

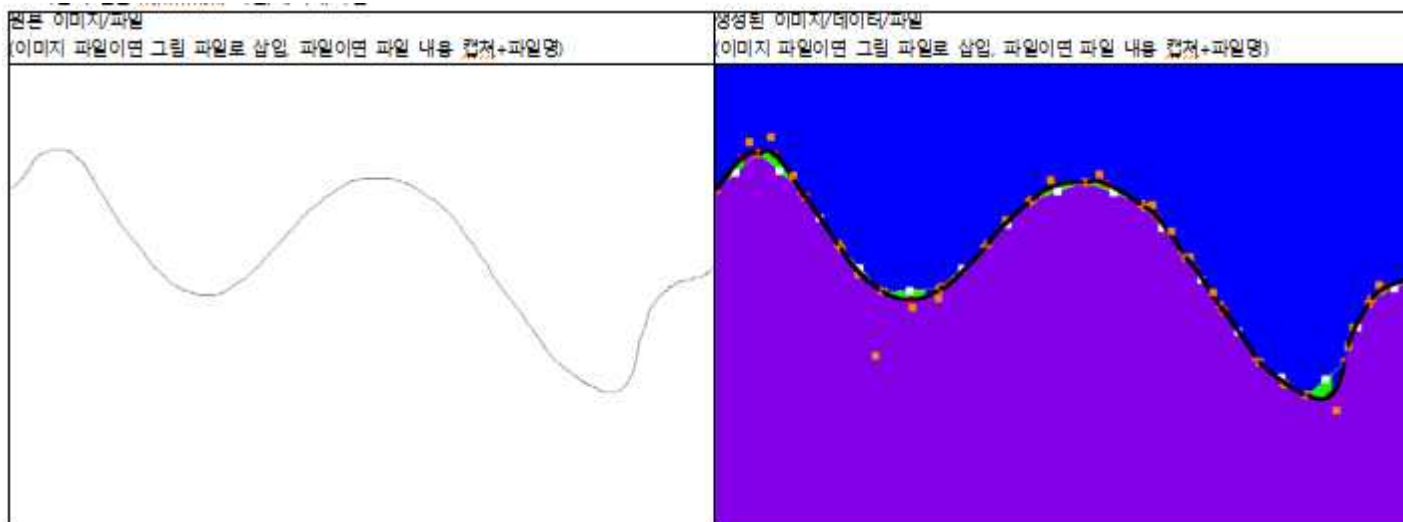
학번 : 20046

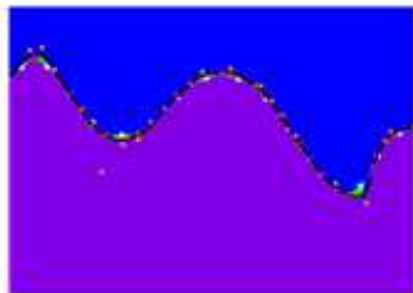
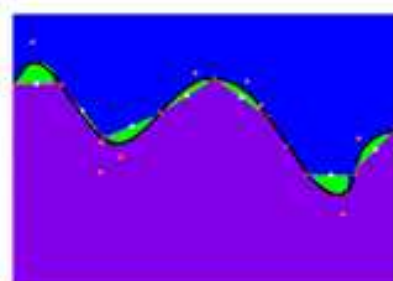
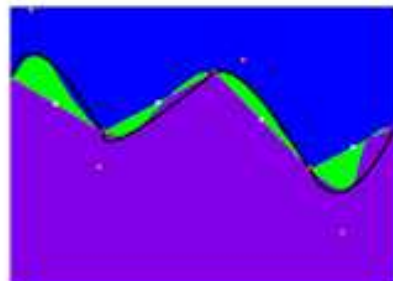
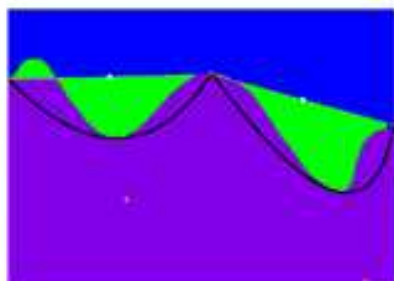
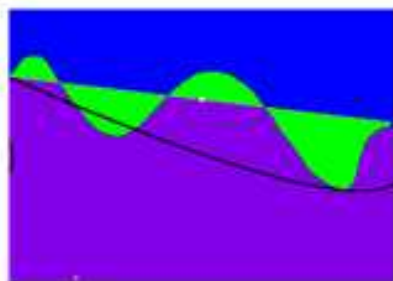
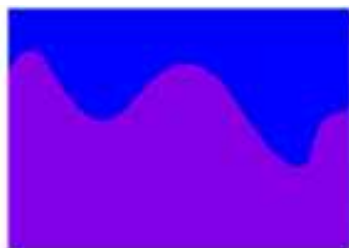
이름 : 박건호

