컴퓨터애니메이션 (CA) #HW1

HW1: Implement 2D curve editing interfaces for natural cubic splines

Self-scoring table

	1	2	3	4	Total
Score	**1/1**	**1/1**	**1/1**	**1/1**	**4/4**

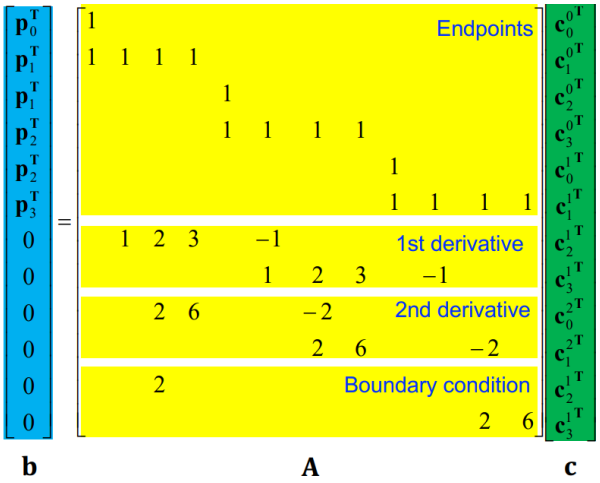
2018707068 김경환

KwangWoon University

Code Analysis:

이번 과제에서는 natural cubic spline을 위한 2D curve editing interface를 구현한다.

먼저 가장 중요한 n+1 point에 대한 n개의 curve segment의 Cubic polynomial coefficient, 총 4n개에 대한 값을 얻 기 위해 4n개의 equations을 구하는 과정을 거친다.



4개의 data point에 대한 Linear System

4개의 data point에 대한 Linear System 이때 해당 4n개의 equation는 endpoint interpolation에 대한 2n개, tangential continuity에 대한 (n-1)개, second derivative continuity에 대한 (n-1)개, 마지막으로 Natural boundary condition 만족하는 2개로 이 루어진다. 이렇게 Cubic polynomial coefficient 4n개를 얻기 위해 A4nx4n*C4nx3=B4nx3을 세우고 eigen에서 제공하는 함 수를 통해 값을 구해낸다. 4n개의 equation이 A4nx4n, Cubic polynomial coefficient이 C4nx, equation에 대한 답이 B4nx3이 된다. Curve의 각 point를 구할 때는 위해서 구한 Cubic polynomial coefficient를 사용하여 Cubic polynomial을 완성하고, 해당 equation에서 0에서 1까지 변하는 변수의 값을 대입하여 좌표를 얻는다. 이때 Cubic polynomial coefficient은 x, y, z의 값을 다르게 갖고 있는 크기 3의 vector이므로 좌표를 얻을 때는 해당 좌표 성분에 맞는 coefficient를 적절히 부여해야한다

위 과정을 통해서 n+1 point에 대한 n개의 curve segment를 Cubic polynomial로 그릴 수 있게 되었다.

이제 2D curve editing interface의 구현에 대한 설명을 하겠다.

첫 번째로 Add는 마우스 좌클릭을 눌렀을 때 data point의 개수를 표현하는 변수 N을 1 증가시키고, data point의 좌표를 저장하는 C++ STL의 2차원 vector에 1차원 vector를 추가한다. 그리고 마우스 커서의 좌표를 얻어와서 N번째 vector에 커서의 x, y, 0을 순서대로 추가한다. 여기서 Curve는 2D에 그려지므로 z좌표는 0을 대입한다. 그리고 마우스 커서의 좌표는 screen 좌표에서 얻어지므로 이를 world 좌표로 바꿔주는 과정을 수행한다. 이러한 원하는 마우스 커서의 좌표를 얻기 위해 좌표를 변환하는 과정은 4가지 기능 전부에서 이루어진다.

