

高分辨率三维扫描系统说明

广东工业大学 视觉实验室 2024 年 7 月

1. 设备说明

1.1. 概述

高分辨率三维扫描系统主要由工业级二维激光雷达和分度仪组成。扫描过程如下：二维激光雷达扫描垂直平面，得到物体表面的距离 d 和反射光强 i 以及垂直方向角 θ ；分度仪水平旋转，得到水平方向角 ϕ 。每个扫描点的激光雷达数据 (d, θ, ϕ, i) 转换为点云的点 (x, y, z, b) ，其中 b 为亮度，取值 $0 \sim 1$ 。

本系统主要用于精细三维建模，类似于高分辨率相机，可用于室内、洞穴等环境或建筑、桥梁等大型对象的三维建模与几何测量。

原型系统硬件部分如图 1 所示，具有以下特点，详细指标见表 1：

- 垂直方向的大视角（ 270° ）和高分辨率（ 0.25° ）扫描。
- 水平方向的全视角（ 360° ）和超高分辨率（最高 0.005° ）扫描。
- 可获取物体表面纹理。
- 便携结构，内置电池，用 WiFi 无线连接，无需连线。

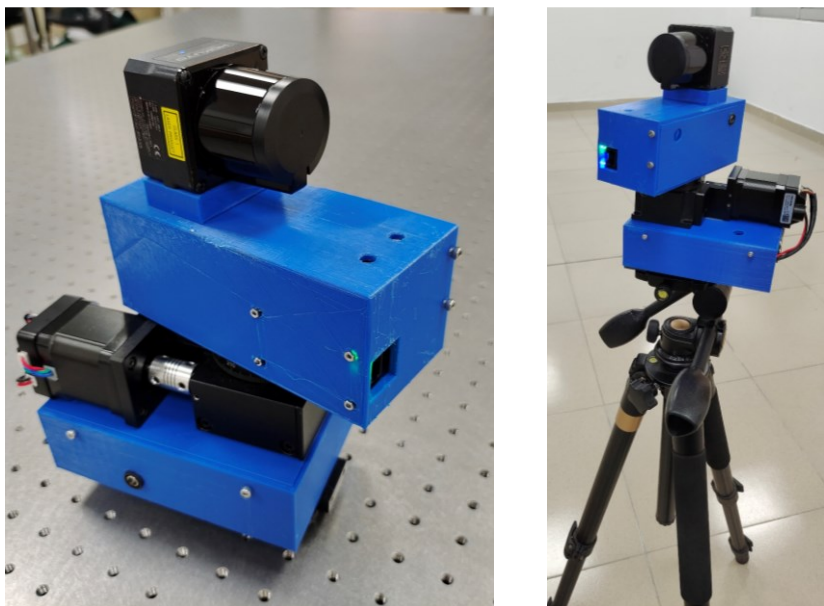


图 1 高分辨率三维扫描系统

表 1 高分辨率三维扫描系统硬件指标

指标名称	指标数值
垂直方向角度范围	270°
垂直方向角分辨率	0.25°
垂直方向点数	1080
水平方向角度范围	360°
水平方向角分辨率	0.005°~0.4°
水平方向点数	900~72000
水平扫描速度	0.01°/s ~ 15°/s
扫描距离	<60m
数据传输率	40000 点/秒
数据格式	距离+反射光强
连接方式	300M WiFi
工作电压	12V DC

软件系统包括硬件系统控制程序和实时 3D 显示程序两部分，界面如图 2 所示，具有以下特点：

- 实时显示扫描过程。
- 支持多台扫描系统同时扫描。
- 扫描过程用指令序列控制，可编程自主扫描。
- 可读写数据格式：L3D（原始数据）、PCD（PCL 点云库）、PLY（三角网格）。
- 开放接口。

