



无人驾驶车介绍

无人驾驶对安全精度和各种复杂技术的集成程度要求非常高，因此到目前为止，在民用领域里，无人驾驶车辆的技术难度之高是数一数二的。

做一辆无人驾驶车肯定要比做一个手机APP要复杂很多倍。无人驾驶车到底包括哪些核心技术呢，接下来我将为大家一一讲解。



备受大家热捧的无人驾驶车并不是一辆普通的车，因为无人驾驶的车一定要是线控车。所谓线控(by-wire)，就是说这个车是能被电脑控制的。车的方向盘和轮子之间是有电机连接的，方向盘传动到电机上，然后电机去控制车轮，这样的话电脑就可以通过控制电机的方式来控制这个车轮。

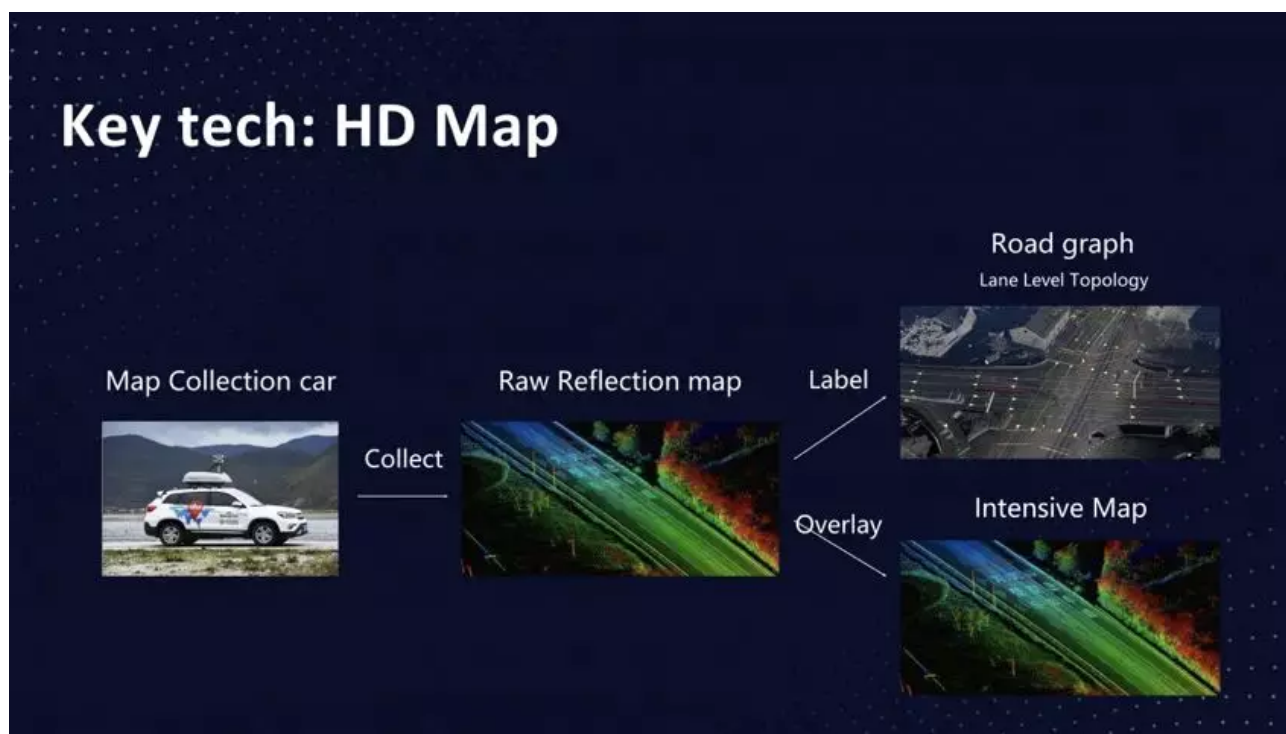
为了保证无人驾驶车的安全性，当它在路上行驶的时候，必须做到它跟云端是有连接的。无人驾驶车并不需要时时跟云端汇报接下来会如何处理，而是要告诉云端它的位置以及行驶规划。