두 번째로 Remove는 마우스 좌클릭을 눌렀을 때 data point를 전부 탐색하며 마우스 커서의 위치와 가장 가까운 data point를 찾아내고 이를 remove한다. 이때 커서의 위치와 data point의 위치의 차이가 임의로 부여한 epsilon보다 작을 때만 이를 수행한다.

세 번째로 Drag는 먼저 D키를 눌러 Drag Mode로 변경했을 때 해당 Mode가 여러 번의 Drag 이후에도 계속 유지되도록 하기 위해 마우스 왼쪽 버튼을 놓았을 때 pickMode를 COMPLETE로 바꿔준다. 그리고 Drag를 위한 좌클릭을 눌렀을 때 만약 pickMode가 COMPLETE라면 DRAGGING으로 바꿔주고 DRAGGING에 알맞은 일을 수행해주게 된다. 이제 DRAGGING에 알맞은 일은 먼저 Remove와 유사하게 모든 data point를 탐색하며 마우스 커서와 가장 가까운 data point를 찾고 해당 data point를 move의 이동에 따라 바뀌는 마우스 커서에 맞춰 좌표를 변경해준다.

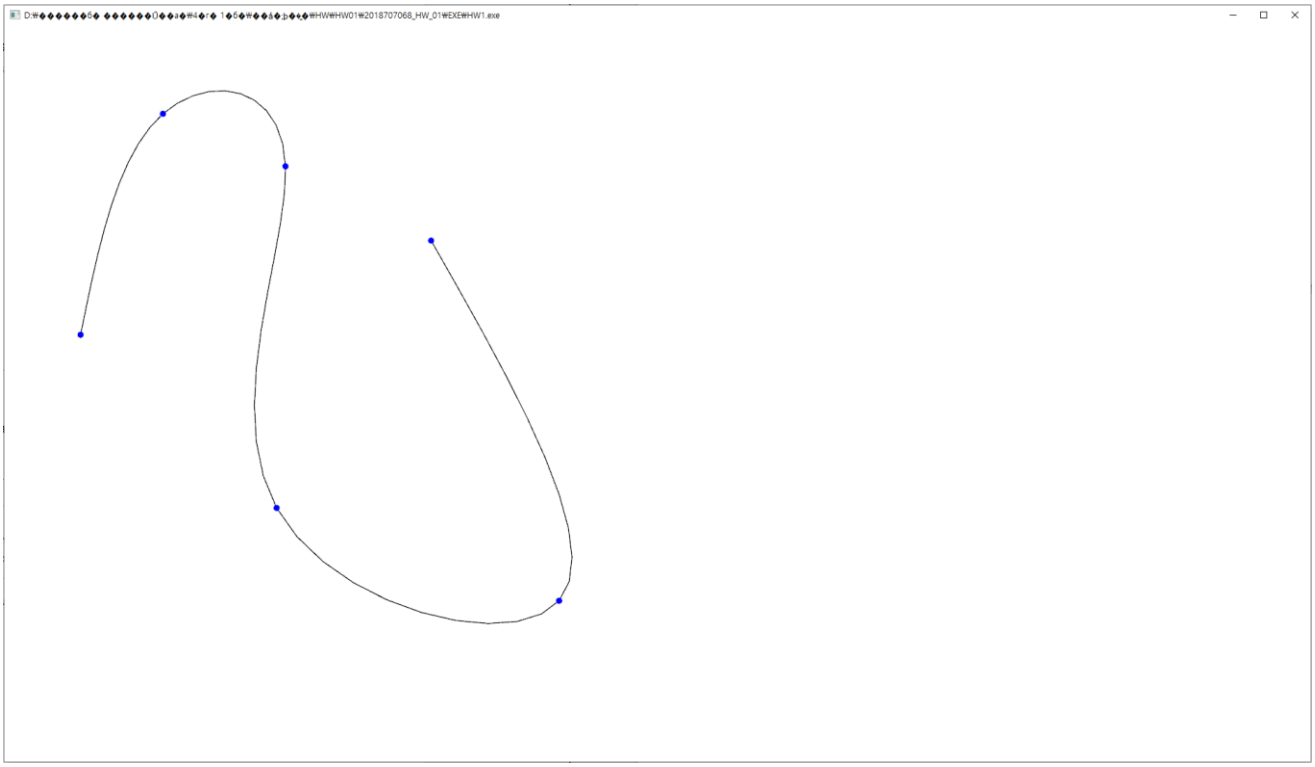
참고로 두 번째와 세 번째에서 마우스 좌표와 data point 사이의 최단 거리를 구하는 방법으로는 두 point를 시작과 끝으로 갖는 벡터의 norm을 구함으로 계산할 수 있었다.