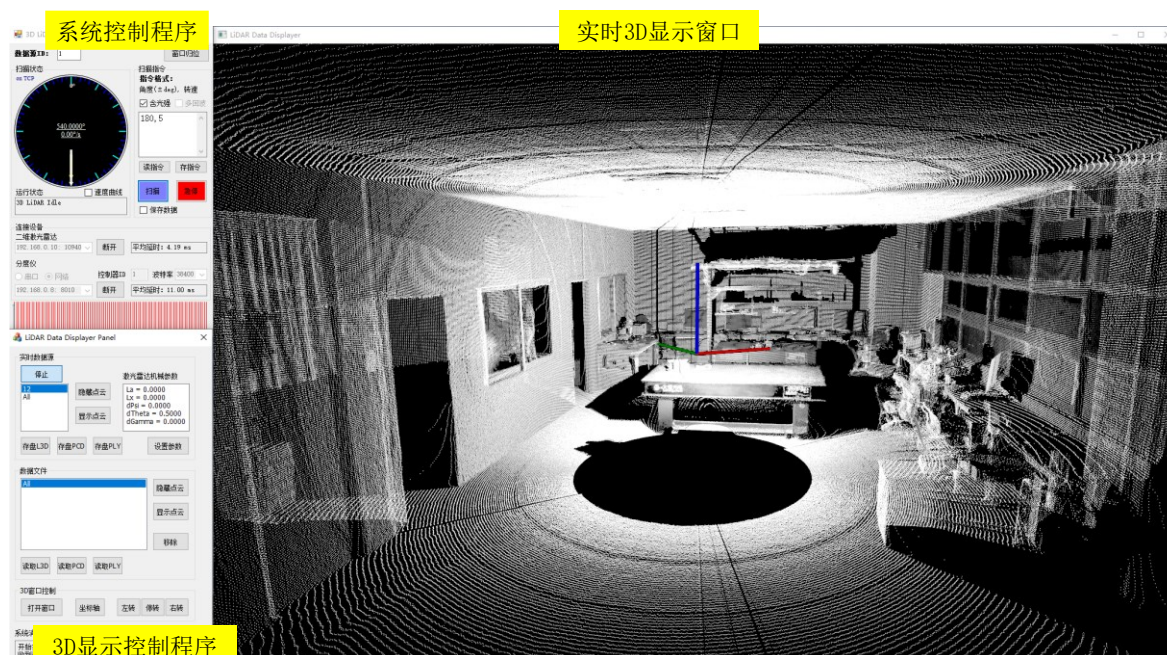


图 2 三维扫描系统控制程序和 3D 显示程序

1.2. 系统坐标系

三维扫描系统坐标系定义如图 3。系统水平放置，首先定义二维激光雷达坐标系 $x_L y_L z_L$ ：旋转轴为 y_L 轴（水平方向）；旋转轴与扫描平面 $x_L O_L z_L$ 的交点为原点 O_L ；二维激光雷达的正向（ 0° ）为 z_L 轴。

然后定义系统坐标系：

- （1）分度仪的旋转轴为 z 轴，向上为正向。
- （2）过二维激光雷达中心 O_L 与 z 轴垂直的平面为 xOy 平面，与 z 轴的交点为原点 O 。理想情况下 O_L 与 O 重合。
- （3）分度仪 0° 方向为 y 轴。 y_L 与 y 的夹角 ϕ 是分度仪的转角。
- （4）坐标轴的正向由右手螺旋定则决定。
- （5）分度仪转角 ϕ 和激光扫描角 θ 的正向均为顺时针方向，与右手螺旋定则相反。

