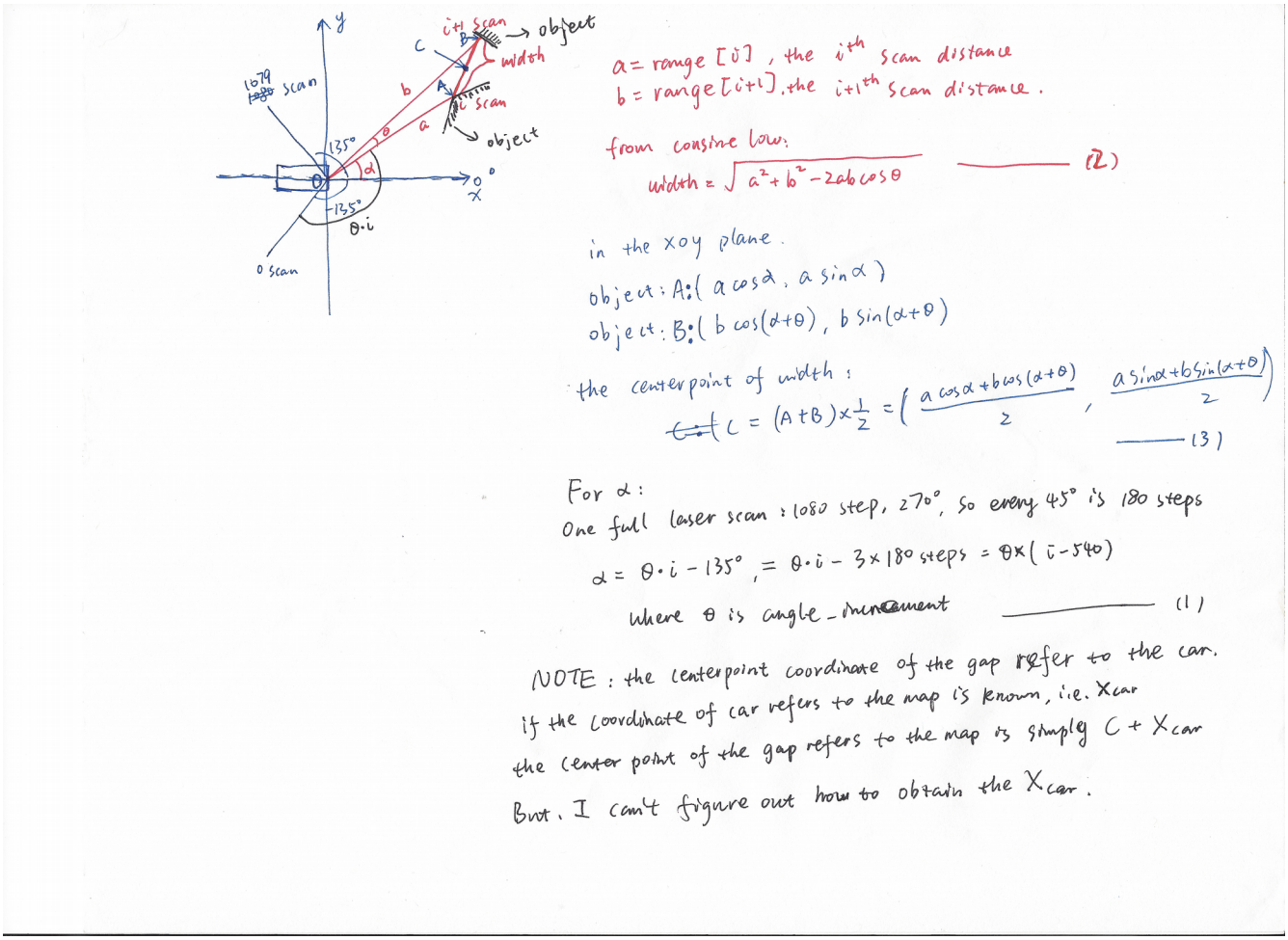
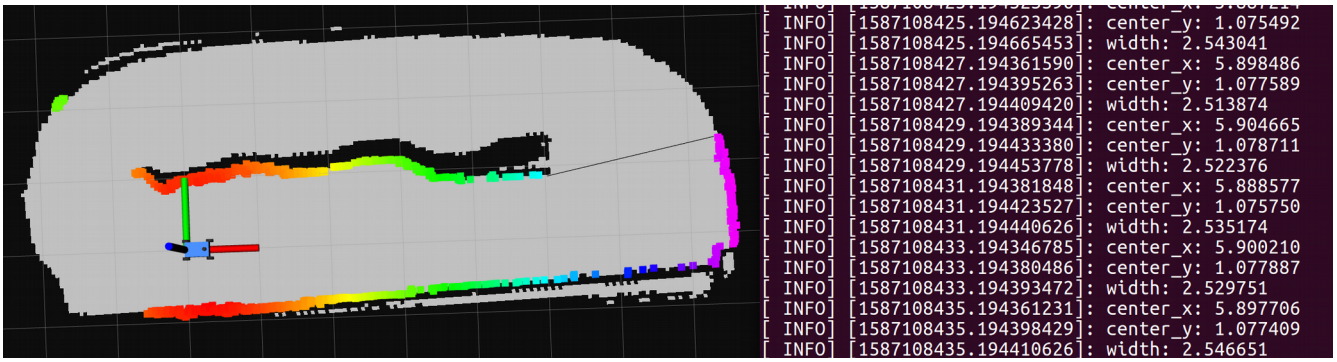
```
gaps GAP::gap_cal()
// a method in GAP class, return data type is gaps
{
	gaps result;
// the data published called result, which is type of gaps message that I created. gsps.msg contains two types of data 1) center: geometry_msgs/Vector3; 2) width: std_msgs/Float 32.

//initializationhas
	result.center.x = 0;
	result.center.y = 0;
	result.center.z = 0; //the z-axis doesn't need to change in the furture, due to 2-D movement.
	result.width.data = -1.0; //width can not be negetive, so here set it to negetive.

	float a = 0;//target scan range, one side length of a triangle
	float b = 0;//another side length of the triangle
	float alpha = 0;//target scan angle -135 deg ~ 135 deg, 0 deg towards to car front
	float width_max = -1;//max gap width

	for(int i=360; i<1080-360; i++ )
	//screen range from -45 deg ~ 45 deg, make sure the car moves forward
	{
		if( abs(range[i+1] - range[i]) > 0.5 )
		//find i+1 laser scan, which has 0.5m range difference from i
		{
			a = range[i];
			b = range[i+1];
			alpha = theta*(i – 540);//discuss later----------(1)
			width_max = a*a+b*b-2*a*b*cos(theta);//simple consine law calculation----------(2)

			if(result.width.data < width_max)
			// compare width to the previous one
			{
			 // find largest gap width and store the range
				result.width.data = sqrt(width_max);
			}
		}
	}

	//gap center point corelate to car
	result.center.x = ( a*cos(alpha) + b*cos(alpha + theta) )/2;//discuss later----------(3)
	result.center.y = ( a*sin(alpha) + b*sin(alpha + theta) )/2;//discuss later
```

