



PARAMEISTER & COMPANION

멋진 예술공간을 만들어주는 Parametric Wall
전용 Design Program

Parametric Wall은 예술성에 힘입어 멋진 공간의 연출은 물론 공간의 음향 특성 개선에 탁월한 효과를 발휘합니다. 나만의 Design에서 제작에 이르는 모든 과정을 지원하는 신개념의 Parametric Wall 설계 프로그램을 소개합니다.

yskim57@gmail.com

목차

1. Parametric Wall 이란?	2
2. Image 기반의 Parametric Wall Designer (Parameister)	4
2.1. 화면 Layout.....	4
2.2. 호환 Image Type 및 화소값의 범위	5
2.3. 실행 Button.....	5
2.4. Application Program launcher.....	23
3. 조각 경로 작성 Tool (PathMaker)	24
3.1. 화면 Layout.....	24
3.2. Button.....	25
3.2.1. Level.....	25
3.2.2. Track File 관리.....	25
3.2.3. Trace 의 생성/삭제/변경	26
4. Geometric Layout Designer (GeoLaD)	34
4.1. 화면 Layout.....	34
4.2. Button 및 Parameter.....	35
4.2.1. Imaging Region Parameter 와 Button:.....	35
4.2.2. File 관리	35
4.2.3. Layout Item.....	39
4.2.4. Item 중첩 Image 생성	46
5. Reference	48
5.1. Video	48
5.2. Woodmeister 사용법	48
5.3. 목공과 Coding 의 만남	49

1. Parametric Wall이란?

Parametric Wall은 예술성과 음향 효과를 모두 갖춘 최상급 Interior의 용도에 많이 사용하고 있습니다. 다양한 재질로 이를 구성할 수 있으나 일반적으로 목재 또는 이와 비슷한 성질의 재료를 많이 사용하고 있습니다. 아래의 그림은 Pinterest에서 Parametric Wall을 입력하여 나열한 예로써 매우 다양한 Pattern을 사용하여 만들 수 있으며 그 용도에 따라 Pattern의 Design 방법도 서로 달리할 필요가 있습니다.

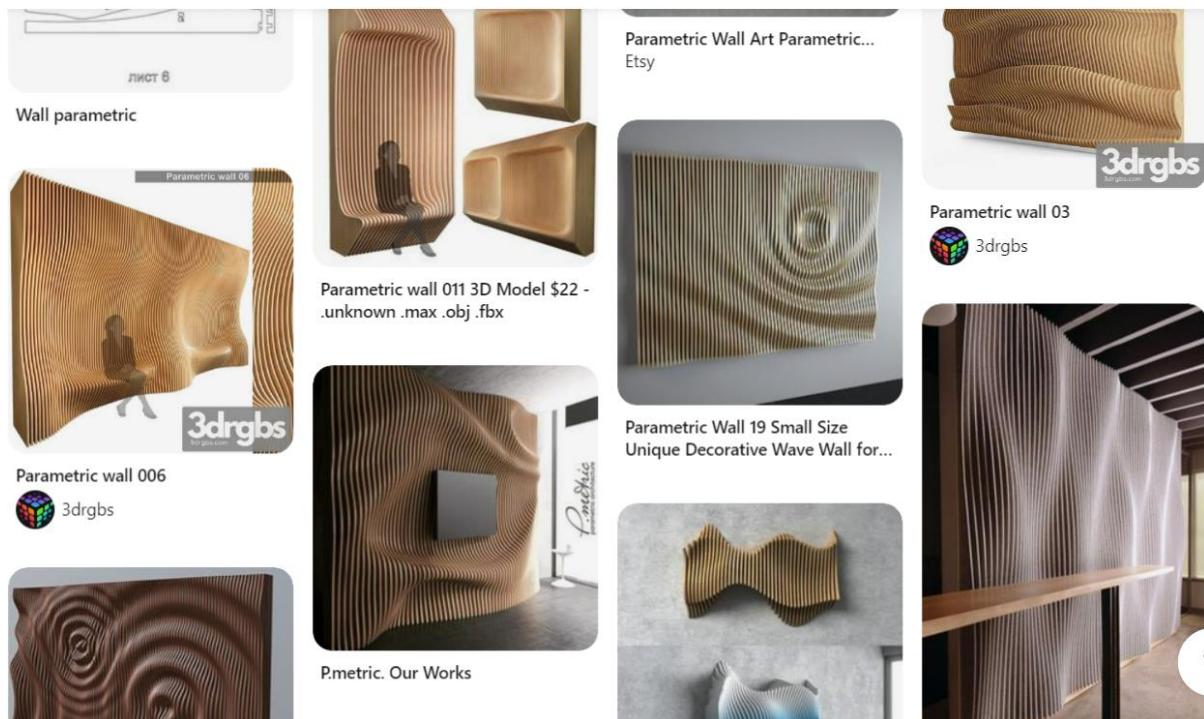


그림 1. Parametric Wall의 예

본 문서에서는 Image를 기반으로 만드는 Parametric Wall 설계 방법 및 이를 구현한 전용 Design Program인 Parameister와 관련 프로그램에 대한 설명을 하고자 합니다

아래의 그림은 이의 전형적인 화면의 하나로 Parametric Wall의 기본이 되는 Image를 열어 놓은 상태의 화면을 보여줍니다.

3 목공과 Coding 의 만남 – Parametric Wall Design Application (Parameister)

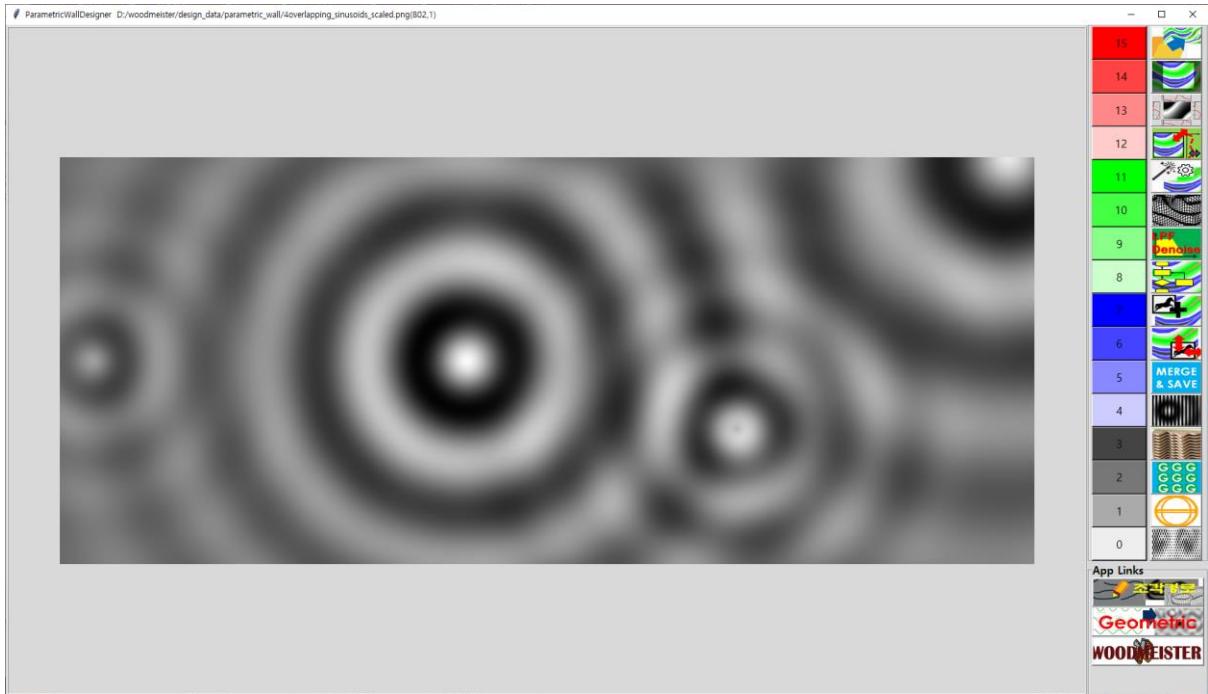


그림 2. Parameister의 실행 화면 예

이 Image를 처리하여 1440mm x 600mm 의 크기로 제작한 Parametric Wall의 Sample이 그림 3이다.



