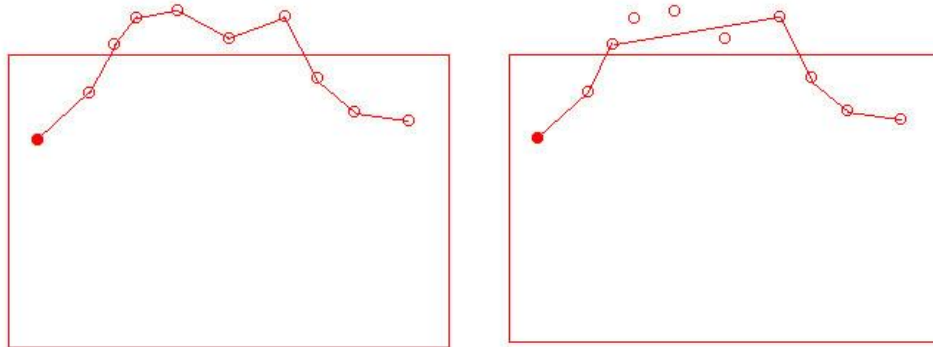


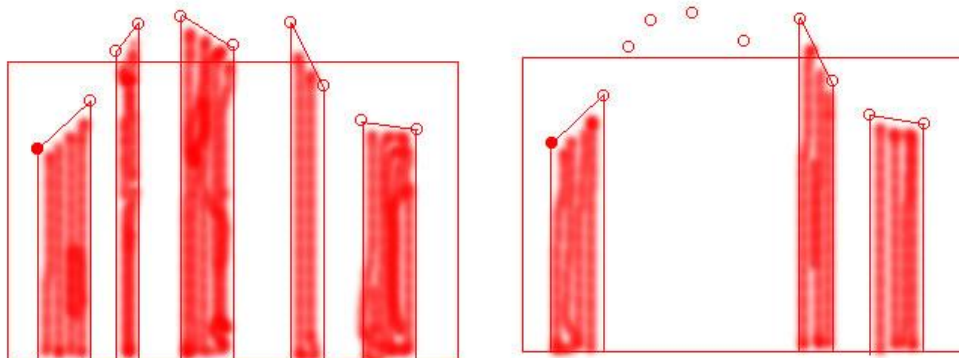
ST_Curve 升级报告(2)

2010 年 11 月，ST_Curve 控件完成了第三次算法上的升级，相比于第二次升级（参看 ST_Curve 升级报告.doc），这次升级简单许多，主要是通过跳过不必要的绘制点，达到提高绘制速度的目的。

在以下的图片中，红圈代表起始点，箭头表示曲线延伸方向（一次曲线没有画出，因为很直观），蓝圈代表拐点（拐点即为了保持曲线外观而必须要绘制的画布外点，只有二次曲线才会有拐点）；红色矩形代表画布，在画布之外的任何绘制是不可见的（在示意图中有绘制，以方便你看到优化前多余的绘制）。

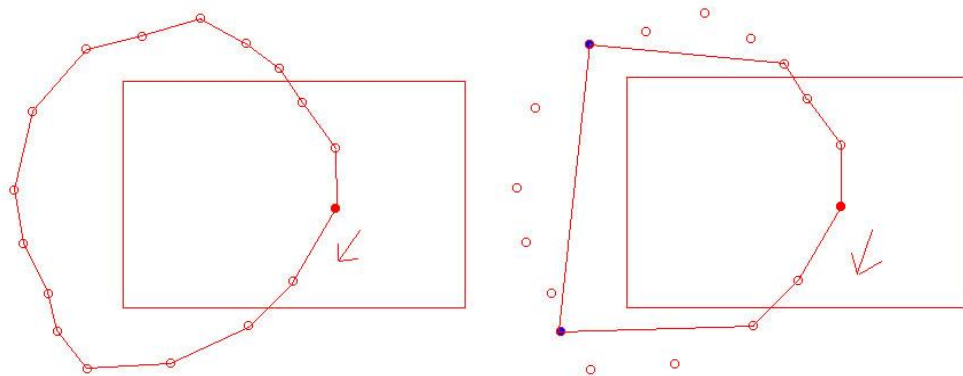


这是一维曲线优化前后对比

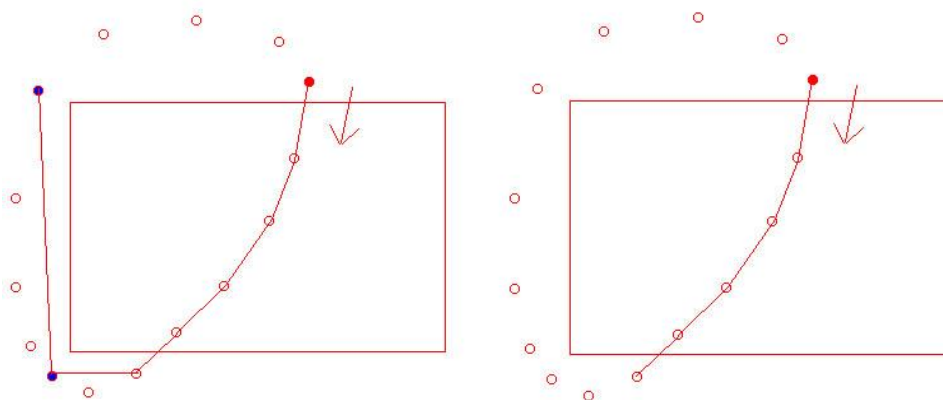


这是一维曲线（带填充）优化前后对比

从上面两组对比图可以看出，在绘制一维曲线，如果不填充，则优化前后，效果完全一样；如果有填充，则可能不一样（如上第二组图片）。



这是二维曲线优化前后对比



这是二维曲线的进一步优化效果对比

第一组是最普通的二维曲线优化效果，其中蓝色点为拐点。

二维曲线如果有填充，同样也会出现和一维曲线一样的情况，即优化前后效果可能不一样，希望二次开发者向自己的客户好好做一翻解释，这都是为了更快的绘制速度。

但在二维曲线下，有一点是特殊对待的，就是在绘制封闭（起始点与结束点重合）且填充的二维曲线时，控件会强行绘制起始点和结束点（中间的优化还是照样进行），这样可以保证在优化前后，显示效果是完全一样的。

另外，在二次曲线下，还有一种叫做“进一步优化”的步骤，如上第二组图。控件会删除最后的连续的拐点，因为绘制不具备实际意义。

请大家认真领会我上面的意思，特别是要仔细琢磨一下示意图，如果有疑问可以向我咨询，总之这次优化就一个原则：**尽量不绘制看不见的点（只看得见填充不算可见，如上面一维曲线柱状图那张示意图），但有一点例外，在绘制二次封闭填充曲线时，强行绘制起始点和结束点（这样保证填充始终是在曲线内部）。**

young wolf

2010 年 11 月 15 日