

基于 Arduino 的 3D 超声波扫描装置

课题研究记录

穆纯亮¹

(1. 复旦大学物理学系, 上海 200433)

1. 课题设计

本课题在设计过程中主要经历了两个阶段。阶段 1 未能达到理想效果后, 改换了思路, 转向阶段 2。

1.1. 阶段 1 设计: 手持式+MPU6050 探测仪器位置与姿态 (未能达到理想效果)

我最初的想法是, 构造一个手持式的 3D 超声波扫描装置。将 MPU6050 运动传感器和 HC-SR04 超声波传感器结合在一起。利用 Arduino 单片机进行控制, HC-SR04 负责测距, MPU6050 负责侦测仪器的位置, 从而达到 3D 扫描整个空间的效果。随后, 再通过数据线 (构想中还考虑过蓝牙或 Wi-Fi) 把信息传回电脑, 通过 Python 脚本写成的客户端, 实时显示扫描效果图。装置图和客户端如图所示:

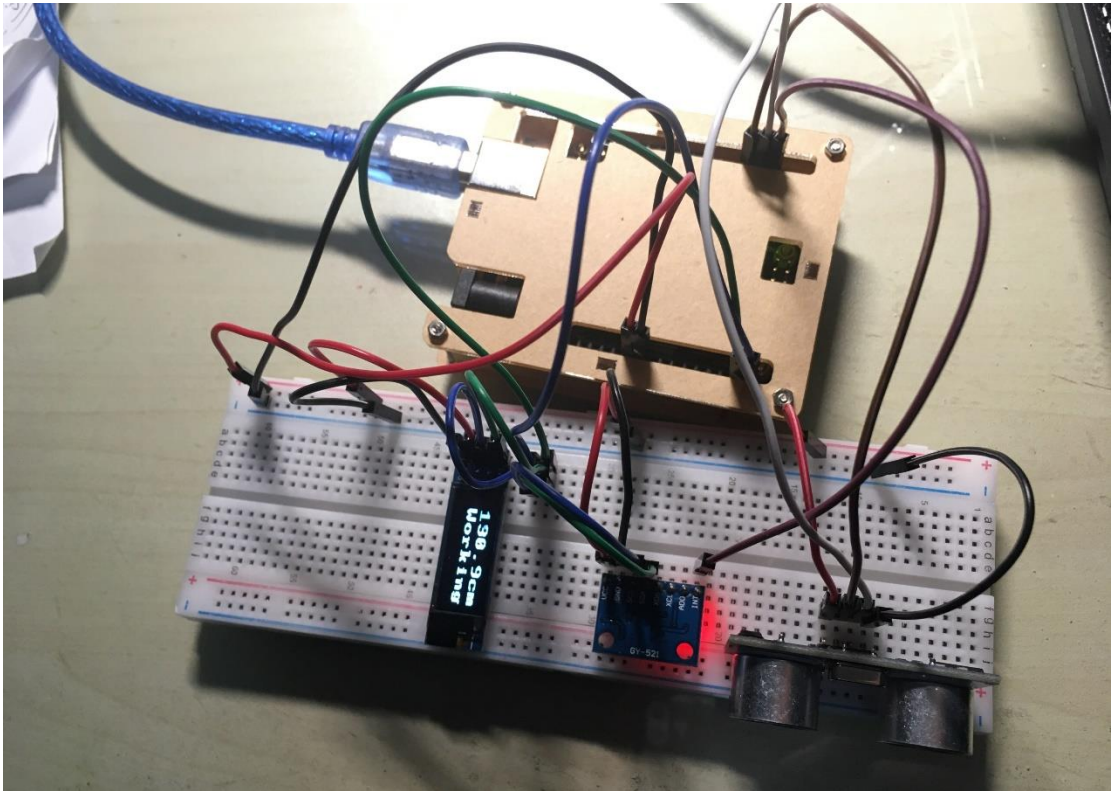


图 1 阶段 1 设计-装置 效果展示。面包板上从左到右依次为: OLED 显示屏 (用于提示仪器运行状态), MPU6050 运动传感器, 和 HC-SR04 超声波探测器。