그림 3. 그림 2의 Image에 기반하여 만든 제작 Sample (크기 : 1440mm x 600mm)

2. Image 기반의 Parametric Wall Designer (Parameister)



Parameister는 Python으로 작성한 프로그램으로 Image의 처리를 통하여 Parametric Wall을 Design하고 제작을 위하여 필요한 여러가지 가공 부품의 도면을 Woodmeister에 호환되는 양식으로 생성해낼 수 있는 프로그램이다. 그림 1이나 그림 3에 표기된 Parametric Wall의 제작에는 각각의 위치에 따라 상면의 굴곡이 서로 다른 여러 개의 패널들을 필요로 한다. 이들 패널을 최적의 방식으로 배치하고 가공하는 것은 매우 중요하고도 시간이 많이 소요되는 후 공정 작업이다. 이 같은 후 공정 관리 부분도 프로그램에서 지원하는 분야로 소재의 낭비가 없이 빠른 속도로 가공할 수 있는 여러가지 기능들을 내장하고 있다. 그림 4는 이 프로그램을 처음 수행하였을 때의 화면이다.

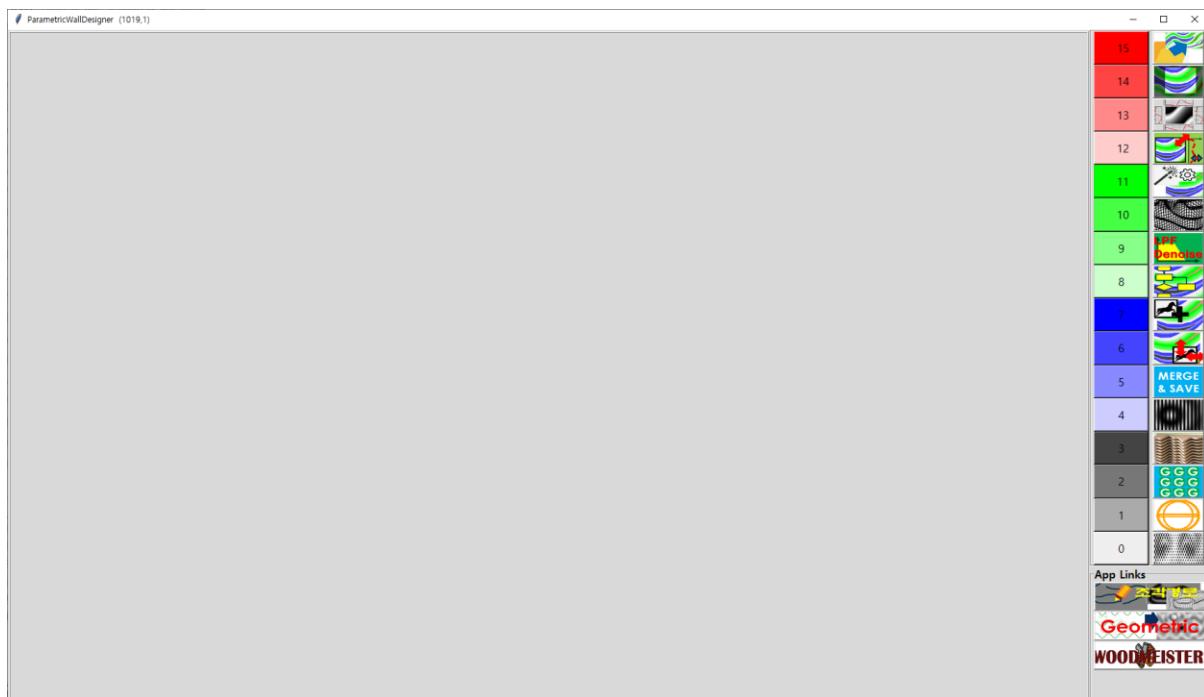


그림 4. Parameister의 수행 후 첫 화면

2.1. 화면 Layout

그림 4의 오른쪽에 16가지의 대표적인 Level을 나타내는 색상을 가지는 color (Level) Button들과 그 오른쪽에 Image의 조작 등에 활용하는 Button들이 나열되어 있다.

현재 이 프로그램에서 Level에 관한 Button은 사용하지 않고 있으며 그 오른쪽에 위치한 Button들에 기능이 부여되어 있다.

Screen의 대부분을 차지하는 왼쪽 부분이 Design 화면으로써 여기에 처리하고자 하는 Image와 이에 부수되는 정보들이 표기될 수 있다.

기능이 부여된 Button들은 크게 4개의 부류로 나뉘어 있으며 이들 4개의 부류는 Image 입출력 및 처리, Image 생성/합성, 가공용 CAD/CAM data 생성 및 관련된 Application Program 호출 Button들이다.



그림 5. 기능이 부여된 Button 들의 분류

2.2. 호환 Image Type 및 화소값의 범위

Parameister 및 이에 연동되는 Application program들은 ".png" type의 Image format을 사용한다. 이들은 8bit, 16bit, color Image들이며 모든 화소값은 정규화 과정을 거쳐 0 ~ 255의 값으로 표현된다. 16bit image의 경우 pixel 값을 256으로 나눈 값을 화소값으로 취급한다. 또한 Color Image의 경우도 gray level 변환을 통하여 16bit Image로 변환되고 정규화 과정을 통하여 0 ~ 255의 값으로 표현된다. Image의 화소값 1은 높이 1mm에 대응하도록 프로그램 되어있다.

2.3. 실행 Button



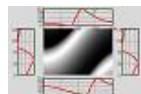
작업을 진행하고자 하는 기본 Image를 열어서 화면에 띄워 준다. 여기에 사용하는 Image는 가로/세로의 크기가 각각 화소수로 1599 x 990이 넘지 않도록 한다. 만일 크기가 이보다 크다면 다른 편집 수단을 사용하여 이 크기 이하로 줄여서 처리하는 것을 권장한다. 또한 Image의 각 화소는 16bit 값을 가지는 것이 바람직하다. 이를 위하여 가급적 16bit PNG Image를 기반으로 작업하는 것을 권장한다.

Image의 한 화소는 1mm X 1mm의 크기로 취급되며 필요한 부분에서 축소 또는 확대를 할 수 있다. 예를 들어 800 X 600의 Image라면 이는 크기가 800mm X 600mm인 것으로 취급된다.



- **Crop Image.**

선택된 영역을 잘라내어 다른 이름으로 저장하도록 한다. 영역 선택을 위하여 왼쪽 최상단점에서 Mouse 왼쪽Button을 누른 상태로 오른쪽 하단까지 Drag한 후 Release한다. 선택된 영역이 초록색 점선으로 표기된 사각형으로 표기된다. 이 후 CROP Button을 눌러서 선택된 영역을 새로운 Image로 저장할 수 있다. 저장은 16bit PNG File로 이루어 진다.



- **Image Profile.**

Image의 개략적인 단면 모습을 그림 6과 같은 방식으로 표기한다. Image외곽 네 개의 변에 표기되는 Graph중 붉은색으로 표기된 것은 해당 변의 Gray Level 분포를 0~255 사이의 값으로 표기한 값이며 녹색으로 표기된 그래프는 해당 Row 또는 Column 상의 최대치 또는 최소치를 표기한다.