제목 : 베지어 곡선을 이용한 임의의 연속 함수 근사시키기

만든 것에 대한 설명(결과 그림 캡처 후 삽입)

- 관련 수학/과학 교과명 : 수학(기초 R&E)
- 관련 내용 또는 주제 : 베지어 곡선, 리만적분
- 자신이 만든 이미지처리 사진/데이터/파일





이차 베지에 곡선(quadratic Bezier curve)이란?

세 점 P_0, P_1, P_2 와 $0 \leq t \leq 1$ 에 대하여

$\overline{P_0P_1}$ 을 $(1-t):t$ 로 내분하는 점을 Q_0 , $\overline{P_1P_2}$ 를 $(1-t):t$ 로 내분하는 점을 Q_1 이라 할 때,

이차 베지어 곡선 위의 한 점 $B(t)$ 는 $\overline{Q_0Q_1}$ 을 $(1-t):t$ 로 내분하는 점이다.

이 때 세 점 P_0, P_1, P_2 를 베지어 곡선의 조절점(control point)라 한다.

식으로는 다음과 같이 나타나진다.

$$Q_0 = (1-t)P_0 + tP_1$$

$$Q_1 = (1-t)P_1 + tP_2$$

$$\begin{aligned} B(t) &= (1-t)Q_0 + tQ_1 \\ &= (1-t)[(1-t)P_0 + tP_1] + t[(1-t)P_1 + tP_2] \\ &= (1-t)^2P_0 + 2t(1-t)P_1 + t^2P_2 \end{aligned}$$

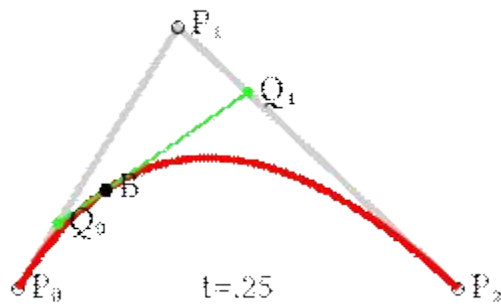


Fig 1. $B(t) = (1-t)^2P_0 + 2t(1-t)P_1 + t^2P_2, t = 0.25$

근사시키는 방법 은

1. 양끝점을 잇는 선을 만든다
2. 선 기준으로 원래 곡선이 얼마나 떨어져 있는지 수직거리를 통해 구한다.
3. 수직거리가 최대인 점을 기준으로 노란색 선의 중간 점을 점 대칭시킨다, 그리고 그 점이 우리가 그릴 베지어 곡선의 P1이 될 것이다.
4. P0 P2 는 구간의 양끝점을 잡으면 된다.
5. P0 P1 P2 가 정해졌으니 우리는 이를 통해 베지어 곡선을 그릴 수 있다.
6. 저번 processing 에서는 하우스도르프 거리 기준으로 재귀 과정을 몇 번 돌릴지 결정하였지만 이번에는 하우스 도르프 거리를 구현하기 어려워 그냥 몇 번 근사할지 scanf를 하고 그 횟수만큼 1번에서 그 횟수만큼을 반복한다. 구간은 곡선의 길이를 선적분한 길이에서 절반이 되는 점을 기준으로 나눈다.