네 번째로 Insert는 먼저 Natural Cubic Spline Curve의 모든 curve segment에서 Sub Segment를 통해 curve 위의 point 좌표를 서로 인접한 2개 얻어낸다. 그리고 2개의 point가 이루는 직선과 마우스 point 사이의 최단거리를

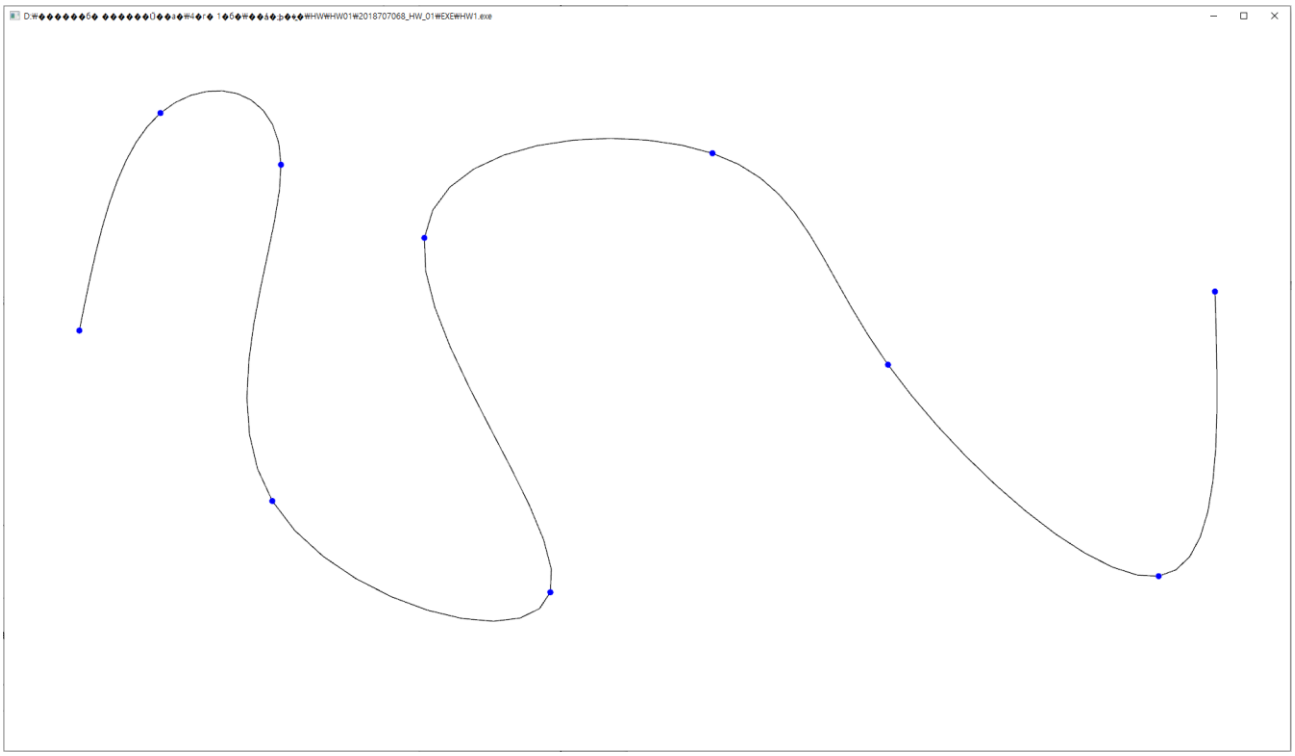
구해 epsilon 보다 작을 때 수선의 발을 계산하여 data point에 추가한다. 이때 data point의 순서에 맞춰 curve segment와 Sub segment의 순서를 고려하여 data point를 적절한 위치에 삽입한다.
참고로 점과 직선의 최단거리를 구하는 방법으로는 직선을 이루는 두 점을 p1, p2라고 하고 점을 p라고 할 때 **V1**(p-p1)와 **V2**(p2-p1)의 cross product를 구한 뒤 **V2**의 길이를 나눠줌으로써 구할 수 있다. 한편 최단거리는 **V1**과 **V2**의 dot product 값이 p1과 p2 사이의 거리보다 작고 **V1**과 **V2**가 이루는 각이 예각일 때만 최단거리가 유의미하다. 그리고 이때 수선의 발 위치를 구하는 방법은 점 p1에서 **V2** 방향으로 **V1**과 **V2**의 dot product만큼 더해줌으로써 구할 수 있다.