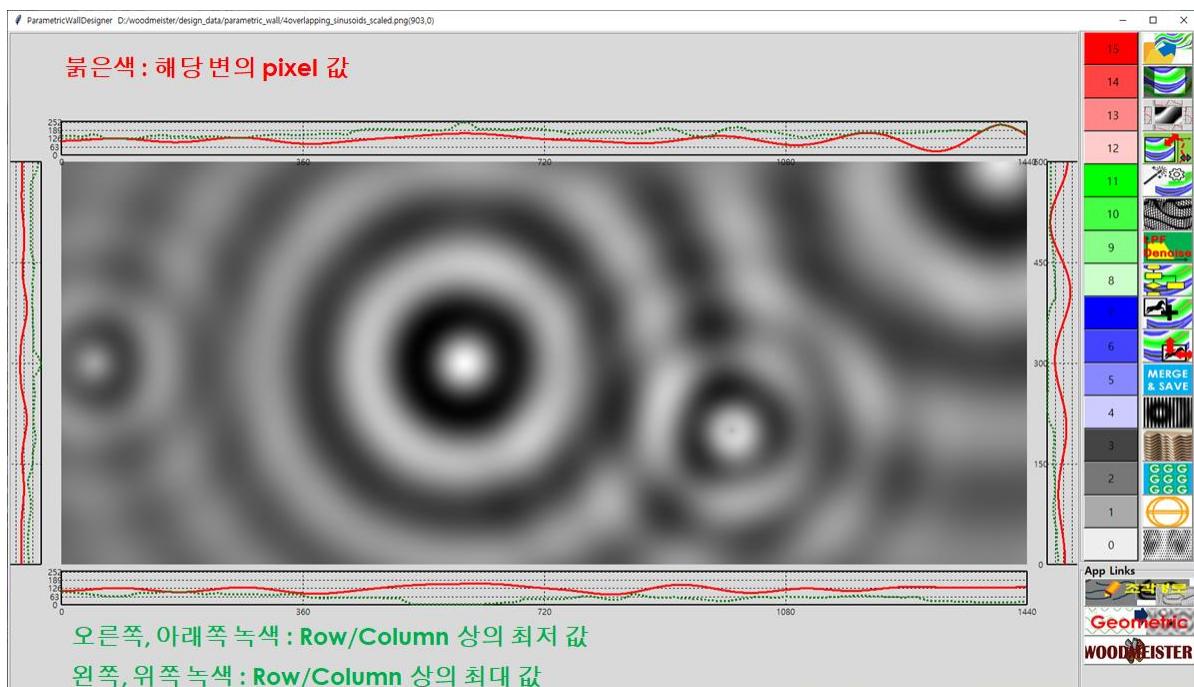


그림 6. Image Profile Button을 누른 후의 화면. Image의 개략적인 Profile을 표기하여 준다. 붉은색 선: 해당 변의 Pixel 값을 0~255의 값으로 Normalize한 값을 표기. 녹색선: 해당 row 또는 column 상의 최대치(위, 왼쪽) 및 최저치(아래, 오른쪽)를 0~255의 값으로 Normalize한 값을 표기.

- 크기 및 Gray Level 조정.

Base Image의 크기 및 Gray Level을 조절하여 준다.

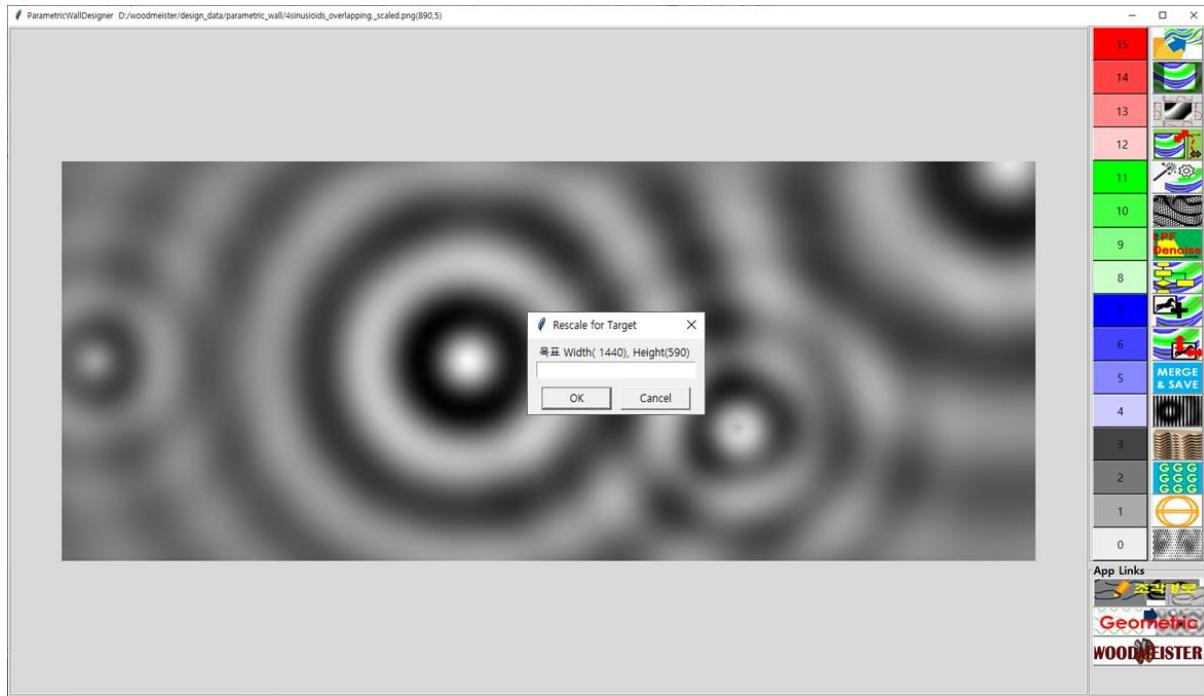
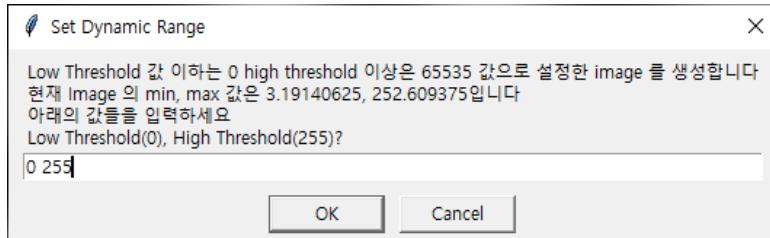


그림 7. Base Image Rescale 화면.

그림 7의 화면에서 수평 수직 크기를 원하는 값으로 설정한다. 단위는 mm이다. 크기를 1600 x 640으로 변환하고 싶으면 아래와 같이 값을 입력 후 "OK"Button을 누른다.



이후 gray Level 값을 조절한다. 이는 아래와 같은 화면에서 진행한다.



완료가 된 후 Image는 기본Image File 이름+ "_scaled.png"로 저장된다. 작업이 끝난 후 생성된 Image를 열어서 화면에 띄워주며 이것이 Base Image로 취급된다.

- Image 처리.

화면에 표기된 Image의 화소값을 변경하여 새로운 Image로 저장하고 화면에 띄워준다.

- Invert

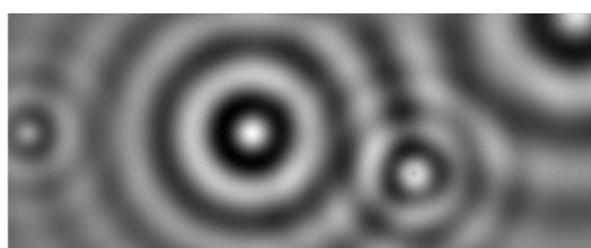
화소의 값을 역전시킨다. 최대값->최소값,...,최소값->최대값

저장 Image는 기본Image File 이름+ "_inverted.png"로 저장된다.