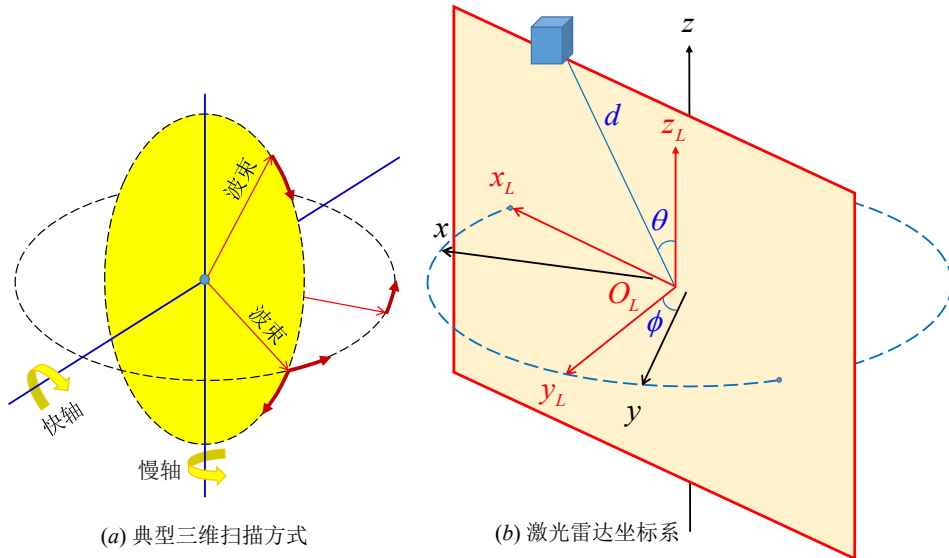


图 3 三维扫描系统坐标系

系统装配偏差用 5 个系统参数 $L\alpha$ 、 Lx 、 $\Delta\psi$ 、 $\Delta\theta$ 、 $\Delta\gamma$ 定义。其中 $L\alpha$ 、 Lx 是 O_L 与 O 的偏移量， $\Delta\psi$ 是 $x_L O_L y_L$ 平面与 xOy 平面的夹角， $\Delta\theta$ 是 z_L 与 z 的夹角， $\Delta\gamma$ 是激光射线与扫描平面 $x_L O_L z_L$ 的夹角。这些参数通过标定得到。

原始数据转换为直角坐标系需要两次坐标变换和两次旋转。旋转矩阵为

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix}, R_y(\alpha) = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix}. \quad (1)$$

首先变换到二维激光雷达坐标系，包括一次坐标变换和一次旋转。初始坐标为 $\mathbf{x}_B = [0, 0, d]^T$ ，变换公式为 $\mathbf{x}_L = R_y(\theta)[R_x(\Delta\gamma)\mathbf{x}_B + \Delta\mathbf{p}]$ 。忽略 $\Delta\mathbf{p}$ 得到

$$\begin{pmatrix} x_L \\ y_L \\ z_L \end{pmatrix} = R_y(\theta)R_x(\Delta\gamma)\begin{pmatrix} 0 \\ 0 \\ d \end{pmatrix} = d\begin{pmatrix} \cos\Delta\gamma\sin\theta \\ -\sin\Delta\gamma \\ \cos\Delta\gamma\cos\theta \end{pmatrix}. \quad (2)$$

表 2 网络设备设置

设备	参数设置
路由器 WiFi 热点	静态 IP: 192.168.0.1。SSID: LiDAR Server xxxx, 密码: 12345678。
迷你网桥	在“LAN 口设置”, 静态 IP: 192.168.0.2xx, 子网掩码: 255.255.255.0, DHCP: 启用。 在“WiFi 搜索”, 选择 WiFi 热点 LiDAR Server xxxx, 输入密码 12345678; 高级设置-热点认证匹配模式: SSID 和密码认证模式(无 MAC 认证)。
二维激光雷达	IP: 192.168.0.10: 10940
EP-E300 串口服务器 (485 转网络)	IP: 192.168.0.8: 8010, 或 USB 串口

1.4. 软件系统结构

三维扫描系统软件包括三个程序，软件结构如图 5：

- (1) **Lidar3DScanner** 是设备控制和数据获取程序。
- (2) **LidarDisplay** 是数据实时显示程序（需要安装 PCL1.13）。
- (3) **LidarDisplaySim** 用于仿真数据获取过程。

后面详细说明。

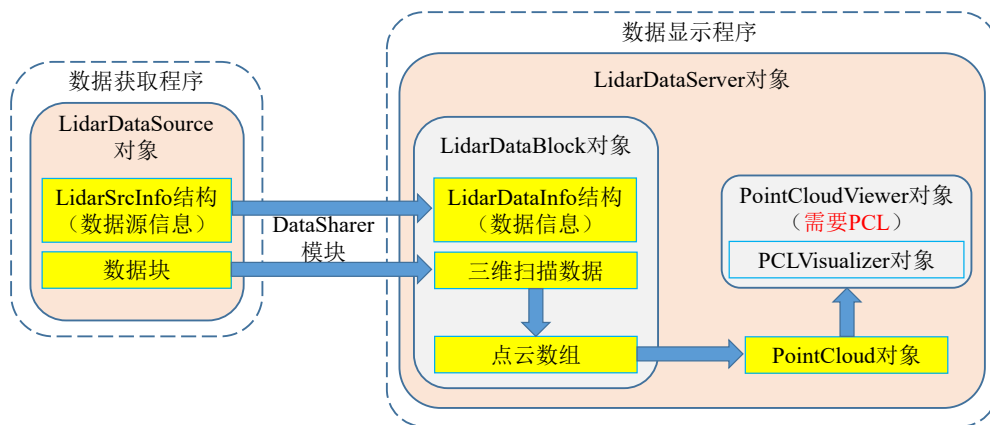


图 5 三维扫描系统的软件结构

2. 控制程序Lidar3DScanner使用说明

2.1. 扫描操作

三维扫描系统控制程序 **Lidar3DScanner** 的界面见图 6，分为四个部分：左上“扫描状态”实时显示设备的运行状态；右上“扫描指令”用于扫描操作；下方“连接设备”用于连接二维激光雷达和分度仪；最下一排是报文监控，每条竖线表示一帧报文，如果间隔不均匀说明有丢帧，通常是因为计算机运行太慢。

