of geometrical size calculation of stairs, the input unstructured point cloud belong to a stair surface, which actually can be segmented into numbers of planar structures. An effective region growing algorithm described in [17] is chosen to cluster those points into different planes. We use the *k-nearest-neighbor* (KNN) to find the neighboring points and estimate the normal vector via *singular value decomposition* (SVD) of each input point. We define $\mathbf{p}_i \in \mathbb{R}^{3\times 1}$ as an input point and $\mathbf{p}_{i,j}$ represents the j^{th} entry in the set of KNN of $\mathbf{p}_i$, $\mathbf{n}_i$ and $\mathbf{n}_{i,j}$ correspondingly as the normal vectors. Initially, all of the input points are marked as unprocessed. In every growth processing, region growing takes the first point $\mathbf{p}_s$ in the untreated point set as the seed point and add $\mathbf{p}_{s,j}$ to the current region $\mathbf{P}_s$ if the following constraints are satisfied:

$$\begin{array}{l} arccos\left(\left|\mathbf{n}_s^T\mathbf{n}_{s,j}\right|\right) \le \theta_{thresh} \\ \left|\mathbf{n}_s^T\left(\mathbf{p}_s-\mathbf{p}_{s,j}\right)\right| \le D_{thresh} \\ \left\|\mathbf{p}_s-\mathbf{p}_{s,j}\right\|_2 \le L_{thresh} \end{array} . \tag{1}$$

Once all the $\mathbf{p}_{s,j}\,(j=1,2,...,k)$ have been calculated by (1), $\mathbf{p}_s$ is marked as processed. A new seed $\mathbf{p}_s$ is chosen from the unprocessed input points to generate a new region until all the points in $\mathbf{P}_s$ are marked as processed. After the region growing procedure, the original input points are separated into several groups in which the points are approximately co-planar,as shown in Fig. 1.

B. Smooth L-0 norm minimization

By observing, we know that small size of point subset appearing at the edge of stairs. After region growing, the smooth L-0 norm minimization algorithm is used to smooth each region whose number of point is larger than a threshold N_{thresh}. An input region $\mathbf{P}=\begin{bmatrix}\mathbf{p}_1 & \mathbf{p}_2 & ... & \mathbf{p}_S\end{bmatrix}^T \in \mathbb{R}^{S\times 3}$ and normal matrix $\mathbf{N}=\begin{bmatrix}\mathbf{n}_1 & \mathbf{n}_2 & ... & \mathbf{n}_S\end{bmatrix}^T$

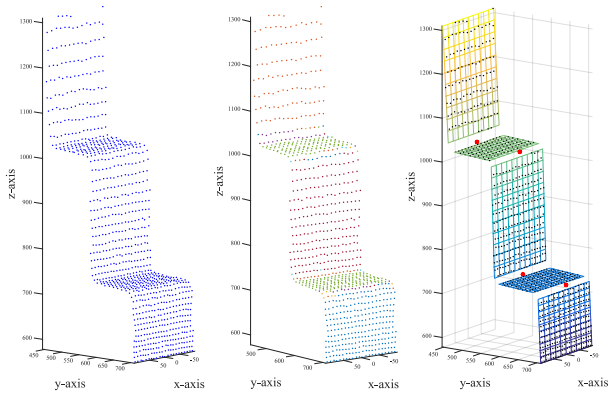

Fig. 1. From left to right, input point cloud collected by a depth camera, the result of region growing, the result of reposition and stair size calculation (the red dots represent the solution of (7)).

which is computed through KNN and SVD mentioned above. In this paper, the L-0 smoothing algorithm based on [18] consist of two parts.