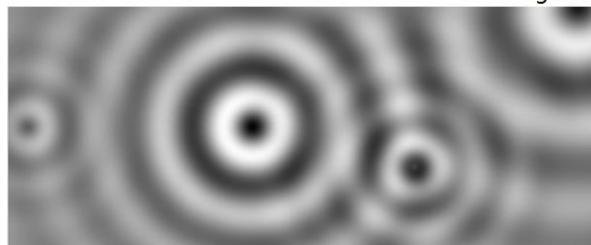
- 이진화(Binarization)

임계치 이상을 high, 이하는 low 값으로 변환한다.

저장 Image는 기본Image File 이름+ "_binary.png"로 저장된다.



Original Image



Inverted(Negative) Image



Binary(이진화) Image

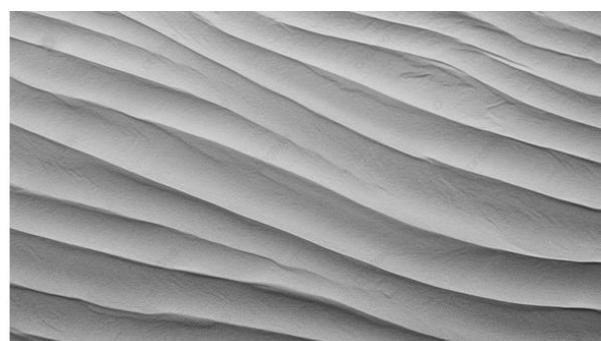
그림 8. 원래의 Image(위)와 Inverted(왼쪽 아래), 이진화(오른쪽 아래) Image

- Grayscale

Color Image를 흑백Image 로 변환시킨다.



Color Image



Grayscale Image

그림 9. Color Image의 Grayscale Image 변환. Color Image를 16bit Image로 변환한다.

- 컬러지형도 변환

Color로 표기된 지형도를 흑백지형도로 변환한다. 지형도에서 사용하는 Color에 따른 고도 값으로 각 Pixel을 생성하여 16bit gray Level Image로 변환 저장한다.



Color 지형도



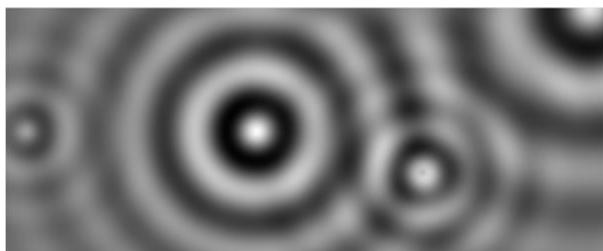
흑백 지형도

그림 10. Color지형도의 흑백 지형도 변환. Color지형도를 16bit Image로 변환한다.

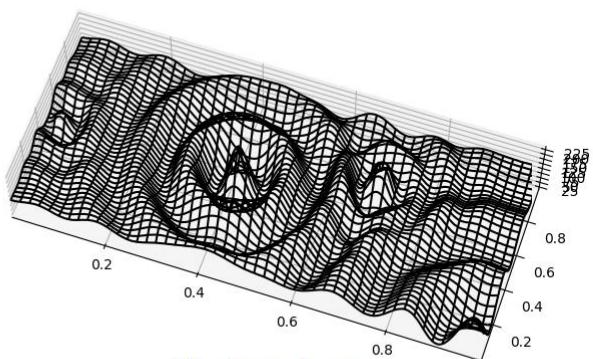
- Wire Frame.



Image의 각 화소값을 높이로 간주하는 3차원 View를 보여준다. View는 Wire Frame 형식으로 표현한다.



Base Image



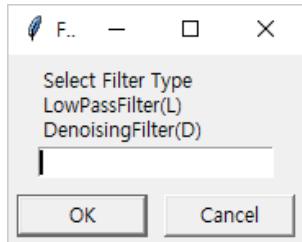
Wire frame Image

그림 11. Base Image와 이것을 Wire Frame으로 표현한 결과. Wire Frame은 제작 및 설치가 완료된 후 대략의 형태를 3차원으로 보여준다.

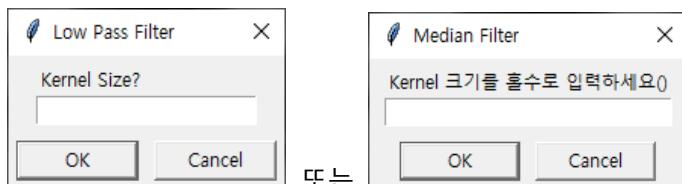
- Low Pass Filter/잡음제거.



Base Image가 가공 및 표현이 부적절한 지나치게 세세한 부분 또는 Noisy Pixel (Pepper Noise 등)을 포함하고 있을 때 이를 완화시키기 위해 저역 Filter(Low Pass Filter) 또는 잡음제거 Filter (Denoising Filter)를 적용하여 부드러운 Image로 변환한다.



Low pass Filter와 Denoising Filter의 경우 각각 아래와 같은 창에서 Kernel의 크기를 입력한다. Kernel은 처리를 위한 연산을 할 때 한 개의 Pixel에 대한 처리를 할 때 포함하는 수평 및 수직 방향 인접화소의 갯수를 나타낸다.



와 같은 화면에서 Kernel의 크기를 입력 한다.

- ✓ Denoising의 경우 base Image 이름 + "_denoise.png"로 저장된다.
- ✓ Low pass Filter의 경우 base Image 이름 + "_lpf.png"로 저장된다.



● Geometric Image 생성.

수학적인 방법으로 다양한 형태의 Image를 생성한다. 이를 실행하여 여러 방식으로 Image를 생성할 수 있다. 이를 생성하는 프로그램은 다양한 Needs를 즉시 반영할 수 있도록 Source Program으로 관리한다. 이를 수행하는 프로그램은 "_gen_geofigure.py"이며 이 프로그램을 수정하여 사용자의 용도에 맞는 다양한 Image를 생성하여 사용할 수 있다. 아래는 필자가 만들어 놓은 프로그램을 사용하는 경우의 첫 수행 화면이다. 여러가지의 Option 중 일반적으로 사용이 가능할 것으로 생각하는 것에 대한 실행 결과를 Image로 표기하였다.

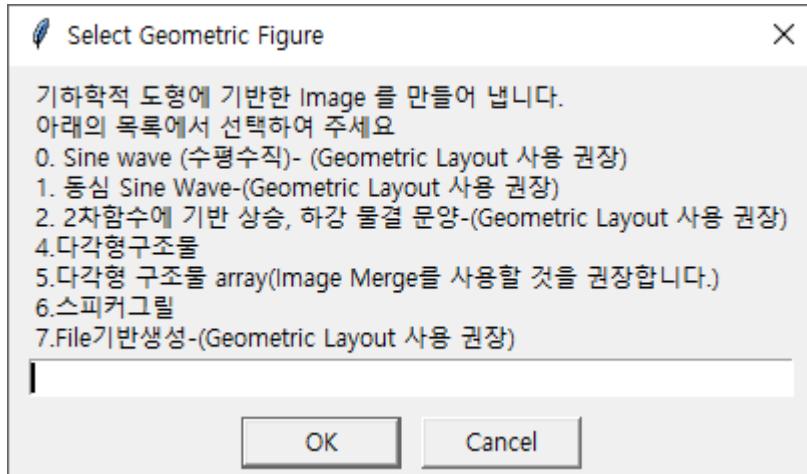


그림 12. “Algorithm 기반 Image 생성” 메뉴 실행 첫 화면의 예

- Sine wave (수평 수직)