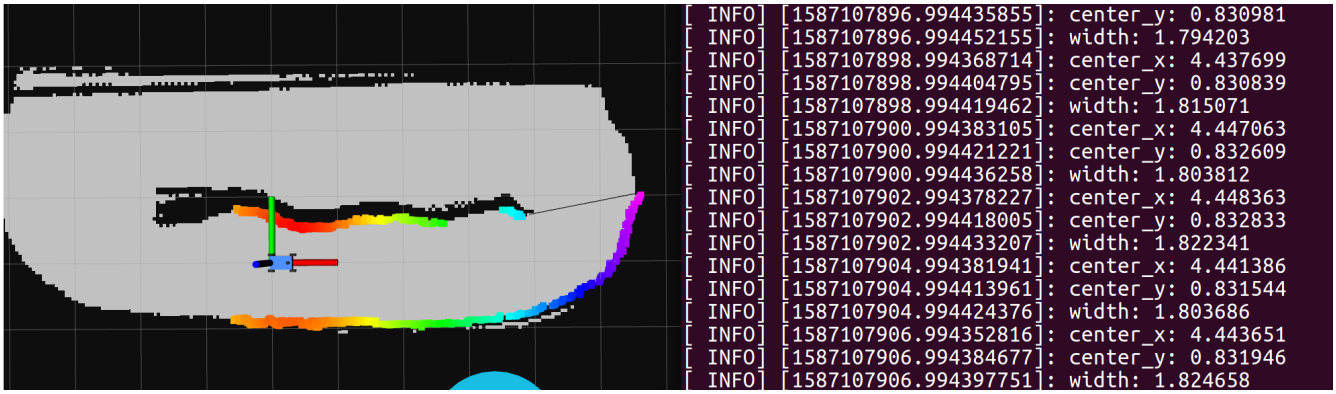


here is some results about the algorithm.

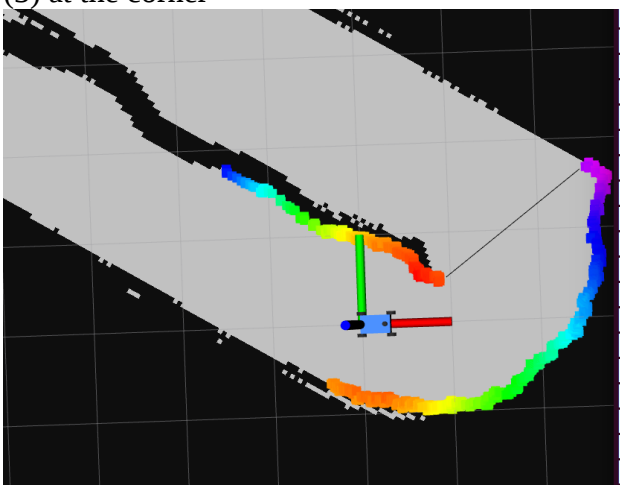
(1) away from corner



(2) approach corner



(3) at the corner



```
[ INFO] [1587108288.705948139]: width: 2.140066
[ INFO] [1587108290.705971561]: center_x: 1.472820
[ INFO] [1587108290.706029861]: center_y: 1.092511
[ INFO] [1587108290.706060510]: width: 2.141387
[ INFO] [1587108292.705884544]: center_x: 1.470512
[ INFO] [1587108292.705941697]: center_y: 1.090773
[ INFO] [1587108292.705961414]: width: 2.157025
[ INFO] [1587108294.705884127]: center_x: 1.472000
[ INFO] [1587108294.705994242]: center_y: 1.091870
[ INFO] [1587108294.706019610]: width: 2.164587
[ INFO] [1587108296.705895456]: center_x: 1.467859
[ INFO] [1587108296.705958483]: center_y: 1.088772
[ INFO] [1587108296.705982517]: width: 2.177323
[ INFO] [1587108298.705881166]: center_x: 1.471293
[ INFO] [1587108298.705919298]: center_y: 1.091358
[ INFO] [1587108298.705969534]: width: 2.153619
[ INFO] [1587108300.706132101]: center_x: 1.461618
[ INFO] [1587108300.706369536]: center_y: 1.084159
[ INFO] [1587108300.706448360]: width: 2.156353
```