Firstly, we smooth the normal of each point in the input region $\mathbf{P}$ by the object function:

$$\min_{\mathbf{N},\|\mathbf{n}_i\|_2=1} \left\|\mathbf{N}-\hat{\mathbf{N}}\right\|_2^2+\eta\left\|D(\mathbf{N})\right\|_0, \tag{2}$$

where $D(\mathbf{N})_{ik+j}=\mathbf{n}_i-\mathbf{n}_{ik+j}$, subscript $(ik+j)$ means the j^{th} elements in a collection of KNN of point $\mathbf{p}_i$ and $\hat{\mathbf{N}}$ is the initial value of $\mathbf{N}$, $D(\mathbf{N})$ can approximatively measure the gradient of each point in the KNN from the noisy sampling results from a real surface. As we all know that L-0 norm minimization is an NP problem. But by introducing an auxiliary variable $\boldsymbol{\theta}\in\mathbb{R}^{S\times 3k}$, we can get an approximate solution to this problem. The objective function (2) can be rewritten as following:

$$\min_{\boldsymbol{\theta},\mathbf{N},\|\mathbf{n}_i\|_2=1} \left\|\mathbf{N}-\hat{\mathbf{N}}\right\|_2^2+\beta\left\|D(\mathbf{N})-\boldsymbol{\theta}\right\|_2^2+\eta\left\|\boldsymbol{\theta}\right\|_0. \tag{3}$$

The objective function (3) can be solved by alternating optimization method described as following.

Step 1: Optimization of $\boldsymbol{\theta}$

Fixing the normal matrix $\mathbf{N}$, then optimize the auxiliary variable $\boldsymbol{\theta}$ via objective function:

$$\min_{\boldsymbol{\theta}} \beta\left\|D(\mathbf{N})-\boldsymbol{\theta}\right\|_2^2+\eta\left\|\boldsymbol{\theta}\right\|_0. \tag{4}$$

The solution of above function is given by $\boldsymbol{\theta}_{ik+j}=0$, if $\left\|D(\mathbf{N})_{ik+j}\right\|_2^2<\frac{\eta}{\beta}$, or $\boldsymbol{\theta}_{ik+j}=D(\mathbf{N})_{ik+j}$.

Step 2: Optimization of N

Fixing the auxiliary variable $\boldsymbol{\theta}$, then optimize the normal matrix $\mathbf{N}$ via objective function:

$$\min_{\mathbf{N},\|\mathbf{n}_i\|_2=1} \left\|\mathbf{N}-\hat{\mathbf{N}}\right\|_2^2+\beta\left\|D(\mathbf{N})-\boldsymbol{\theta}\right\|_2^2. \tag{5}$$

We use the subscript "*vec*" to represent the vectorization of a matrix, for example,$\mathbf{P}_{vec}=\begin{bmatrix}\mathbf{p}_1^T & \mathbf{p}_2^T & ... & \mathbf{p}_S^T\end{bmatrix}^T \in$

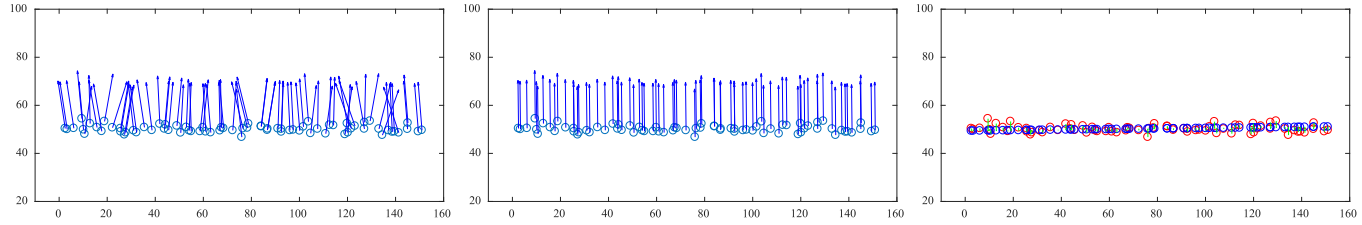

Fig. 2. Simulation results of the L-0 smoothing algorithm. From left to right: the normal vectors of the input points corrupted with Gaussian noise ($\sigma = 2$), smoothed normal vectors by our method, result of reposition procedure.