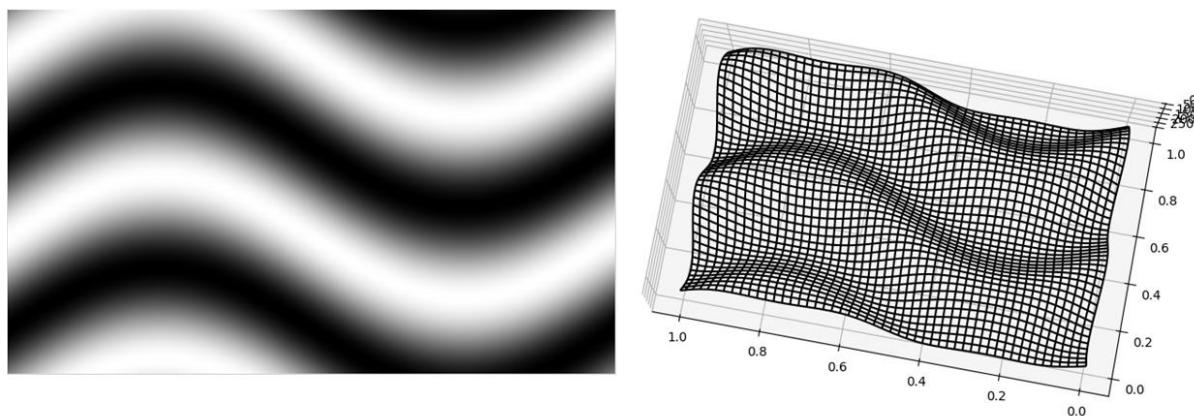


그림 13. 수평 수직 Sine Wave

- 동심 Sine Wave`

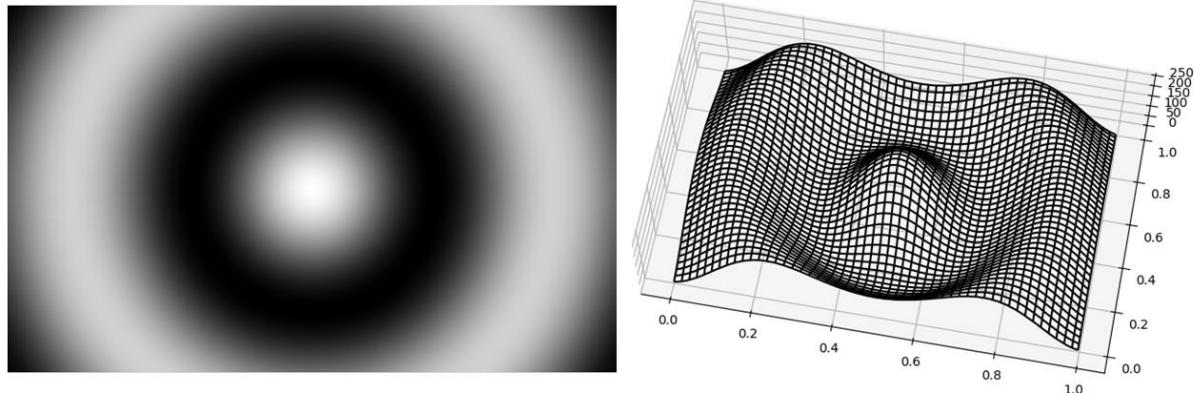


그림 14. 동심 Sine Wave

- 2차함수 기반 상승, 하강 물결 문양

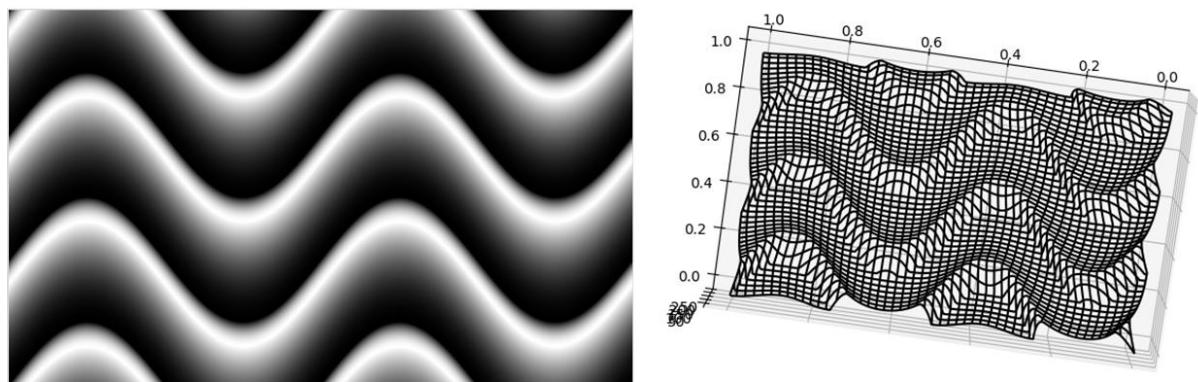


그림 15. 2차 함수 기반 상승, 하강 물결 문양

- 다각형 구조물

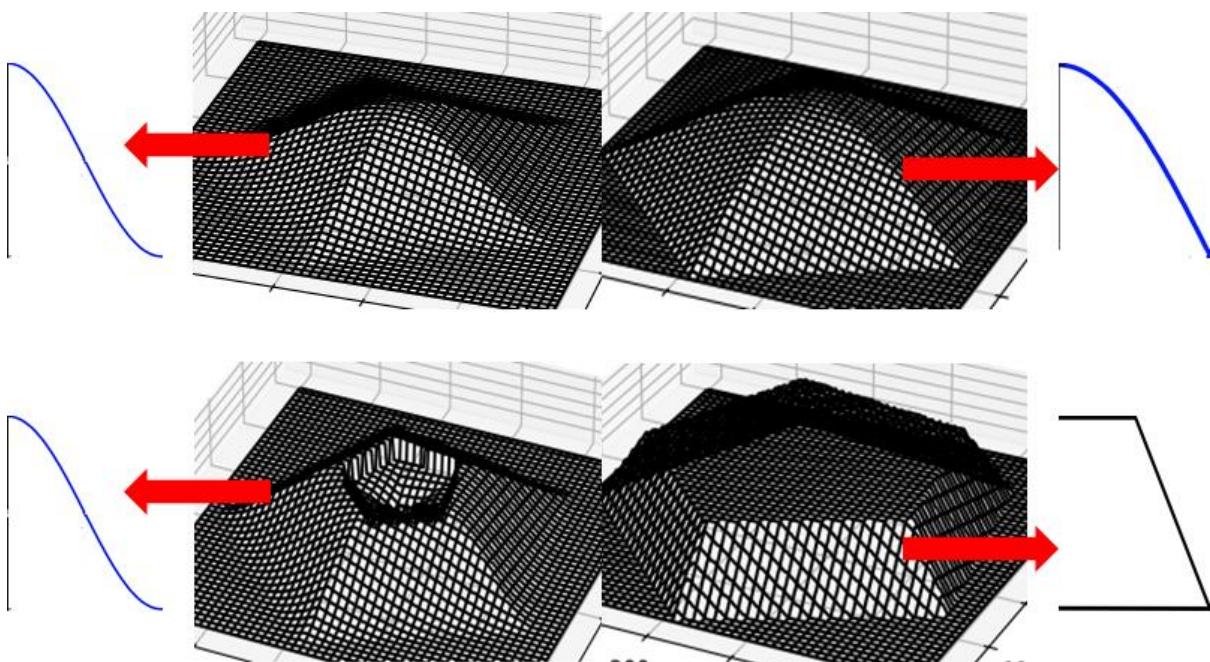


그림 16. 다각형 구조물의 예. 기본 도형: 5 각형

- 다각형 구조물 Array

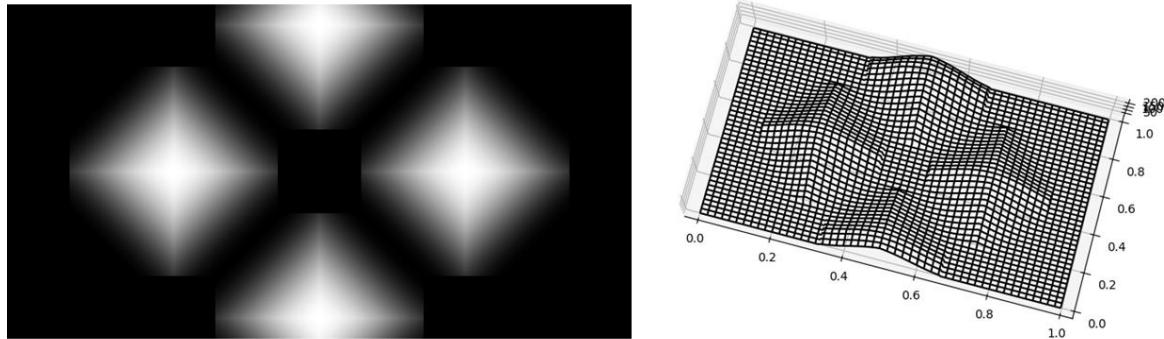


그림 17. 다각형 구조물 Array

- File 기반 생성

생성용 File 기반으로 Image를 합성하는 방법이다. 아래의 그림은 생성 용 File의 예이다 (4_overlapping_sinusoid.gen). File의 Type은 ".gen"이다.