1. Add 10 data points (a key):

Mouse Cusor의 Click으로 3개의 Data Point를 추가해 총 3개의 Data Point로 그려진 모습이다.

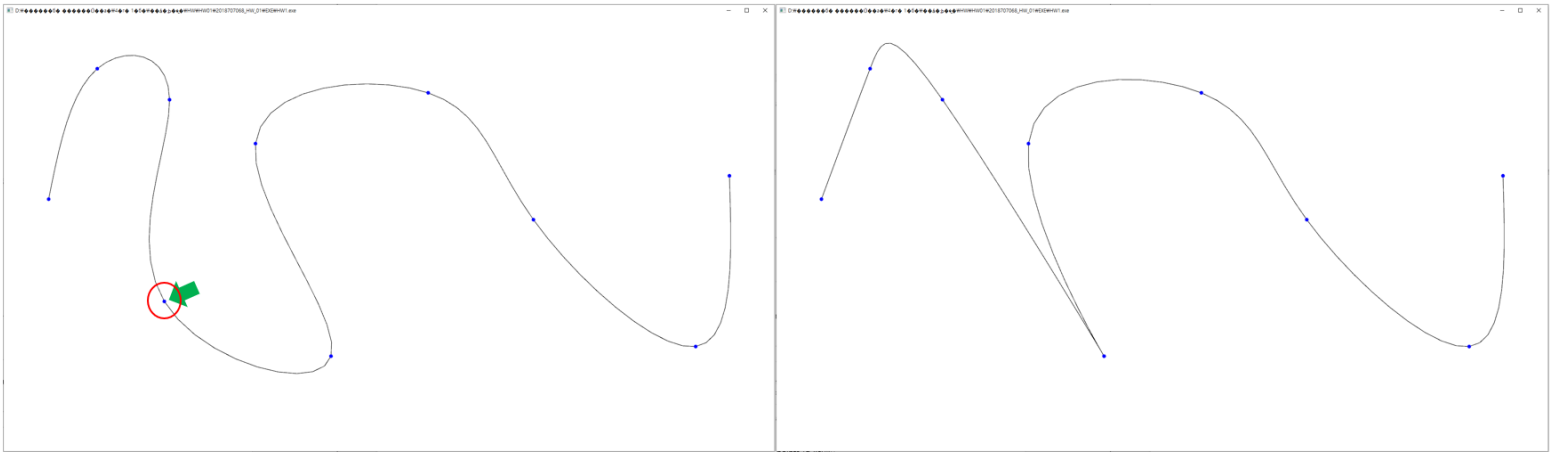


Mouse Cusor의 Click으로 3개의 Data Point를 더 추가해 총 6개의 Data Point로 그려진 모습이다.

