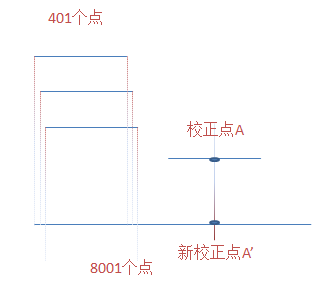
智能里程矫正算法说明

1. 从已经修正后线路中得到所有的里程矫正点, 也就是从”IndexOri”表中取得相关的数据。
2. 在未修正的线路中，分别标出与1步骤中里程矫正点相同里程（此处是原始cit里程，不是用户的更新里程）的里程点。 步骤1中的每个校正点，都含有cit中的原始里程和用户修改后的更正里程。此处，是用原始里程去查找。
3. 在步骤1中的里程矫正点左右选择个数相同的(100或200)采样点。这样，得到一批采样点（201或801），这一批点是以某个校正点为中心。
4. 同样也在步骤2中标出的里程点左右选择个数相同的(4000)采样点。这样，得到一批采样点（8001），这一批采样点是以某个校正点为中心。
5. 然后将由步骤3所构成的小波形图（201或801个点）依次在步骤4所构成的大波形图（8001个点）上通过相关性算法找出相匹配的波形段，以此确定里未修正的线路的一个程矫正点。



从大波形的开始处开始（假设大波从第0个采样点开始），运行里程相关性算法，看小波是否和0~400这段匹配，如果不匹配，再和大波的1～401这段再比较，如果还不匹配，再和大波的2～402这段再比较…….当匹配成功后，比如，找到大波的300～700这段是最匹配的（其中还通过比较一些通道(例如超高）的值来确定最终的里程矫正点）。那么300～700这段的中点，即编号为500（点的编号从0开始）的那个点，就是要找的里程标记点A’，它对应着那条修正线路的修正点A，这样A’的修正里程就是A的修正里程，A’的文件位置也是知道的。从而，为待修正线路的”IndexOri”表，就可以添加一行新数据了。

然后，再继续对已修正线路的第二个标记点执行同样的过程……直到遍历完所有的用户标记点。

上面的这个过程简单的说：就是已知一条修正线路的用户标记点，找到未修正线路的对应的标记点。

有了“IndexOri”表，再用里程修正类即可生成“IndexSta”表，从而完成“IndexSta“的构造。

上面步骤中提到的“相关性修正算法”，还提到了“其中还通过比较一些通道(例如超高）的值来确定最终的里程矫正点”，对这两点的解释如下：

（1）相关性修正算法公式为：

其中V是修正算法的结果值，N是数组的个数，是修正后的其中第i个里程值，是未修正后的其中第i个里程值。这里需要注意的是每次调用算法，需要a的个数和b的个数相同。

（2）对b进行循环得到V值最大的那个值与参数中通道（例如超高）的阈值做比较，如果大于通道的阈值，将该处的标记点和指针位置保存起来。