첫 글자가 '#'인 행은 comment로써 생성에 관여하지 않는다. 이 방법은 제시된 여러가지의 단일 구조물들을 생성하여 하나로 통합함으로써 Image를 만드는 방법으로 이의 실행을 지시할 File을 생성하여 Image 합성을 하도록 한다.

```
4_overlapping_sinusoid - Windows 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
#      Size of image
#      S      Width     Height
#      S      1440      600
# Vertical_Sinusoidal, Horizontal : Sinusoidal peak shift
#      0      center_x  center_y  hor_period      ver_period      ver_max_shift
#      0      500        0          1000        200          100        80
# Radial decaying sine
#      1      center_x  center_y  period      decay      begin_phase      peak
#      1      1000       400        100        0.9        40         50
#      1      50         300        100        1          90         30
#      1      1400       0          200        0.6        90         70
#      1      600        300        150        0.6        90         80
# Desert Sand Pattern
#      2      center_x  center_y  hor_period      ver_period      ver_max_shift
#      2      500        0          1000        200          100        0.2        80
# Elliptic decaying sine
#      3      center_x  center_y  period      decay      begin_phase      peak      long_sh
#      3      500        300        150        0.6        90        150          2          45
# ctr file based generation
#      4
#      E
```

그림 18. Image 합성을 위한 생성 File 의 예

아래의 그림은 이 생성용 File 을 기반으로 생성된 Image 이다.

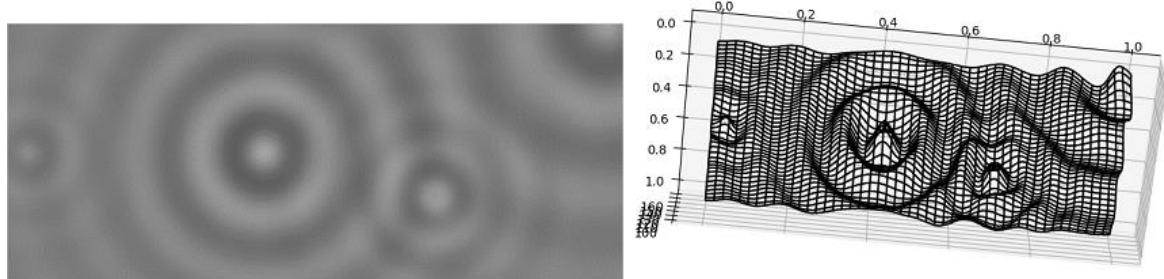


그림 19. 앞의 Image 생성 File로 생성된 Image

- **Image Clip 추가.**

현재 화면 위에 새로운 Image를 추가한다. 이를 Clip Image라고 부른다. 추가한 Clip Image는 마우스 오른쪽 Button을 누른 채 Drag하여 위치를 이동할 수 있다. Clip Image는 배경이 되는 Image에 잘 중첩이 되기 위하여 0 Level을 128로 설정한 Image를 사용하도록 한다.

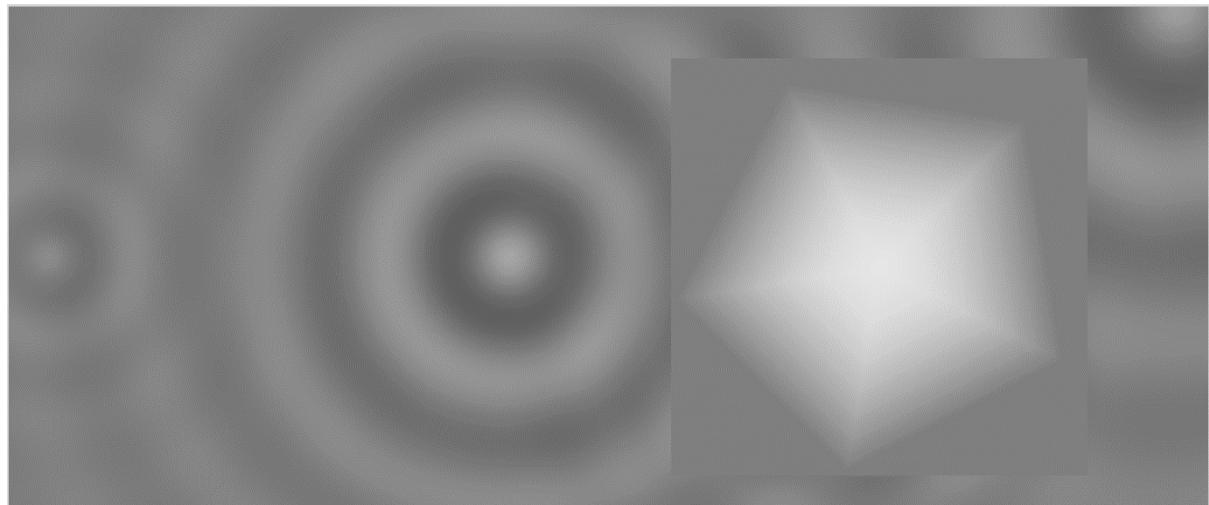


그림 20. 클립 Image를 더한 Image의 예

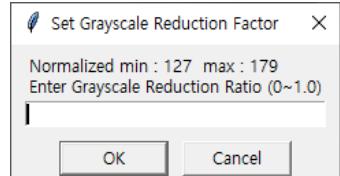
- **Clip Image 크기 변경.**

Clip Image의 크기를 변경한다.

원하는 clip을 Mouse 오른쪽 Button으로 선택한 후 이 Button 을 누르면 떠오르는



와 같은 화면에서 목표가 되는 Image 크기를 입력한다.



이후 떠오르는 와 같은 화면에서 Gray scale scaling factor를 입력한다.



- **Merge Image.**

현재의 화면에 배치된 모든 Image를 중첩하여 한 개의 Image로 저장한다. Image 저장 시 각 화소에 중첩된 Image의 값을 모두 더하여 새로운 화소값을 생성한다. 각 화소는 16 bit이며 $32768(2^{15})$ 을 Level 0로 하고 $65535(2^{16} - 1)$ 이 최대 값, 0이 최소값(-32768)에 해당하도록 Mapping 된다. Image의 합성에 필요한 각 Clip Image들은 이 같은 조건에서 적절한 합성이 이루어질 수 있도록 주의를 기울여야 한다.