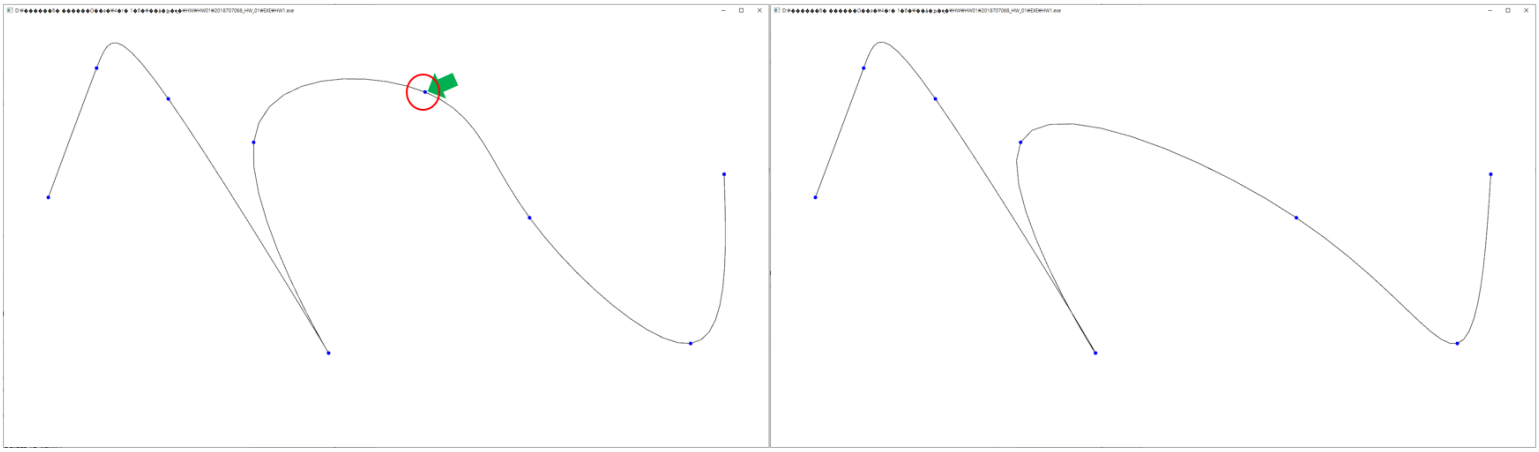


Mouse Cusor의 Click으로 4개의 Data Point를 더 추가해 총 10개의 Data Point로 그려진 모습이다.

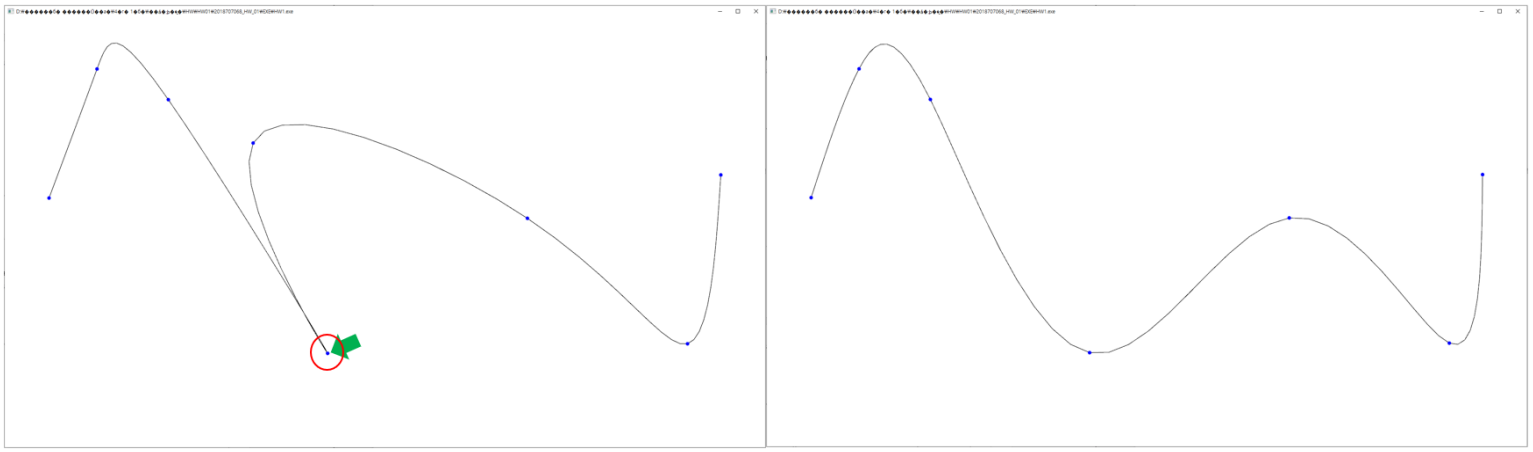
2. Select/remove 3 data points (r key):