操作步骤分为两步。首先按照物理连接方式选择接口，分别连接二维激光雷达和分度仪。然后输入扫描指令序列，点击“扫描”顺序执行扫描指令。具体说明如下：

- 扫描指令序列由多行组成，每行为一条指令，由目标角(°)和转速(°/s)组成，用逗号分隔。目标角以“@”为前缀时是绝对角度，否则是相对角度；转速总为正值，旋转方向取决于旋转角度：>0 是顺时针旋转，否则逆时针旋转。例如“@-180, 10”是以 10° /s 转到-180°，“-180, 10”是以 10° /s 逆时针转 180°。
- 如果选中“重复指令”选项，则扫描指令序列执行完毕后再次执行，否则自动停机。
- 启动扫描时读取配置文件 Lidar3DScanner.ini。配置文件每行定义一个参数值，格式为“参数名=参数值”，参数名及其含义见表 3。
- 扫描过程中，一方面定时共享采集的数据，在应用程序 LidarDisplay 中实时显示；另一方面，如果选中“保存数据”，则三维扫描数据实时保存为 L3D 格式。
- 扫描过程中可以点击“急停”强行结束扫描。



图 6 Lidar3DScanner 的界面

表 3 Lidar3DScanner.ini 的参数

参数名	含义
La 或 LiDAR_p1	三维扫描系统参数值 (La)
Lx 或 LiDAR_p2	三维扫描系统参数值 (Lx)
dPsi 或 LiDAR_p3	三维扫描系统参数值 ($\Delta\psi$)
dTheta 或 LiDAR_p4	三维扫描系统参数值 ($\Delta\theta$)
dGamma 或 LiDAR_p5	三维扫描系统参数值 ($\Delta\gamma$)
LiDAR_p6 ~ LiDAR_p8	其它参数值 (不使用)
period	数据发送给显示程序的周期 (ms)
pack_rows	每次发送显示程序的最大帧数 (行数)

2.2. L3D 数据文件格式

三维扫描数据是三维矩阵, 即行、列、内容, 每个数据点的基本内容是物体距离 d , 其它内容如反射光强 i 、颜色 c 等取决于设备类型。

三维扫描数据可以保存为 L3D 文件。扫描时如果选中“保存数据”, 则数据实时保存为 L3D 文件。如果不实时保存, 在数据传输给显示程序 LidarDisplay 后也可以保存为 L3D 文件。

L3D 文件是文本数据表, 大小为 $(1+M) \times (3+NP)$, 格式见表 4。

- 第一行是标题行, 内容为行数 M 、列数 N 、内容数 P 、列角度 θ (M 个)、系统参数 (8 个), 剩余部分为 0。其中系统参数包括 5 个系统装配偏差 La 、 Lx 、 $\Delta\psi$ 、 $\Delta\theta$ 、 $\Delta\gamma$ 以及 d_{max} (最大探测距离)、 dT_s (帧间时间间隔)、0。
- 第一列是数据行号 (数据帧序号)。
- 第二列是数据帧时戳 (ms), 间隔应为 dT_s 。
- 第三列是行角度 (ϕ)。
- 右下角 $M \times (NP)$ 矩阵是三维扫描数据, 每个数据点有 P 个数据。

表 4 三维扫描数据文件 (L3D) 格式 (单位: 时间为 ms, 距离为 m, 角度为 $^\circ$)

	列 1	列 2	列 3	列 4 ~ 列 3+P×N		
标题行	P,	N,	M,	angleCol[0], ..., angleCol[N-1], $La, Lx, \Delta\psi, \Delta\theta, \Delta\gamma, d_{max}, dT_s, 0, \dots$		
数据行 1	1,	tsRow[0],	angleRow[0],	rawdata[0][0],	...,	rawdata[0][N-1]
...	...,	...,	...,	...,	...,	...
数据行 M	M,	tsRow[M-1],	angleRow[M-1],	rawdata[M-1][0],	...,	rawdata[M-1][N-1]