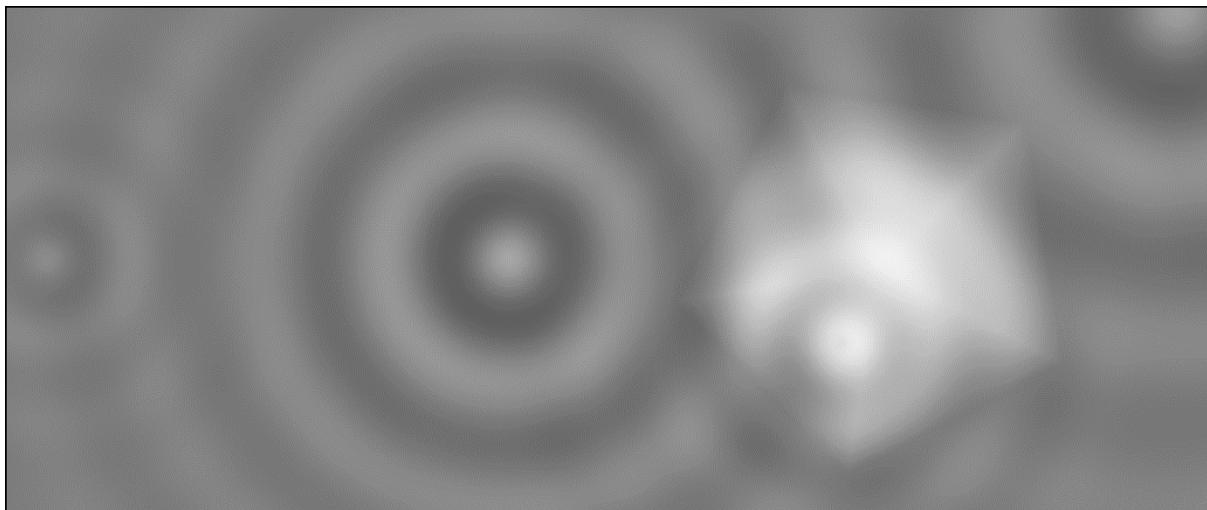


그림 22. 화면의 모든 Image를 더하여 저장한 Image



- **Slice Image.**

Image를 일정한 간격으로 수직방향으로 절단한 절단면의 Profile을 생성해낸다. 이는 Parametric Wall의 제작에 소요되는 각 Panel의 단면을 나타내는 것으로 매우 중요한 작업이다.

아래와 같은 화면에서 각각의 Panel 생성에 필요한 변수를 입력한다. 이들은 배수, 재료의 두께, Panel 사이의 간격이다. 배수는 원 Image와 만들고자 하는 목표물의 크기 비율이다. 예를 들어 2

로 한다면 원래의 Image에 대비하여 가로/세로를 각각 2배 크기로 만들기 위한 Panel들을 생성하도록 Profile을 조절한다. 높이는 각 Panel이 가질 수 있는 최대의 진폭을 뜻하는 것으로 이를 크게 하면 Panel의 폭이 넓어지게 된다. 수행이 끝나면 Slice된 각 Panel의 Profile을 저장하고 또한 이들 Panel들의 조립 후 모습을 Wire Frame을 통하여 개략적으로 보여준다. 이와 아울러 Slice된 Image를 원래 File이름 + '_sliced.png'라는 이름으로 저장하여 필요시 참조할 수 있도록 한다.

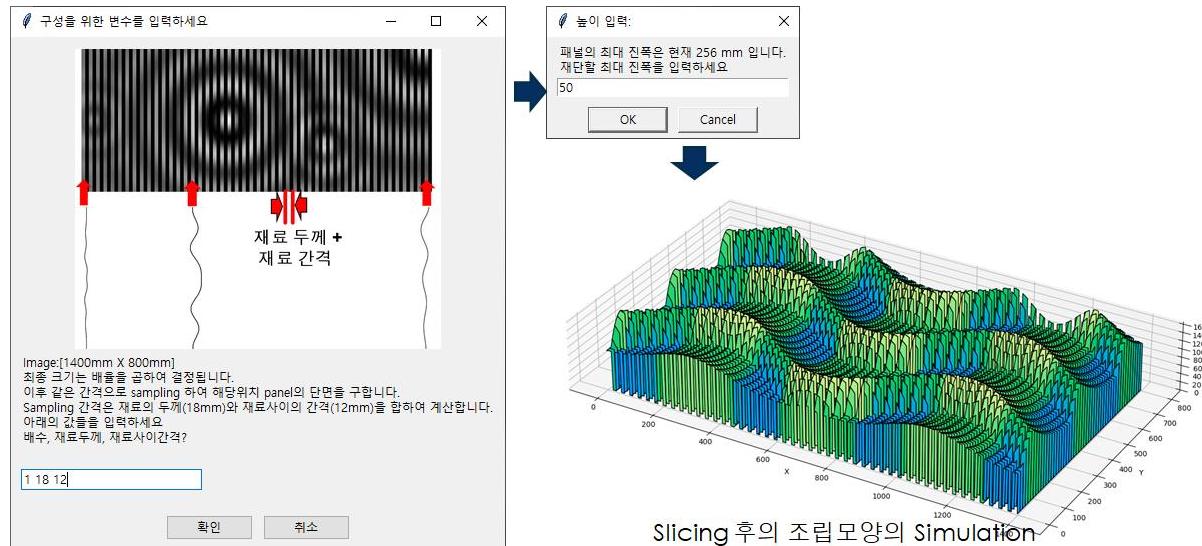


그림 23. 각 Panel 의 단면 Profile을 생성하기 위한 입력 창. 및 Slicing 후 조립된 형태의 Simulation Image.



● Panel 생성 .

Slice된 단면 Profile에 기반하여 각 Panel을 가공할 수 있는 CAD 도면을 생성한다. CAD 도면은 Woodmeister에 호환된다. 도면은 생성되는 모든 절단 부품의 개별 도면과 이들을 통합하여 가공할 수 있는 통합 도면의 2가지 종류로 생성된다. 커다란 원판을 사용하여 가공 시 후자의 도면을 사용한 가공이 편리하고 가공 시간도 절약하고 재료의 사용도 최적화 할 수 있다.

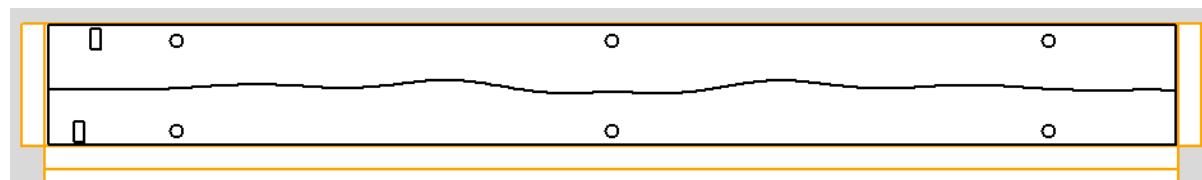


그림 24. 생성된 Panel Sample

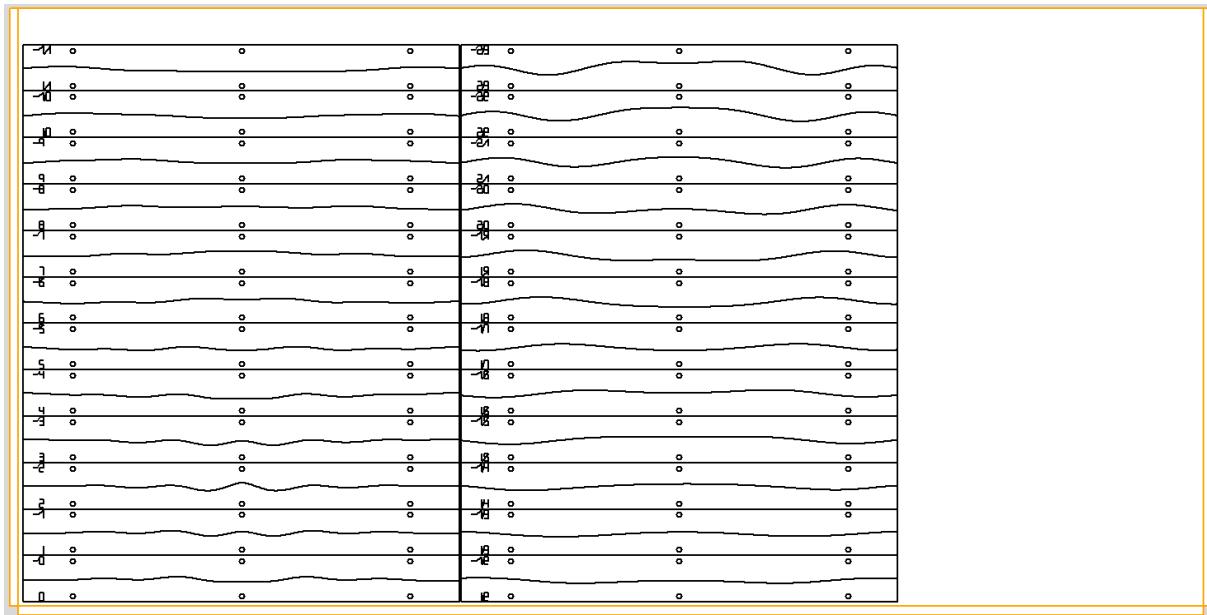


그림 25. Panel 통합 가공을 위하여 생성된 Design 도면



- 조립구조물 생성.