$\mathbb{R}^{3S\times 1}$ as the vectorization form of the input matrix $\mathbf{P} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \dots & \mathbf{p}_S \end{bmatrix}^T \in \mathbb{R}^{S\times 3}$. The KNN indices of each point compose a matrix

$$\mathbf{K} = \begin{bmatrix} \dots & \dots & \dots \\ \dots & \mathbf{K}_{ik+j} & \dots \\ \dots & \dots & \dots \end{bmatrix} \in \mathbb{R}^{S\times k},$$

where $1 \le \mathbf{K}_{ik+j} \le S (i = 1,2,...,S; j = 1,2,...,k)$. Then we define a partitioned matrix $\mathbf{C}_j \in \mathbb{R}^{3S\times 3S}$ whose submatrix is either $\mathbf{0}$ matrix or unit matrix. The location of the non-zero block in each row of $\mathbf{C}_j$ is decided by $\mathbf{K}_{ik+j}$ and denoted by $\mathbf{C}_j\left(i, \mathbf{K}_{ik+j}\right) = \mathbf{I}_3$. According to the definition of $D(\mathbf{N})$ and $\mathbf{C}_j$, we can rewrite (5) as:

$$\min_{\mathbf{N}_{vec}, \|\mathbf{N}_i\|_2=1} \left\|\mathbf{N}_{vec} - \hat{\mathbf{N}}_{vec}\right\|_2^2 + \beta \sum_{j=1}^{k} \left\|\mathbf{A}_j \mathbf{N}_{vec} - \boldsymbol{\theta}_{:k+j}\right\|_2^2, \quad (6)$$

where $\mathbf{A}_j = \mathbf{I} - \mathbf{C}_j$. The solution vector of (6) can be obtained by

$$\mathbf{N}_{vec} = \left(\mathbf{I} + \beta \sum_{j=1}^{k} \mathbf{A}_j^T \mathbf{A}_j\right)^{-1} \left(\hat{\mathbf{N}}_{vec} + \beta \sum_{j=1}^{k} \mathbf{A}_j^T \boldsymbol{\theta}_{:k+j}\right).$$

The (4) and (5) are alternated minimization and parameter β is updated by multiplying a constant κ_1 in each iteration until its value reaches to $\beta_{\max}$.

Secondly, we reposition each input point along its respective normal which has been optimized by (3) through the following object function:

$$\min_{\mathbf{P}} \left\|\mathbf{P} - \hat{\mathbf{P}}\right\|_2^2 + \delta \|D(\mathbf{P})\|_0, \quad (7)$$

where $D(\mathbf{P}) \in \mathbb{R}^{S\times k}$ and $D(\mathbf{P})_{ik+j} = \left(\mathbf{p}_i - \mathbf{p}_{ik+j}\right)^T \mathbf{n}_i \in \mathbb{R}$ measure the deviation from a KNN point to the plane defined by an input point and respective normal. Just like the solving method used to (2), object function (7) also can be approximately solved by introducing another auxiliary variable $\mathbf{W}$ which has the same dimension as $D(\mathbf{P})$. Then the objective function can be modified as:

$$\min_{\mathbf{P},\mathbf{W}} \left\|\mathbf{P} - \hat{\mathbf{P}}\right\|_2^2 + \lambda \left\|D(\mathbf{P}) - \mathbf{W}\right\|_2^2 + \delta \|\mathbf{W}\|_0. \quad (8)$$

Step 3: Optimization of W

Fixing the point matrix $\mathbf{P}$, then optimize the auxiliary variable $\mathbf{W}$ via objective function:

$$\min_{\mathbf{W}} \lambda \left\|D(\mathbf{P}) - \mathbf{W}\right\|_2^2 + \delta \|\mathbf{W}\|_0. \quad (9)$$

If $\left\|D(\mathbf{P})_{ik+j}\right\|_2^2 < \delta/\lambda$, $\mathbf{W}_{ik+j} = 0$, else $\mathbf{W}_{ik+j} = D(\mathbf{P})_{ik+j}$.

Step 4: Optimization of P

Fixing the auxiliary variable $\mathbf{W}$, then optimize the point matrix $\mathbf{P}$ via objective function:

$$\min_{\mathbf{P}} \left\|\mathbf{P} - \hat{\mathbf{P}}\right\|_2^2 + \lambda \left\|D(\mathbf{P}) - \mathbf{W}\right\|_2^2, \quad (10)$$