对于无人驾驶来说，在马路上行驶一定要有高精地图，这里所说的高精地图跟我们平时开车用的导航地图有什么不同呢？

所谓的导航地图就是我们平时使用的百度地图、谷歌地图等，在导航地图上，它会告诉你怎么从A点到B点，然后有几条路，预计需要用多长时间等信息。

而高精地图相对于导航地图来说，最大的特点就是高精度，它是需要做到车道线级别的。也就是说，高精地图不光要知道你在哪条路上，还需要知道你在这个路的哪条车道线上，因为只有这样才能准确地告诉无人驾驶车应该在哪个车道行驶，接下来应该怎么拐。



高精地图对于无人车来说，具有非同寻常的价值。

第一，**高精地图能够给无人车很多预判的空间**。当无人车通过高精地图知道前方的路况和交通标识信息后，能够提前做行驶规划，保证了行车的平稳性和经济性。

第二，**高精地图能够帮助无人车减少计算量**。当无人车需要通过路口时，它需要提前感知前方信号灯的状态，这时高精地图就可以帮助它定位到信号灯所在的特定区域，从而有效降低了全范围扫描

识别的计算量。

除此以外，高精地图将道路及周围的所有静态障碍物进行收集，**减少无人车对静态障碍物的算法处理。**



无人车有了高精地图之后，还需要知道它在地图的什么位置，这里就需要使用定位技术。说到定位，很多人可能最先想到GPS。那么GPS定位的原理是什么呢？其实**GPS定位原理实际上就是一个相对定位。**

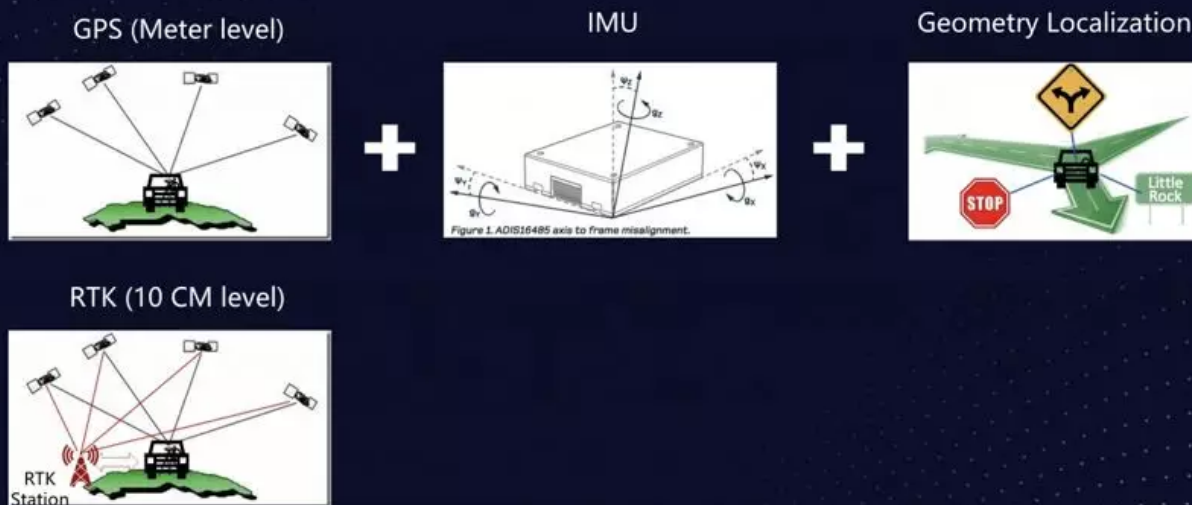
每个卫星会不停地发电磁波，当收到电磁波信号之后，根据收到信号的时间与光速算出和卫星之间的距离。有了与四个卫星的距离之后，就可以解一个方程算出所在位置。但由于电离层、反射作用等因素的干扰，**GPS定位是有误差的，它的精度只能达到米级。**对于无人车来说，米级的定位精度是不安全的，为了解决无人车定位问题，还需要其他技术。

GPS定位还有一个问题，它是跳动的。GPS定位是每时每刻根据当前的时间去算的，容易出现计算结果不准的情况。为了抹平GPS的跳变，需要用到IMU（惯性导航），一般来说GPS和IMU是一块用的，GPS不停的去给IMU一个方向去校准，然后IMU再给GPS一个方向。