【注】 M 、 N 、 P 分别为 rowTotal (行数)、colTotal (列数)、datumSize (内容数)。如果 $P > 1$, 则 rawdata 的每个元素的长度为 P , 标题行的末尾用 0 补足。

L3D 文件可以在 MATLAB 中用 importdata() 函数读取。也可以用如下 read_L3D() 函数读取和分析, 用 dist2points() 函数转换为点云, 用内置函数 scatter3() 显示。例如:

```

% Show data in 'Data'
%% read data
fn = 'Scan3D20220620160441.L3D';
[rData, angleV, angleH, timestamp, datumSize, La, Lx, Dpsi, Dtheta, Dgamma] = read_L3D(fn);

%% show point cloud
% La = 0; Lx = 0; Dpsi=0; Dtheta=0; Dgamma=0; % overwrite parameters
ps = dist2points(rData, angleV, angleH, La, Lx, Dpsi, Dtheta, Dgamma);
figure(10); scatter3(ps(:,1),ps(:,2),ps(:,3),1,'');
% az = 10; el = 20; view(az,el)
% xlim([-2,0.3]); ylim([-0.4 2]); zlim([-1,3])
xlabel('x'); ylabel('y'); zlabel('z');

function [rData, angleV, angleH, timestamp, datumSize, La, Lx, Dpsi, Dtheta, Dgamma] = read_L3D(fn)
% Function: load L3D file.
% Input:
%     fn - data file name (L3D format).
% Output:
%     rData - range data (H*V).
%     angleV - vertical angles theta (V*1).
%     angleH - horizontal angles phi (H*1).
%     timestamp - timestamp of rows (H*1)
%     datumSize - size of one point
%     La, Lx, Dpsi, Dtheta, Dgamma - 3D LiDAR parameters
%
% Written by LIN, Jingyu (linjy02@hotmail.com), 20200127
%

rawdata = importdata(fn);
cols = size(rawdata, 2); % total columns

% resolve title row
datumSize = rawdata(1,1);
V = rawdata(1,2);
H = rawdata(1,3);
angleV = rawdata(1, 3+(1:V));
La = 0; Lx = 0; Dpsi=0; Dtheta=0; Dgamma=0;
if cols >= 3+V+5
    La = rawdata(1,3+V+1);
    Lx = rawdata(1,3+V+2);
    Dpsi = rawdata(1,3+V+3);
    Dtheta = rawdata(1,3+V+4);
    Dgamma = rawdata(1,3+V+5);
end
end

```



```

% resolve data rows
timestamp = rawdata(2:end, 2);
angleH = rawdata(2:end, 3);
rData = rawdata(2:end, 3+(1:datumSize*V));

if length(angleH) ~= H
    disp 'rows error'
end

function pntcloud = dist2points(rData, angleV, angleH, La, Lx, Dpsi, Dtheta, Dgamma)
% Function: convert distance array to point sets.
% Input:
%     rData - LiDAR data (H*V*szData).
%     angleV - vertical angles theta (V*1).
%     angleH - horizontal angles phi (H*1).
%     La, Lx, Dpsi, Dtheta, Dgamma - 3D LiDAR parameters
% Output:
%     pntcloud - x, y, z and intensity of points ((H*V) * 4)
%
% Written by LIN, Jingyu (linjy02@hotmail.com), 20210428
%

% preprocessing
angleV = angleV(:)' + Dtheta; % first rotation
SV = sind(angleV); % row vector
CV = cosd(angleV); % row vector
Rot = [1 0 0; 0 cosd(Dpsi) -sind(Dpsi); 0 sind(Dpsi) cosd(Dpsi)];
Cgamma = cosd(Dgamma); Sgamma = sind(Dgamma);

% prepare for transform
H = length(angleH);
V = length(angleV);
datumSize = size(rData,2)/V;
x = zeros(H,V);
y = zeros(H,V);
z = zeros(H,V);
SH = sind(angleH);
CH = cosd(angleH);
for i = 1:H % for each scanning plane
    % polar coordinates to Cartesian coordinates
    dist = rData(i,1:datumSize:end);
    rc = dist*Cgamma;
    yL_hat = -dist*Sgamma;
    xL_hat = rc.*SV;
    zL_hat = rc.*CV;