The reposition of each point is restricted to along its normal direction

$$\mathbf{p}_i = \hat{\mathbf{p}}_i + \alpha_i \mathbf{n}_i. \quad (11)$$

The (11) can be rewritten as:

$$\mathbf{p}_i = \hat{\mathbf{p}}_i + diag\left(\mathbf{n}_{i,vec}\right) \boldsymbol{\alpha}_{i,vec}, \quad (12)$$

where $\boldsymbol{\alpha}_{i,vec} = \begin{bmatrix} \alpha_i & \alpha_i & \alpha_i \end{bmatrix}^T$. So we can optimize vector $\boldsymbol{\alpha}_{vec}$ instead of optimizing $\mathbf{P}$, where $\boldsymbol{\alpha}_{vec} = \begin{bmatrix} \alpha_1 & \alpha_1 & \alpha_1 & \dots & \alpha_S & \alpha_S & \alpha_S \end{bmatrix}^T \in \mathbb{R}^{3S\times 1}$.
The expanded form of the first item in (10) is

$$\begin{aligned} \left\|\mathbf{P} - \hat{\mathbf{P}}\right\|_2^2 &= \left\|\hat{\mathbf{P}}_{vec} + diag\left(\mathbf{N}_{vec}\right) \boldsymbol{\alpha}_{vec} - \hat{\mathbf{P}}_{vec}\right\|_2^2 \\ &= \boldsymbol{\alpha}_{vec}^T diag(\mathbf{N}_{vec})^T diag\left(\mathbf{N}_{vec}\right) \boldsymbol{\alpha}_{vec}. \end{aligned}$$

The expanded form of the second item in (10) is

$$\begin{aligned} &\lambda \left\|D(\mathbf{P}) - \mathbf{W}\right\|_2^2 \\ =&\lambda \sum_{i=1}^{S} \sum_{j=1}^{k} \left\|D(\mathbf{P})_{ik+j} - \mathbf{W}_{ik+j}\right\|_2^2 \\ =&\lambda \sum_{i=1}^{S} \sum_{j=1}^{k} \left(\mathbf{p}_i - \mathbf{p}_{ik+j} - \mathbf{W}_{ik+j}\mathbf{n}_i\right)^T \mathbf{n}_i \mathbf{n}_i^T \\ &\left(\mathbf{p}_i - \mathbf{p}_{ik+j} - \mathbf{W}_{ik+j}\mathbf{n}_i\right) \\ =&\lambda \sum_{j=1}^{k} \left(\mathbf{A}_j \boldsymbol{\alpha}_{vec} + \mathbf{B}_j\right)^T \mathbf{N}_N \left(\mathbf{A}_j \boldsymbol{\alpha}_{vec} + \mathbf{B}_j\right), \end{aligned}$$

where

$$\mathbf{A}_j = diag\left(\mathbf{N}_{vec}\right) - diag\left(\mathbf{C}_j \mathbf{N}_{vec}\right) \mathbf{C}_j,$$

$$\mathbf{B}_j = \left(\mathbf{I} - \mathbf{C}_j\right) \hat{\mathbf{P}}_{vec} - diag\left(\mathbf{N}_{vec}\right) \mathbf{W}_{:k+j,vec},$$

$$\mathbf{W}_{:k+j,vec} = \left[\mathbf{W}_{1k+j}\,\mathbf{W}_{1k+j}\,\mathbf{W}_{1k+j}\ldots\mathbf{W}_{Sk+j}\,\mathbf{W}_{Sk+j}\,\mathbf{W}_{Sk+j}\right]^T,$$

$\mathbf{W}_{:k+j,vec} \in \mathbb{R}^{3S\times 1}$ and

$$\mathbf{N}_N = \begin{bmatrix} \mathbf{n}_1\mathbf{n}_1^T & 0 & 0 \\ 0 & \ldots & 0 \\ 0 & 0 & \mathbf{n}_S\mathbf{n}_S^T \end{bmatrix} \in \mathbb{R}^{3S\times 3S},$$

The solution vector of (10) can be obtained by

$$\boldsymbol{\alpha}_{vec} = \left\{ diag(\mathbf{N}_{vec})^T diag\,(\mathbf{N}_{vec}) + \lambda \sum_{j=1}^{k} \mathbf{A}_j^T \mathbf{N}_N \mathbf{A}_j \right\}^{-1} \left\{ -\lambda \sum_{j=1}^{k} \mathbf{A}_j^T \mathbf{N}_N \mathbf{B}_j \right\}.$$

The (9) and (10) are alternatively minimized and parameter λ is updated by multiplying κ_2 in each iteration until its value reaches to $\lambda_{\max}$. Fig. 2 shows the result of the algorithm on 2D simulation data corrupted by Gaussian noise.