함수 소개

void Bezier(int P0x,int P0y,int P1x,int P1y,int P2x,int P2y) P0 P1 P2가 정해졌을때 베지어 곡선을 그려줍니다.(굵은 검은색으로 체크합니다)

int Pcheck(int low,int high) P0 P1 P2를 정해줍니다.(점대칭 시켜줍니다)(그리고 이를 주황색으로 체크해줍니다)

void Ymid(int low,int high) 노란선의 중간점을 흰색으로 체크해줍니다.

void ctmax(int low,int high) (노란색 선으로부터 각각이 원래 곡선과 얼마나 떨어져 있는지 배열로 알려줍니다)

void bugcheck(int low1,int high1,int low2,int high2) (가끔씩 정확도를 알려주는 연두색이 빠져나올 때가 있는데 이를 보정하여 줍니다)

void green(int low1,int high1,int low2,int high2) 연두색으로 정확도를 알려줍니다.

void upblue() 윗부분을 오류가 안 나도록 완전한 파란색으로 칠해줍니다

void line(int low,int high) 노란색 선을 그려줍니다.

void checkblack2(int where) 어느 x 값에서 검은색이 어디에 위치되는지 알려줍니다.

void checkblack1(int where) 어느 x 값에서 검은색이 어디에 위치되는지 알려줍니다.(위와 다른 변량에)

void half(int low1, int high1,int low2,int high2) 선적분을 하여 길이의 중간 점을 찾아줍니다) 빨간색으로 표시합니다)

void print2() 이미지를 출력합니다.

void print1() 이미지를 출력합니다.

void color(int low1, int high1,int low2,int high2) 검은색 선 기준으로 아래는 보라색 위를 파란색으로 칠해줍니다.

void colorblack() 검은색 원래 선을 다시 그려줍니다

void blackcheck() 검은색 원래 선의 위치를 체크하여 둥니다.(배열에)

void check2() 이미지에서 색깔을 인식합니다.

void check1() 이미지에서 색깔을 인식합니다.

void header1() 헤더를 옮겨줍니다.

void header2() 헤더를 옮겨줍니다.

void cleaner(int low,int high) low부터 high 까지 흰색을 초기화시켜 줍니다.

void goBezier(int low,int high) 구간이 정해졌을 때 추세선을 그려주는 총괄 함수입니다.

void makeBezier(int low,int high,int k) 구간을 나누어 추세선을 그리도록 명령하는 총괄 함수입니다.

참고한 예시, 내용, 출처

참고한 예시는 없지만 내용은 베지어 곡선의 위키백과 내용을 참고하였다.

자신이 창의적으로 만들고 구현해본 내용

처음에는 어떤 이미지의 색상 변화와 같아 색깔을 바꾸거나 , 태두리를 따는 것(하지만 명암 때문에 생각보다는 잘 안되었다) 등을 구현해보았다

만든 것의 수학 과학 정보 관점의 의미 활용 가능성

수학적 과학적 활용 가능성은 엄청 많이 찾아 볼 수 있다. 먼저 베지어 곡선은 세 점만 잡으면 어떤 부드러운 곡선을 그릴 수 가 있다. 우리는 이를 원래는 점을 하나하나 찍어야 그래프가 나오지만 점 3개만 이용하면 만들 수 있다는 점에서 활용 가능성이 매우 높다. 그리고 베지어 곡선 자체를 그리는데 데에 메모리를 별로 잡아 먹지 않지만 심지어 베지어 곡선뿐만이 아니라 임의의 연속된 곡선만 이용

된다면 활용가능성이 더 올라간다. 처음에는 잘 근사가 안 된다는 단점이 있긴 하지만 원호만 가지고 해본다면 4번만에 10^{-3} 만큼의 오차를 보인다. 즉, 1번 1번 할때마다 근사 정확도가 엄청 빠르게 증가한다는 의미이다. 우리가 이를 임의의 이미지에 활용하기만 한다면 이 이미지를 저장시키는 메모리를 엄청 축소시킬 수 있다. 그리고 전의 processing에서 만든 작품은 함수식이 들어가야 근사가 된다는 단점이 있었는데 그림판에 양함수만 그리면 쪽 작동되도록 만듦으로서 더 알앤이가 확장된 것 같았다. 그리고 이를 이용하여서 samsung 팬기능에 도형을 그리고 2번 연속으로 클릭하면 딱 교정해주는 기능을 만들 수 있다. 그리고 이를 애니메이션을 만드는데 이용한다면 더 우리가 보기에 더 부드럽게 움직이게 보이도록 애니메이션의 발전에 더 기여할 수 있으며 이 방법은 3차원에서도 유효하여 3차원 곡선을 근사시키는데도 쓰일 수 있다. 그러면 3차원 모양을 사영시킴으로서 더 부드럽게 보이도록 교정할 수도 있으며 그냥 3차원 모션으로 홀로그램등에도 쉽게 쓰일 수 있다.