```

```

% scanning plane to LIDAR coordinates
x_tilde = Rot*[xL_hat;yL_hat;zL_hat]; % rotation from deviation angle
x_tilde(1,:) = x_tilde(1,:) + Lx; % translation
x_tilde(2,:) = x_tilde(2,:) + La; % translation
x(i,:) = x_tilde(1,:)*CH(i)+ x_tilde(2,:)*SH(i); % Zrot(-phi)
y(i,:) = -x_tilde(1,:)*SH(i)+ x_tilde(2,:)*CH(i);
z(i,:) = x_tilde(3,:);
end
x=x';y=y';z=z';
if datumSize > 1
    c = rData(:,2:datumSize:end);
else
    c = ones(H,V);
end
pntcloud = [x(:), y(:), z(:), c(:)];

```

3. 3D显示程序LidarDisplay

LidarDisplay 是三维扫描数据接收和 3D 显示终端，可扩展数据处理功能。程序界面见图 7，分为操作面板（左侧）和 3D 窗口（右侧）两部分。操作面板有三部分，说明如下。



图 7 LidarDisplay 界面

(1) 3D 窗口控制

用程序界面的按钮可以切换显示坐标轴和旋转视角。此外，3D 窗口还可以用鼠标键盘控制，见表 5。

(2) 实时数据源监控

点击“接收数据”按钮后 LidarDisplay 开始实时接收多个数据源的数据，并在 3D 窗口显示，同时按钮标签变为“停止”。再次点击该按钮停止接收数据。在列表框可以查看指定数据源的参数和点云。停止接收数据时，可以修改数据源的系统参数：首先在文本框中修改参数，然后点击“设置参数”。用“隐藏点云”和“显示点云”切换查看点云。实时数据可以保存为不同格式的文件。

(3) 点云文件访问

选择不同格式的点云文件，载入后在列表框中选择文件，用“隐藏点云”和“显示点云”查看点云。

表 5 3D 窗口的鼠标键盘操作说明

鼠标键盘操作	效果	相机参数
鼠标按下左键移动	坐标轴旋转。	pos, view
鼠标滚轮，或鼠标右键加移动	视点沿 Z 轴平移。	pos[2], clip
Shift+鼠标按下左键移动	平移焦点。	focal
Shift+鼠标左键	选点。	
x	切换鼠标左键的橡皮筋（rubber band）选择模式。	
+, -	增减点的大小。	
g	切换窗口尺度显示。	
u	切换颜色条显示。	
o	切换透视（缺省）和投影模式。	
p, s, w	切换 point-based、wireframe-based、surface-based 显示。	
j	截屏保存为 screenshot-xxx.png 和 screenshot-xxx.cam。	
r	复位相机。	
c	显示相机参数（在命令行窗口）。	
h	显示帮助信息（在命令行窗口）。	
q, e	关闭 3D 窗口。	
上下左右和空格（新增）	视点沿 X、Y 轴平移，空格恢复 focal 为原点。	pos, focal
PgUp 和 PgDn（新增）	伸缩焦距。PgUp 增大焦距/前移焦点、减小视角，放大点云。	fovy, focal

4. 扫描仿真程序LidarDisplaySim

LidarDisplaySim 读取 L3D 文件及相应的配置（LidarDisplaySim.ini）并分段传给 LidarDisplay 显示，用于模拟三维扫描系统扫描获取数据。

程序界面见图 8，操作分为两步：首先点击“读取数据”，选择三维扫描数据文件并读取，同时从参数文件 LidarDisplaySim.ini 读取参数值。然后点击“扫描仿真”，分段给 LidarDisplay 发送数据，全部数据发送后停止，或者发送过程中再次点击“扫描仿真”

停止发送。参数文件 LidarDisplaySim.ini 定义系统参数和传输参数，每行一个参数，格式为“参数名=参数值”，其中参数名见表 6。

表 6 LidarDisplaySim.ini 的参数

参数名	含义
La、Lx、dPsi、dTheta、dGamma	三维扫描系统机械参数
LiDAR_p6、LiDAR_p7、LiDAR_p8	三维扫描系统附加参数
period	数据发送周期（ms）
pack_rows	每次发送数据的最大帧数（行数）

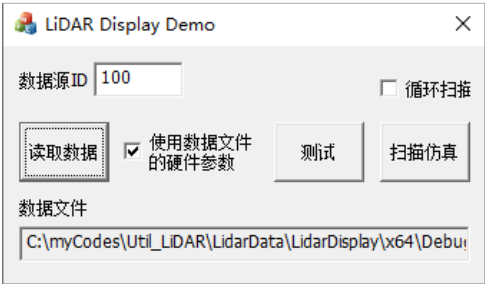


图 8 LidarDisplaySim 界面