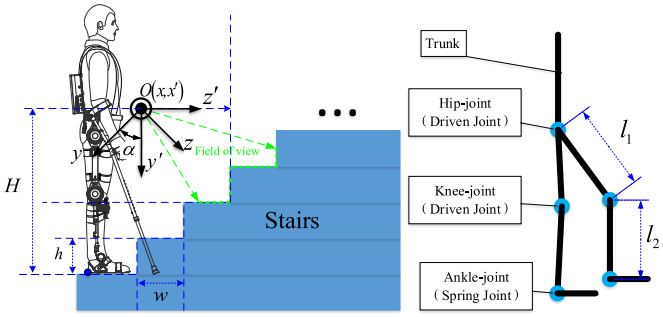

Fig. 3. A simple block diagram of the wearer walking on stairs with the exoskeleton.

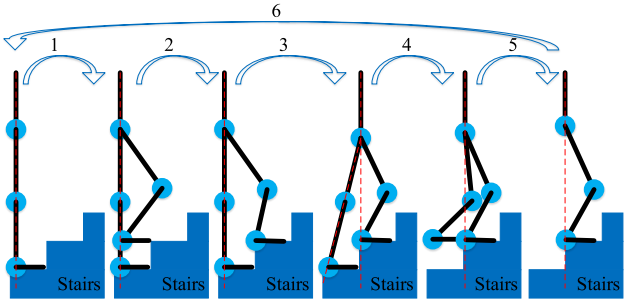

Fig. 4. From left to the right, corresponding to the 6 phases of gait pattern about going up the stairs. Right and left leg are represented by bule and red respectively.

C. Stair size calculation

We define $\mathbf{P}_i(i = 1,2,...,r)$ to represent each smoothed region and use $\mathbf{c}_i$ to describe the centroid of $\mathbf{P}_i$ and calculate a corresponding normal vector $\mathbf{n}_i$ of each $\mathbf{P}_i$ via the Least Square Regression. Based on prior knowledge, $\mathbf{n}_i$ and $\mathbf{c}_i$ can be reordered by the distance between $\mathbf{c}_i$ and the origin of the camera coordinate system along the y-axis. The sorting operation allows the mutually perpendicular normal vectors stored alternately. To obtain the width and height of each stair, we define a projection plane which is vertical with the mutually perpendicular surface of the stair and through the origin of camera coordination system. The following formula calculates the normal vector of the plane

$$\mathbf{n}_p = \frac{1}{r-1}\sum_{i=1}^{r-1} \mathbf{n}_i \times \mathbf{n}_{i+1}, \tag{13}$$

where $\mathbf{n}_i(i = 1,2,...,r)$ has been reordered by y-axis component of $\mathbf{c}_i(i = 1,2,...,r)$. Therefore, the angle points (As shown in Fig. 1.) of the step can be deduced by

$$\begin{cases} \mathbf{n}_i^T(\mathbf{x}_j - \mathbf{c}_i) & = 0 \\ \mathbf{n}_{i+1}^T(\mathbf{x}_j - \mathbf{c}_{i+1}) & = 0 \\ \mathbf{n}_p^T\mathbf{x}_j & = 0 \end{cases} \tag{14}$$

where $i = 1,2,...,r-1$. The distance between $\mathbf{x}_j$ and $\mathbf{x}_{j+1}$ $(j = 1,2,...,r-2)$ is exactly the width or height of the stairs we need. The width and height of the stairs are indicated by w and h.