만들면서 시도해보거나 시도해본 것들

처음에는 양함수에 관련된 것만 만들고 이에 성공하였는데 이를 다양한 이미지에 접목시킬 수 있도록 임의의 닫힌 곡선 즉, 양함수로 표현되지 않는 곡선에 대하여 이를 적용시킬 수 있도록 하는 방법에

대하여 생각해보았다. 그래서 어떤 태두리가 있을 때 양함수로 쪼개서 각각을 근사시키고 이를 합치는 방법에 대해 생각해보았다. 그래서 이를 닫힌 구간 내에 있는 한 점을 원점으로 설정하고 극좌표에 적용시킨다면 되지 않을까라는 생각을 해보았지만 생각보다 잘 되지 않았다. 그래서 다음에는 어떤 한 점을 기준으로 주변을 탐색하여 태두리를 따는 방법으로 이를 해보야겠다라고 생각하였다.

만들면서 이해하게 되었거나 느낀 점들

근사시키는 과정에서 시작점과 끝점을 잇고 탐색시키는 과정이 있는데 잇는 과정에서 각 픽셀이 int 이기 때문에 약간씩 끊기는 현상이 있는데 이 때문에 끊기는 부분에서 오류가 좀 발생하였다. 이렇게 사선을 긋는 부분에서 버그가 안 나도록 추가적인 함수를 짜고 특히 버그가 어디에서 나는지 확인할 때 되게 힘들었다.

만들면서 가장 많은 도움을 받은 사람

추세선을 그리는 방법과 전반적인 내용은 제가 다 만들었지만 그 중에서 가장 도움을 많이 받은 사람

은 김진휘형이다. 그 정확도를 나타내는 초록색 부분을 계속 그리는데 이상한 부분에서 오류가 났었다. 그것이 가끔씩 빠져나오는 초록색 부분이 있어 그렇다는 것을 알려주었다. 그리고 이 버그는 노란색 선 기분으로 위아래가 보라색이면 위로 탐색 위아래가 파란색이면 아래로 탐색하도록 while 문에 들어나는 수에 (-1)를 계속 곱해나가도록 코드를 수정하였습니다

나는 만들지 못했지만, 누군가 만들어 주었으면 좋겠다? 라고 생각하는 주제/ 내용?

나는 만들지는 못했지만 만들어 주었으면 좋겠는 거는 지질도이다. 그리고 우리 눈이 아닌 곤충이 보는 세상 그리고 그 전에 선배들 거에서도 보았던 지도를 지구본의 구형으로 변환시키기 같은 방법을 누가 만들어 주었으면 좋겠다. 그리고 embossing도 누가 만들어 주었으면 좋겠다라고 생각했다.

자신의 진로와 관련되어 이미지 파일 처리 프로그래밍의 활용 방법 및 의미

저의 진로는 수학쪽이여서 이미지 파일처리 프로그래밍이 특히 더 중요한 것 같다. 지금은 processing 이 더 자세하게 표현할수 있는 것 같지만 이 파일 처리에 관한 코드를 많이 보면 이미지 파일 처리가

더 활용 가능성이 높은 것 같다. 특히 여러 파일을 동시에 처리한다는 점에서 되게 유용한 것 같다. 이 코드에서도 fopen fclose를 이용하여 많은 파일 처리를 동시에 하였는데 되게 편하였다. 이를 이용하여 어려운 그림을 여러 개 그려야 할 때가 오면 이미지 파일 처리를 써야 할 것 같다.