除此之外，**无人车在接收不到GPS信号的情况下，需要用到另一个技术——几何定位。**

几何定位的原理和GPS原理差不多，就是在道路上选几个feature，根据这些feature计算无人车所在的位置。几何定位的精度很高，可以精确地算出无人车所在的位置。所以，目前比较流行的定位技术就是GPS、IMU和几何定位等一系列技术的融合。

Key tech: Localization



为了提高GPS的定位精度，大家又在GPS的基础上发明了RTK技术。手机GPS定位中，手机就是一个移动站，它会经常变动位置。这里所说的RTK，是一个静止站，它同样也收到卫星的信号。无人车与RTK相隔不太远的情况下，对二者之间的干扰信号用差分抹平，就可以认为无人车和RTK收到的信号是一样的。

RTK的应用能让GPS的定位精度达到10厘米，但是RTK技术的应用有个限制，它要求基站与车的距离要在16公里以内。



我们人类天生具备感知外界的能力，是因为我们具备各种“传感器”。

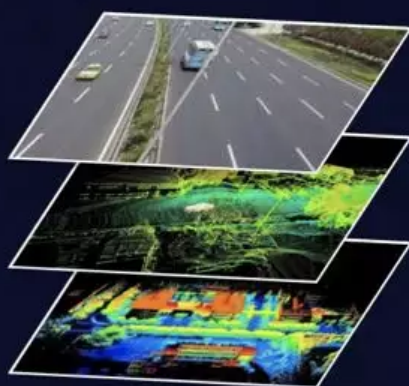
无人车想在道路上安全行驶，也必须具备属于它自己的各类传感器去感知外界环境。目前，**无人车主要的传感器有摄像头（Camera）、雷达（Radar）和激光雷达（Lidar）。**

Key tech: Sensing and Perception

Sensor HW



Sensor Fusion



Perception

Segmentation



Classification



Tracking



摄像头(Camera)

对于无人车来说，观察信号灯以及一些交通标识都需要对**颜色识别**。

摄像头恰能解决这个问题，它的**主要优点是能够辨别颜色，但是它没有对距离的判断的能力**。因此，实现无人车对外界的感知还需要其他传感器的配合。

雷达(Radar)

雷达是无人车比较重要的传感器之一，它是利用**电磁波**探测目标的电子设备。

雷达发射电磁波对目标进行照射并接收其回波，由此获得**目标至电磁波发射点的距离、距离变化率（径向速度）、方位、高度等信息**。雷达的应用场景是全天候的，由于电磁波可以绕过一些东西，所以它的准确性并不太高。

雷达还有一个特点，它对于速度判断非常准，而对于静态物体的误报比较多。

激光雷达(Lidar)

激光雷达是目前无人车发展的一个比较核心的技术，它是以**发射激光束**探测目标的位置、速度等特征量的雷达系统。

激光雷达工作原理是向目标发射探测信号，然后将接收到的从目标反射回来的信号与发射信号进行比较，作适当处理后，就可获得目标的有关信息。

激光雷达的最大优点就是对距离的判断非常精准，但它也有个很大的缺点，就是对环境的要求非常高，比如在雾霾天气里激光雷达的精准度就会降低很多。



激光雷达是一个非常新的技术，在发展无人车之前，它并没有大规模的商业应用场景，因此它的价格非常昂贵。虽然目前激光雷达价格很贵，但它的种类还是挺多的。

大家可能见过无人车上有个不停旋转的Lidar，这种是机械雷达。由于它需要不停旋转，它本身也又大又重，机械损耗较大，容易出现問題。基于机械雷达，大家又发明了一些新的激光雷达，比如MEMS lidar、flash lidar等。

无人车收到各种传感器获取的信息之后，还需要做**传感器融合**。所谓传感器融合，就是要把每个传感器看到的東西全部都叠加在一起。当所有传感器看到的信息综合在一起，无人车才能够更加全面具体地感知外界环境。

轨迹规划

当车辆在路上行驶时，通过传感器获取了周围的信息，接下来就需要知道如何反应。

对于周围的车辆或其他障碍物，无人车是需要避让还是需要停下来，或者需要超越等等，这些都属于约束条件。无人车需要在所有的约束条件中，规划出一条可以走的路线。

Key tech: Trajectory planning

Smoothness of Distance(S), speed ($\Delta s/\Delta t$), acceleration ($\Delta s/\Delta t^2$), ($\Delta s/\Delta t^3$), ($\Delta s/\Delta t^4$)...