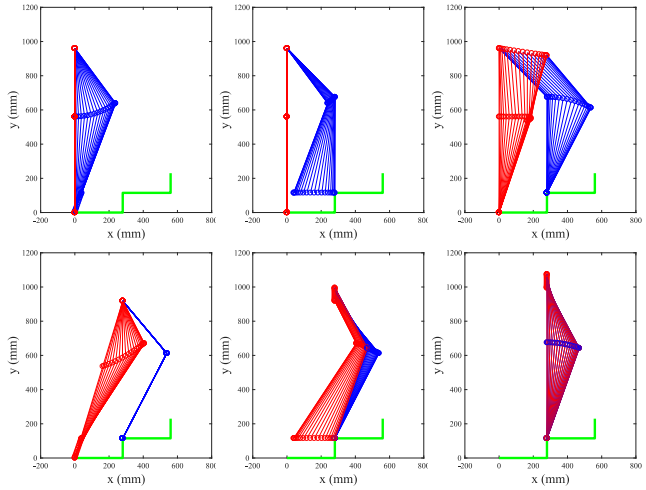

Fig. 5. From left to the right, corresponding to the 6 phases of gait pattern about going up the stairs. Right and left leg are represented by bule and red respectively.

D. Gait pattern generation

Fig. 3 illustrates the working environment of the method proposed in this article. To obtain smooth transition trajectories and continuous control parameters of each joint of the exoskeleton, we use quantic polynomial for path planning in the joint space, shown as (15).

$$\begin{cases} P_f = m_0 + m_1 t + m_2 t^2 + m_3 t^3 + m_4 t^4 + m_5 t^5 \\ \dot{P}_f = m_1 + 2m_2 t + 3m_3 t^2 + 4m_4 t^3 + 5m_5 t^4 \\ \ddot{P}_f = 2m_2 + 6m_3 t + 12m_4 t^2 + 20m_5 t^3 \end{cases} \tag{15}$$

While the boundary conditions of kinematic parameters including position, velocity and acceleration are given, we can get the values of the coefficient in (15). A complete gait cycle of going up the stair can be separated into 6 phases, as shown in Fig. 4. The complete transition trajectories of joints can be calculated by the constraints about planar double bar linkage mechanism.

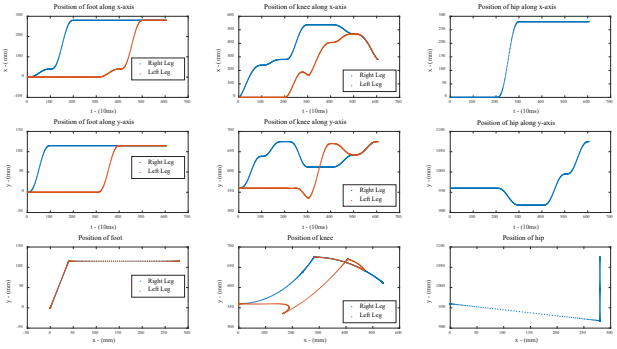

Fig. 6. Position parameters of driving joints.

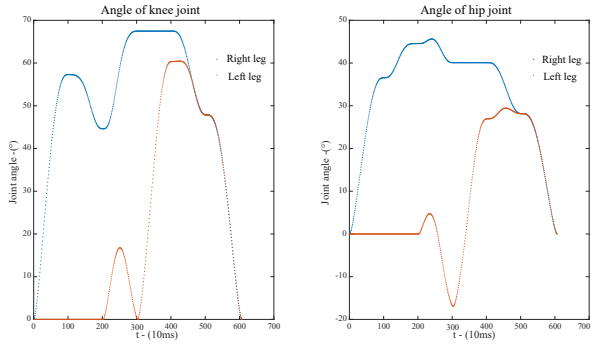

Fig. 7. Angle parameters of driving joints.

III. Experiments and Analysis

The proposed geometric size measurement algorithm for stairs is tested on a dataset which is collected in our laboratory environment by a depth camera (PERCIPIO.XYZ FM830) rigidly mounted on the chest of the exoskeleton without person. According to the results of measurement, a serial of simulations of the gait generation experiment are performed respectively. Our approach is implemented in Matlab and the code of stairs measurement algorithm can be download at: (Github: https://github.com/fengyachun/measurement).

Both in Section II, the coordinates of the point cloud are in the camera coordinates system. The camera we use to collect the point cloud is rigidly attached on near the chest of the exoskeleton. It will bring computational convenience and intuitiveness to transfer point clouds to the ground coordinate system. In the application scenario covered in this article, a single degree of freedom Euler matrix can complete the coordinate transformation and the Euler angles α can be obtained through the IMU.