빨간색 원과 초록색 커서가 가르키는 Point를 클릭해서 Remove한 모습이다.
가장 왼쪽부터 1번이라고 했을 때 4번째 Data Point를 삭제했다.

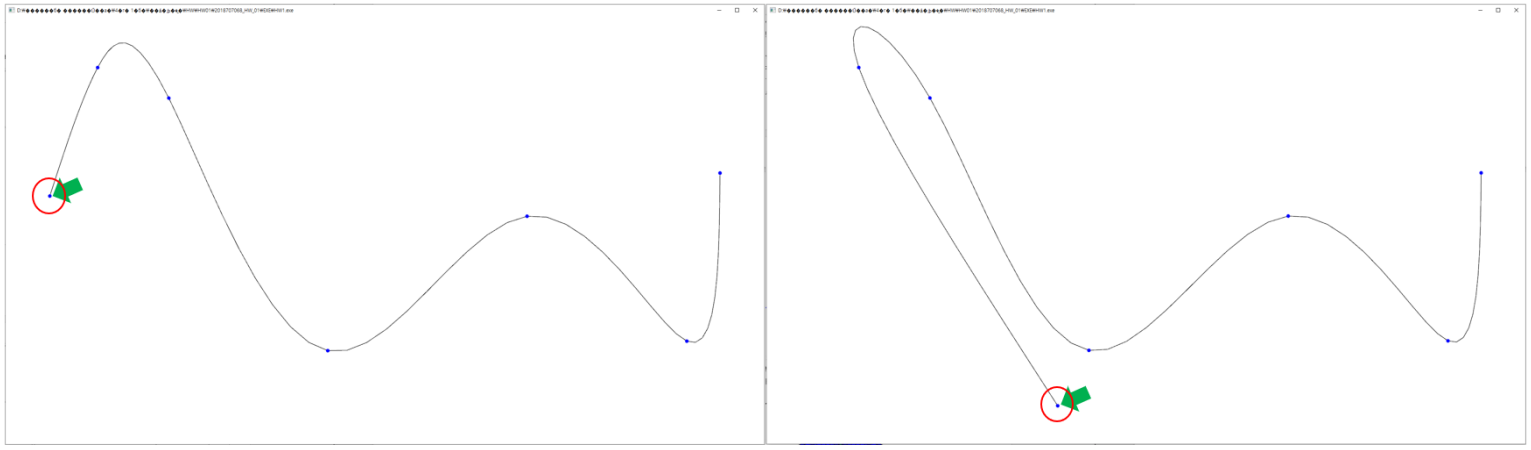


빨간색 원과 초록색 커서가 가르키는 Point를 클릭해서 Remove한 모습이다.
가장 왼쪽부터 1번이라고 했을 때 6번째 Data Point를 삭제했다.