Assume the following speed profile, with parameter (a, b, c, d, e, f) :

$$\begin{aligned} v(t) &= a + b \cdot t + c \cdot t^2 + d \cdot t^3 + e \cdot t^4 + f \cdot t^5 \\ s(t) &= a \cdot t + b \cdot t^2/2 + c \cdot t^3/3 + d \cdot t^4/4 + e \cdot t^5/5 \\ a(t) &= b + 2 \cdot c \cdot t + 3 \cdot d \cdot t^2 + 4 \cdot e \cdot t^3 + 5 \cdot f \cdot t^4 \end{aligned}$$

with constraint,

$$\begin{aligned} v(t_0) &= v_0 \\ v(t_f) &= v_f \\ a(t_0) &= a_0 \\ a(t_f) &= a_f \end{aligned}$$

where t_f is known. With above 5 constraints, we need one more condition to get 6 (or more) parameters. Here we define the non-linear smoothing measurement:

$$\int_{t_0}^{t_f} (v'(t))^4 dt$$

which is the penalty term of smoothing splines. To be specific, we will find parameters (a, b, c, d, e, f) that satisfying:

$$\arg \min_{a, b, c, d, e, f} \left(\int_{t_0}^{t_f} (v'(t))^4 dt \right)$$

Table 1. The following matrix was defined as the constraint matrix A .

	$v(t_0)$	$v(t_f)$	$a(t_0)$	$a(t_f)$	$\int_{t_0}^{t_f} (v'(t))^4 dt$
$v(t_0)$	1	0	0	0	0
$v(t_f)$	0	1	0	0	0
$a(t_0)$	0	0	1	0	0
$a(t_f)$	0	0	0	1	0
$\int_{t_0}^{t_f} (v'(t))^4 dt$	0	0	0	0	1

above term can be expressed as:

$$\int_{t_0}^{t_f} v'^4 dt = \int_{t_0}^{t_f} v^T V V^T v dt = \int_{t_0}^{t_f} v^T V V^T dt \cdot v$$

The second equality is due to matrix integration. It is easy to verify that the middle part has quadratic form with the following:

$$V = \int_{t_0}^{t_f} V V^T dt = \begin{bmatrix} \frac{1}{5} & \frac{1}{4} & \frac{1}{3} & \frac{1}{2} & \frac{1}{1} \\ \frac{1}{4} & \frac{1}{3} & \frac{1}{2} & \frac{1}{1} & \frac{1}{1} \\ \frac{1}{3} & \frac{1}{2} & \frac{1}{1} & \frac{1}{1} & \frac{1}{1} \\ \frac{1}{2} & \frac{1}{1} & \frac{1}{1} & \frac{1}{1} & \frac{1}{1} \\ \frac{1}{1} & \frac{1}{1} & \frac{1}{1} & \frac{1}{1} & \frac{1}{1} \end{bmatrix}$$

Then, we know that the constraint spline problem was transformed to the following:

$$\arg \min_{v} v^T A v$$

with respect to:

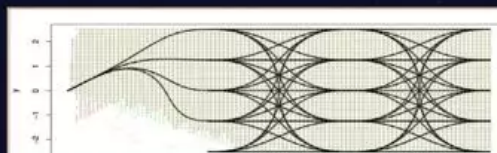
$$b = [v(t_0), v(t_f), a(t_0), a(t_f), \int_{t_0}^{t_f} (v'(t))^4 dt]^T$$

where A is $A^T A$ submatrix extracted from (A, b) to (A, b) dimension. The above quadratic optimization problem within the affine space has exact true solution:

$$\begin{bmatrix} A^T & A^T b \\ A & b \end{bmatrix} \begin{bmatrix} v \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}$$

Smoothness of

- kappa or curvature ($\Delta s/\Delta s$),
- dkappa ($\Delta s/\Delta s^2$)
- ...