A. Parameter Setting

The angle threshold between normal vectors in region growing is set to $\theta_{thresh} = 10^{\circ}$. The ohter two threshold parameters are set to $D_{thresh} = \|\mathbf{p}_s - \mathbf{p}_{sk+3}\|_2$ and $L_{thresh} = 10 * D_{thresh}$. The number of KNN set to $k = 9$ both in region growing and smooth L-0 norm minimization. The threshold of the point number of the region used to L-0 minimization set to $N_{thresh} = 30$. The initial parameters of normal vector smoothing and point reposition set to $\beta = 1$, $\eta = 0.3$, $\lambda = 1$, $\delta = 10$, and the alternating optimizations are repeated with $\beta = 1.2\beta$ and $\lambda = 1.5\lambda$ until their values reach to $\beta_{\max}$ and $\lambda_{\max}$. The length of the thigh and shank of the exoskeleton are $l_1 = 400$mm and $l_2 = 560$mm.

B. Experiment results

Firstly, we change the mounting location and angle of the camera to verify the accuracy and robustness of the measurement algorithm. The height of the mounting location represented by H is ranging from 1000mm to 1150mm at 50mm interval and the mounting angle α is ranging from 27° to 45° at 6° interval. The total number of the test point cloud is 16. The results of geometric size measurement algorithm about stairs are shown in Table I-II. The ground truth of height and width of the stairs used in our lab is 115mm and 280mm respectively. We can see that our algorithm can precisely measure the physical dimension of the stairs. During the experiment, we find that the most critical parameter is the N_{thresh}. Too small N_{thresh} will lead to the region like the angle of stair being regarded as a planar structure, which further leads to miscalculations about the size of stairs. The N_{thresh} is related to the spatial resolution of the point cloud.

Then, according to the results of measurement, we generate the gait patterns of going up the stairs as shown in Fig. 5. The boundary conditions of (15) set as Table III. All of the velocities and accelerations should set as 0. To avoid the interference between the tiptoe and angle of stairs, the length of feet which is represented by ε must be considered. We use the constraint of a planar double bar linkage mechanism to calculate the trajectory of knee joints and generate the control parameters of each driving joint, as shown in Fig. 6 and Fig. 7.

IV. Conclusion

In this paper, we proposed a geometric size measurement algorithm for stairs which can precisely measure the physical dimension of the stairs. We use region growing algorithm to cluster the unstructured point cloud and smooth each region by L-0 norm minimization. We also contribute two closed-form expressions for the quadratic sub-problem of

TABLE I

THE RESULTS OF THE MEASUREMENT ALGORITHM ABOUT STAIRS

	$H=1000$ $\alpha=27^\circ$	$H=1000$ $\alpha=33^\circ$	$H=1000$ $\alpha=39^\circ$	$H=1000$ $\alpha=45^\circ$	$H=1050$ $\alpha=27^\circ$	$H=1050$ $\alpha=33^\circ$	$H=1050$ $\alpha=39^\circ$	$H=1050$ $\alpha=45^\circ$
h	114.7	114.2	113.7	114.5	116.9	115.6	116.9	115.6
w	278.1	277.2	277.6	278.5	276.3	277.2	277.2	278.1

TABLE II

THE RESULTS OF THE MEASUREMENT ALGORITHM ABOUT STAIRS

	$H=1100$ $\alpha=27^\circ$	$H=1100$ $\alpha=33^\circ$	$H=1100$ $\alpha=39^\circ$	$H=1100$ $\alpha=45^\circ$	$H=1150$ $\alpha=27^\circ$	$H=1150$ $\alpha=33^\circ$	$H=1150$ $\alpha=39^\circ$	$H=1150$ $\alpha=45^\circ$
h	116.5	113.3	116.6	114.8	113.4	115.4	114.6	114.1
w	275.6	278.8	277.0	277.4	278.4	278.2	277.0	278.1

TABLE III

THE BOUNDARY CONDITIONS(L=l_1+l_2)

	t_0	t_1	t_2	t_3	t_4	t_5	t_6
$P_{fr}(t)$	(0,0)	$(w-\varepsilon,h)$	(w,h)	(w,h)	(w,h)	(w,h)	(w,h)
$P_{fl}(t)$	(0,0)	(0,0)	(0,0)	(0,0)	$(w-\varepsilon,h)$	(w,h)	(w,h)
$P_h(t)$	$(0,L)$	$(0,L)$	$(0,L)$	$\left(w,\sqrt{L^2-w^2}\right)$	$\left(w,\sqrt{L^2-w^2}\right)$	$\left(w,\frac{2}{3}h+\sqrt{L^2-w^2}\right)$	$(w,h+L)$

the original smooth L-0 norm minimization algorithm. The results of measurement can be directly used to generate the gait pattern of going up the stairs. In our future work, we will research the problem of going down-stairs.