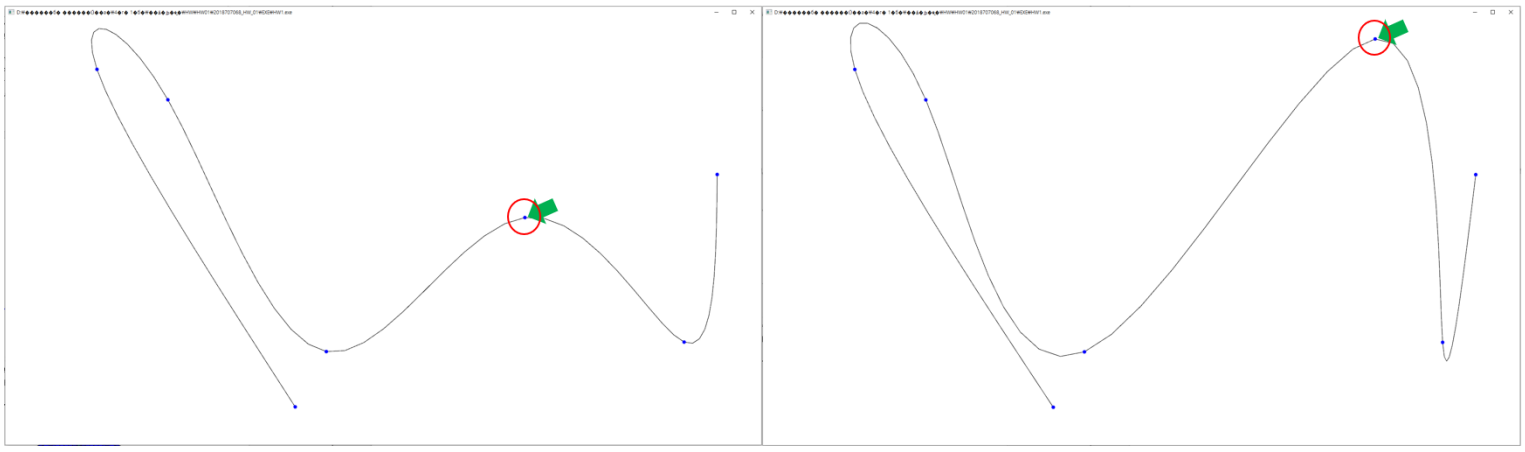


빨간색 원과 초록색 커서가 가르키는 Point를 클릭해서 Remove한 모습이다.
가장 왼쪽부터 1번이라고 했을 때 4번째 Data Point를 삭제했다.

3. Select/drag 2 data points (d key):



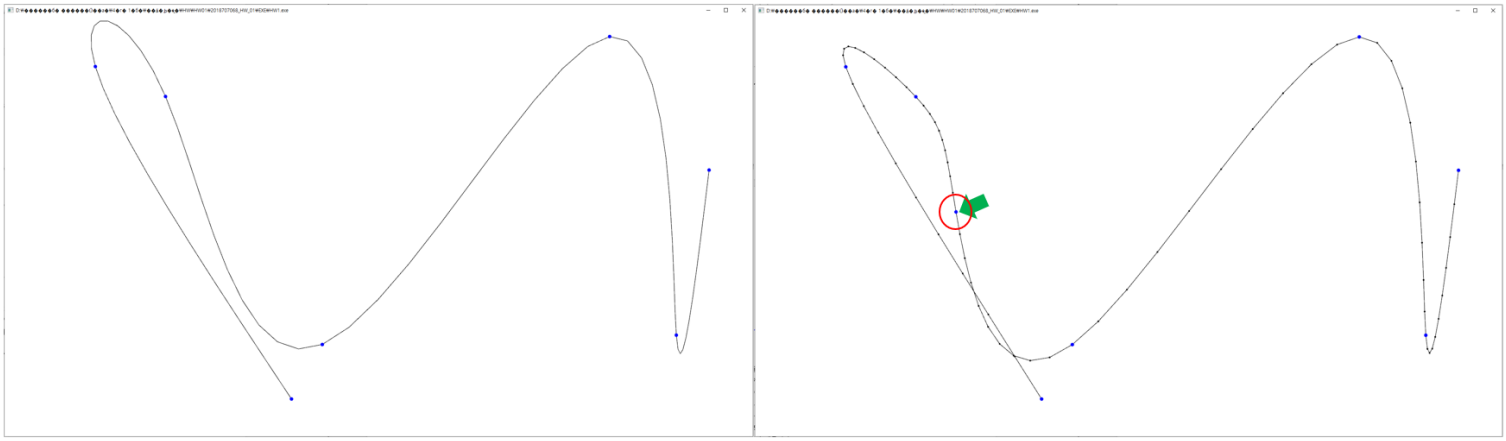
가장 왼쪽부터 1번이라고 했을 때 왼쪽 사진에서 빨간색 원과 초록색 커서가 가르키는 Point에 위치한 1번째 Data Point를 드래그해서 오른쪽 사진에서 빨간색 원과 초록색 커서가 가르키는 Point에 위치하도록 위치시켰다.