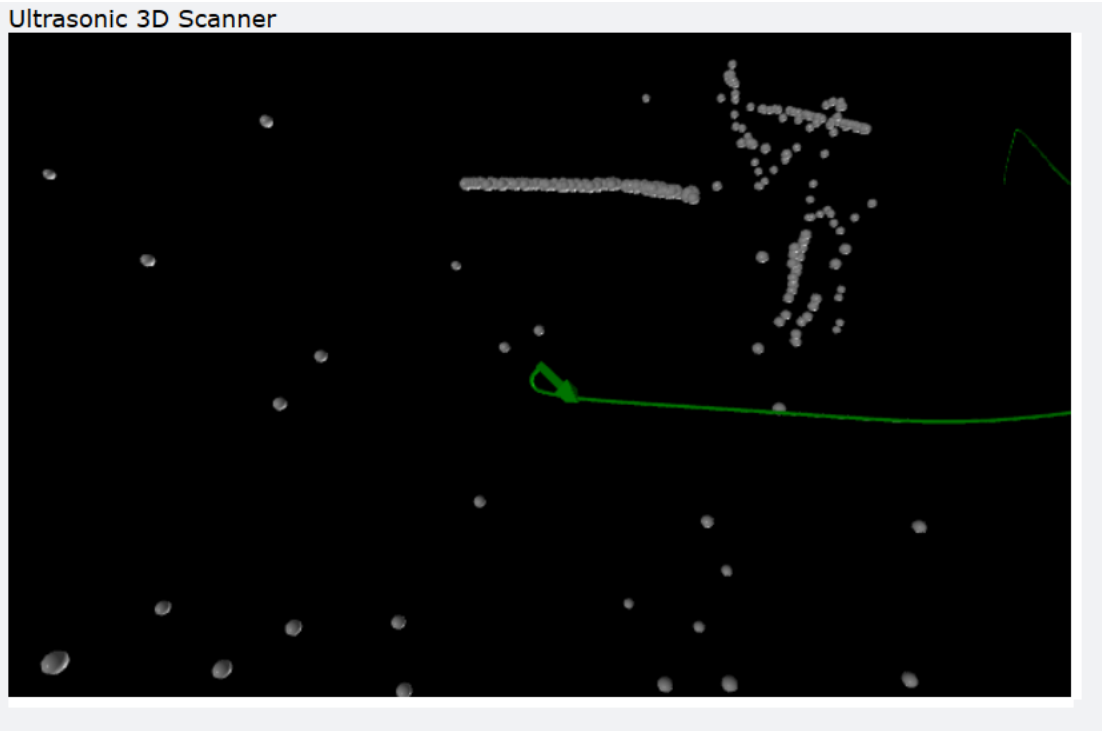


图 2 阶段 1 设计-客户端 效果展示。绿色箭头为当前仪器的朝向。白色的球为测得的数据点。绿色曲线为仪器运行轨迹。整个过程中仪器被放置于桌面上，并没有被明显移动。注：图中白色数据点绘制时并没有加入此仪器位置的修正。因为，如图可见，它的误差漂移实在太大了。

这里的问题出在 MPU6050 运动传感器上。它实际上测量的是 3 维的加速度和角加速度。虽然理论上可以通过两次积分解出位置和朝向信息(我开始课题前也确实有查到资料宣称做到了这一点 (Deng, Qiu, Zhong, & Wan, 2015)，但实际操作下来发现，即使采用了 RK-4 积分法减小误差，误差也会在两次积分下快速累积到不能容忍的地步。结果就是图 2 中明明没有移动仪器、却出现了的仪器运行轨迹。因此这一设计不够合理。最简单的修改办法，就是抛弃手持式的设定和 MPU6050 探测器。

1.2. 阶段 2 设计：固定式+舵机调整仪器朝向

接受了俞老师的建议后，我换了一种思路：固定住仪器的位置，通过 Arduino 控制 2 台 MG996R 舵机旋转两自由度云台来调整仪器朝向，然后 HC-SR04 探测器执行数次扫描取平均值，获取这个朝向上的位置信息，达到三维扫描的效果。虽然这限制了应用效果（不再能随身携带、边前进边扫描），但胜在简单、可靠，而且便于提升精确度。改换设计后的装置如图 3：

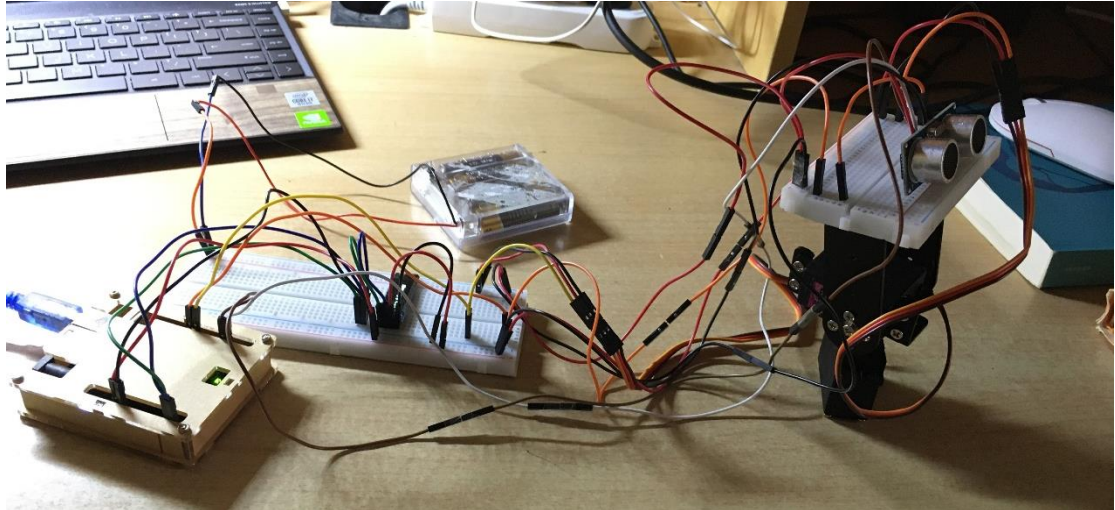


图 3 阶段 2 设计-装置 效果展示。图中左下连接有蓝色 USB 线接头的是 Arduino UNO 单片机；它的右上方是面包板，连有 OLED 显示器和另一个 4 节 5 号电池的额外电源（供舵机使用）。右边是两自由度云台，由 2 台 MG996R 舵机操控；云台上粘贴有一块小面包板，其上为 HC-SR04 超声波测距器。这使得超声波测距器能够扫过整个三维空间。