ACKNOWLEDGMENT

The authors would like to thank anyone who contributed to this paper.

REFERENCES

[1] National Bureau of Statistics of China, Peking, China,"CHINA STATISTICAL YEARBOOK",2019. [Online]. Available: http://www.stats.gov.cn/tjsj/ndsj/2018/indexeh.htm/

[2] Suzuki, Kenta, et al. "Intention-based walking support for paraplegia patients with Robot Suit HAL." Advanced Robotics 21.12 (2007): 1441-1469.

[3] Dollar, Aaron M., and Hugh Herr. "Design of a quasi-passive knee exoskeleton to assist running." 2008 IEEE/RSJ international conference on intelligent robots and systems. IEEE, 2008.

[4] Tsagarakis, Nikolaos G., and Darwin G. Caldwell. "Development and control of a soft-actuated exoskeleton for use in physiotherapy and training." Autonomous Robots 15.1 (2003): 21-33.

[5] Kong, Kyoungchul, and Doyoung Jeon. "Fuzzy control of a new tendon-driven exoskeletal power assistive device." Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2005). 2005.

[6] Wang, Can, et al. "A Flexible lower extremity exoskeleton robot with deep locomotion mode identification." Complexity 2018 (2018).

[7] Wang, Can, et al. "Implementation of a brain-computer interface on a lower-limb exoskeleton." IEEE Access 6 (2018): 38524-38534.

[8] Wu, Xinyu, et al. "Individualized Gait Pattern Generation for Sharing Lower Limb Exoskeleton Robot." IEEE Transactions on Automation Science and Engineering 99 (2018): 1-12.

[9] Kajita S , Kanehiro F , Kaneko K , et al. The 3D linear inverted pendulum model: A simple modeling for a biped walking pattern generation[C]// Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2001.

[10] Kajita S . Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point[C]// Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan. IEEE, 2003.

[11] Shimmyo S , Sato T , Ohnishi K . Biped Walking Pattern Generation by Using Preview Control Based on Three-Mass Model[J]. IEEE Transactions on Industrial Electronics, 2013, 60(11):5137-5147.

[12] He Y, Li N, Wang C, et al. Development of a novel autonomous lower extremity exoskeleton robot for walking assistance[J]. Frontiers of Information Technology and Electronic Engineering, 2019, 20(3): 318-329.

[13] CYBERDYNE Inc., Tsukuba, Japan, HAL for Labor Support/Care Support (Lumbar Type), 2014. [Online]. Available: https:// www.cyberdyne.jp/english/

[14] Souto, Leonardo, et al. "Stairs and Doors Recognition as Natural Landmarks Based on Clouds of 3D Edge-Points from RGB-D Sensors for Mobile Robot Localization." Sensors 17.8 (2017): 1824.

[15] Munoz, Rai, Xuejian Rong, and Yingli Tian. "Depth-aware indoor staircase detection and recognition for the visually impaired." 2016 IEEE international conference on multimedia & expo workshops (ICMEW). IEEE, 2016.

[16] Kolter, J. Zico, Youngjun Kim, and Andrew Y. Ng. "Stereo vision and terrain modeling for quadruped robots." 2009 IEEE International Conference on Robotics and Automation. IEEE, 2009.

[17] Lu, Xiaohu, Yahui Liu, and Kai Li. "Fast 3D Line Segment Detection From Unorganized Point Cloud." arXiv preprint arXiv:1901.02532 (2019).

[18] Sun, Yujing, Scott Schaefer, and Wenping Wang. "Denoising point sets via L0 minimization." Computer Aided Geometric Design 35 (2015): 2-15.