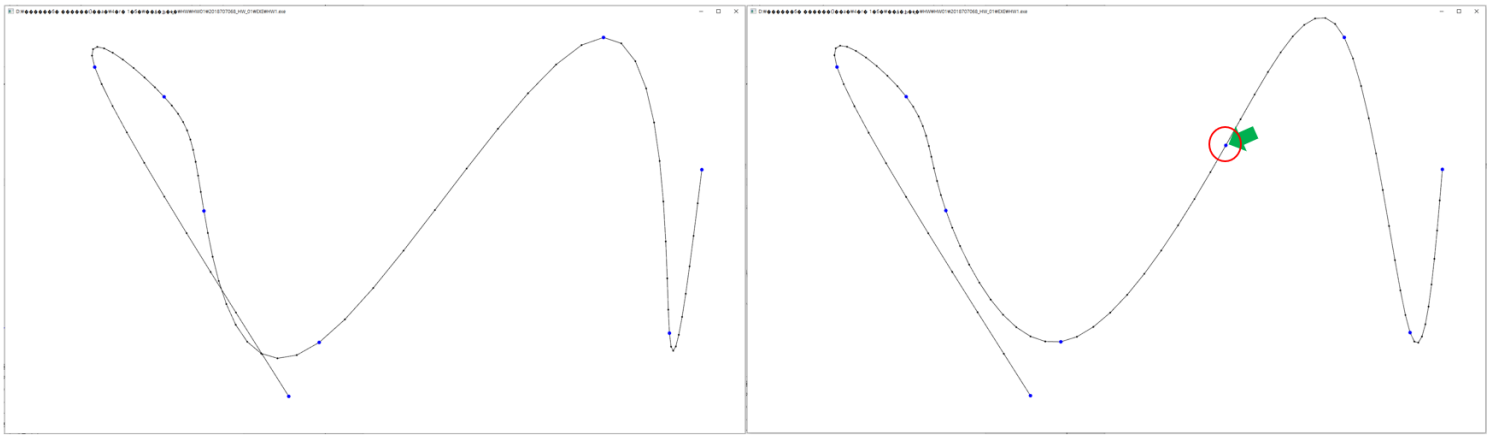
가장 왼쪽부터 1번이라고 했을 때 왼쪽 사진에서 빨간색 원과 초록색 커서가 가르키는 Point에 위치한 5번째 Data Point를 드래그해서 오른쪽 사진에서 빨간색 원과 초록색 커서가 가르키는 Point에 위치하

도록 위치시켰다.

4. Select edges and insert 2 data points (i key):



7개의 Data Point에 의해 만들어진 6개의 Curve Segment 중 3번째 Curve Segment의 3번째와 4번째 sampling point 사이에 커서가 위치하도록 하고 클릭했을 때 해당 cursor에서 line에 내린 수선의 발을 Data Point로 삽입하였다.



8개의 Data Point에 의해 만들어진 7개의 Curve Segment 중 5번째 Curve Segment의 5번째와 6번째 sampling point 사이에 커서가 위치하도록 하고 클릭했을 때 해당 cursor에서 line에 내린 수선의 발을 Data Point로 삽입하였다.