实验仪器包括电脑、Arduino UNO 单片机、HC-SR04 超声波传感器、32*8 OLED、2 个 MG996R 舵机、4 节 5 号电池、2 块面包板、跳线若干、以及一根 USB-A 口转 B 口的线。这里有一个小坑：MG996R 的消耗电流特别大（查资料知可以达到 1~2A），Arduino UNO 板自带的 5V 输出端是不够的，需要额外的电源——这里我使用的是 4 节 5 号电池。

以上装置图还有一点缺憾，那就是两自由度云台没有支架。由于时间和资金关系，这一块并没有被我补上，因此实际操作过程中需要有人手持云台来保持稳定。这会带来一定误差，详见误差分析。

2. 实验过程、结果与分析

2.1. 实验过程与结果

由于缺乏支架，手持两自由度云台进行扫描。扫描地点为宿舍，结果如图所示：

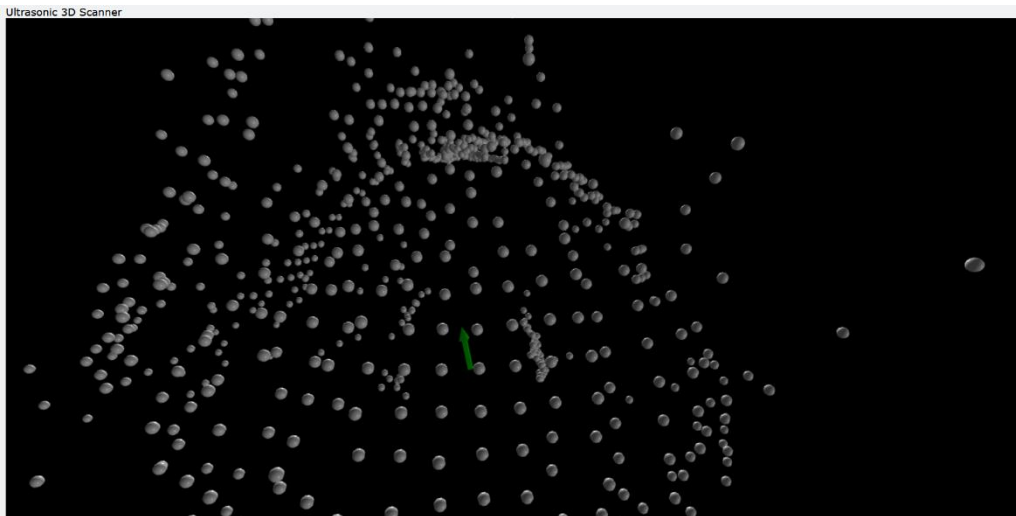


图 4 扫描结果

从图中可以看出处于仪器右侧的测试者、左侧的橱柜墙，以及天花板等结构。

2.2. 误差分析

主要能够造成误差的因素有：

- a. 手持仪器时手的抖动造成的 HCSR-04 仪器位置和朝向的误差；
- b. 舵机运行时的振动导致 HCSR-04 仪器位置和朝向的误差；
- c. 舵机运行到的角度缺乏定标，造成的 HC-SR04 朝向的不准确；
- d. 在云台旋转的过程中，云台旋转的轴并不通过 HC-SR04 传感器，而是有一定的间距，但在程序处理的过程中并没有考虑到这一间距，从而产生的误差；
- e. 扫描对象的表面不够大、不够平整、或存在倾角导致 HCSR-04 距离测量的不准确；
- f. 环境温度变化，导致的空气中声速的变化，进而导致距离测量的不准确。

以上因素中，a 可以通过增加支架来解决，b 可以通过舵机运行完成后等一小段时间后再开始扫描来缓解，c,d 可以通过定标（可以通过扫描一个标准的立方体盒子来计算）来消除，f 可以通过再仪器上增加温度计模块，或在后期数据处理中手动添加温度参数来缓解。但 e 是与扫描对象的性质有关的，难以去除。

3. References

Deng, J. M., Qiu, J. S., Zhong, Z. Y., & Wan, Z. P. (2015). Three-dimensional Trajectory Tracking System Based on Compass and Gyroscope. *International Conference on Education, Management, Commerce and Society*, 545-550.

4. 附录

相关代码见附件。

运行方法：首先按图 3 组装好装置，连接 Arduino UNO 到电脑。正确设置好 Python 客户端中 Arduino UNO 板对应的端口名称后（CONST_SER_PORT_NAME 变量），启动 Python 客户端，扫描会自动开始，并且会自动将数据保存到客户端所在目录下。需要重复扫描的话，按 Arduino UNO 板上的 reset 按钮。如果 Python 客户端显示 open handshake timeout 错误，就再试几次。