无人车的轨迹规划：

一是要满足所有的约束条件。

二是要保证车辆运动的平滑。所谓平滑，是要保证车子的速度不能跳变。

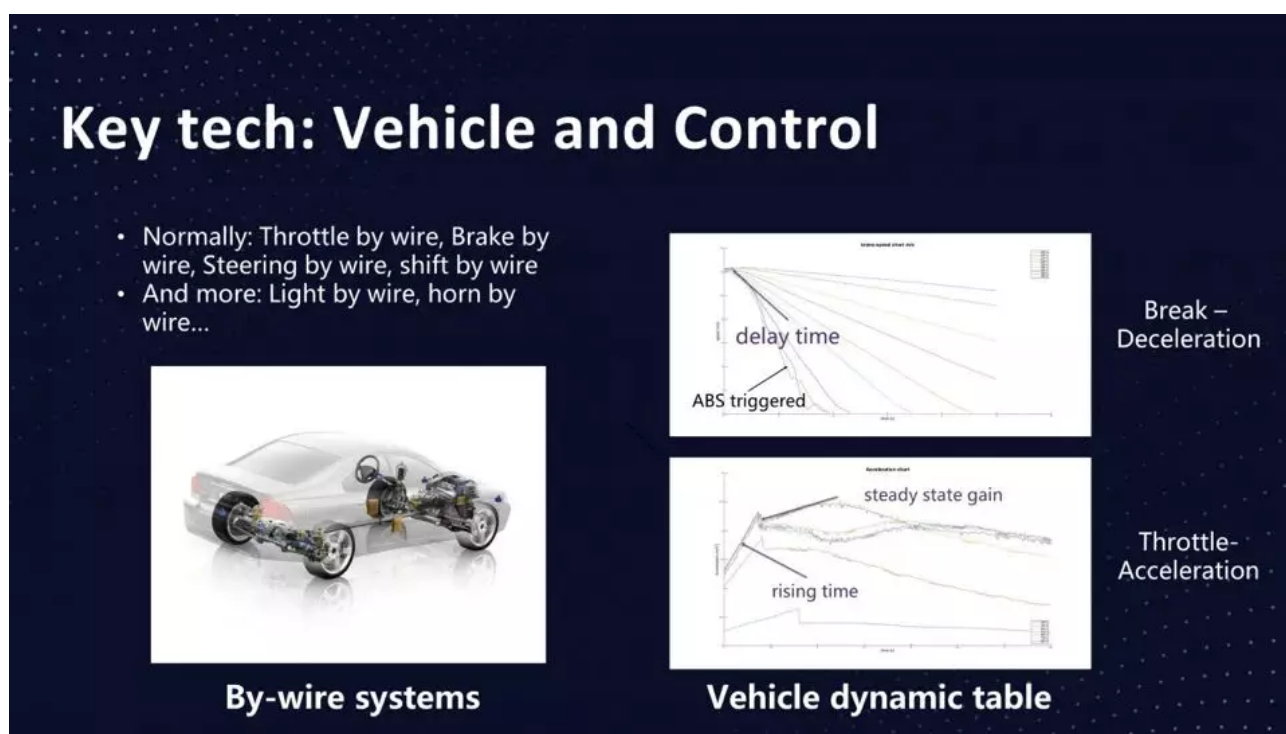
在此基础上，车子的加速度也需要是平滑的。

最终目的是，在人类可感知的范围，车子行驶是顺畅的，没有不平滑不顺的情况。

控制

前面我们已经提到，无人车是线控车，而不是一个仅有齿轮组成的集合体。

实现对无人车的控制，我们需要知道踩刹车和减速的关系、踩油门和加速的关系等，当无人车拿到一些控制学参数后，就可以实现电脑对无人车的控制。



云端

想要保证无人车是非常安全的，我们必须做到让每一辆无人车成为“有经验的司机”。所谓有经验，就是见过很多路况并且知道如何处理这些路况。

“Cloud + Vehicle”



百度在云端后台有个巨大的仿真空间，每一辆无人车可以将自己遇到的复杂路况上传到云端，因此网上就有了一个非常大的数据库。

当无人驾驶的算法有更新时，就可以在云端的仿真场景中跑一下，检验是否能够应对云端的这些路况。这个步骤，就是为了确保每个无人车都称得上是“有经验的司机”。

_____ **END** _____