가공된 Panel들을 같은 간격으로 배열할 때 도움이 되는 Spacer의 형태를 생성한다. 이는 제작된 여러 개의 Panel을 모두 조립하여 원하는 Pattern을 표면에 가지는 형상을 구현하고자 할 때 필요한 것이며 다른 수단을 통한 설계가 가능하지만 이 Button을 통하여 매우 간편하게 원하는 구조물을 설계할 수 있도록 해준다. 이 Button을 누르면 아래와 같은 입력 창이 떠오른다.

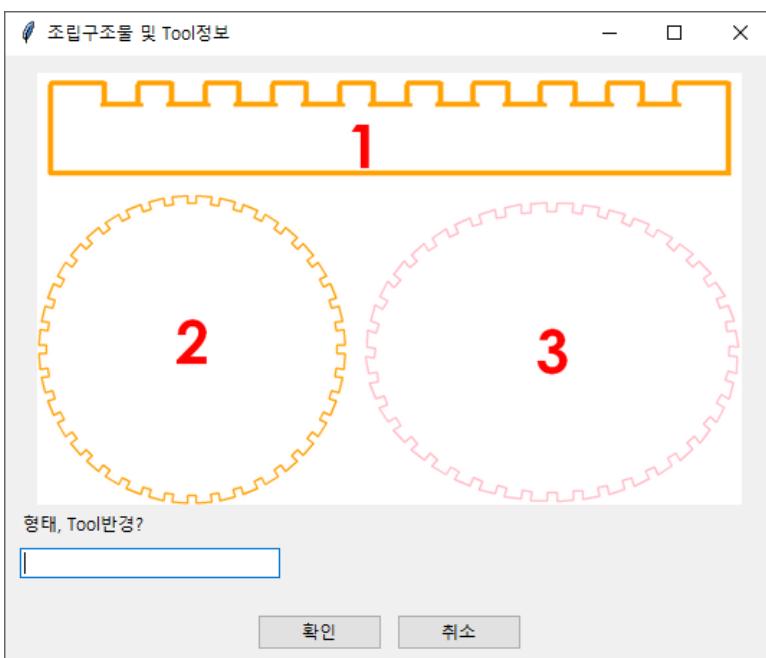


그림 26. 조립용 구조물 설계 Button

그림 26의 화면에서 형태 및 Tool 반경 값을 입력하면 선택한 형태에 따라 서로 다른 구성 용 변수를 입력할 수 있는 창이 떠오른다.

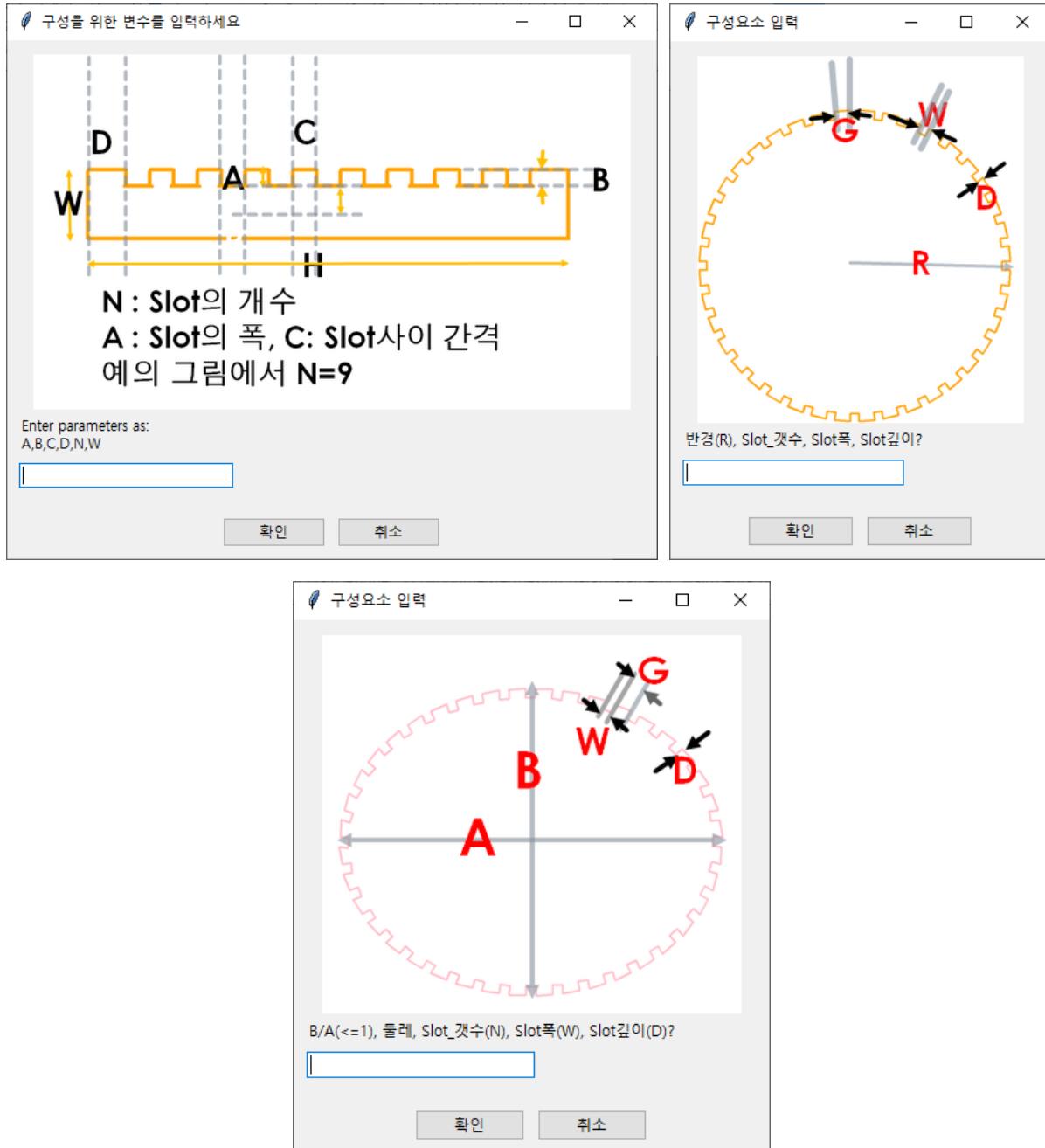


그림 27. 구조물 설계를 위한 입력 창. 직선형 구조물 (왼쪽 위), 원형 구조물 (오른쪽 위), 타원형 구조물 (아래)

그림 28 은 직선 형태 구조물 설계의 예로 제작된 패널들을 같은 간격으로 끼워 넣을 수 있는 Slot들을 가지고 있다. 모두 16개의 slot을 가지도록 설계된 예이다.

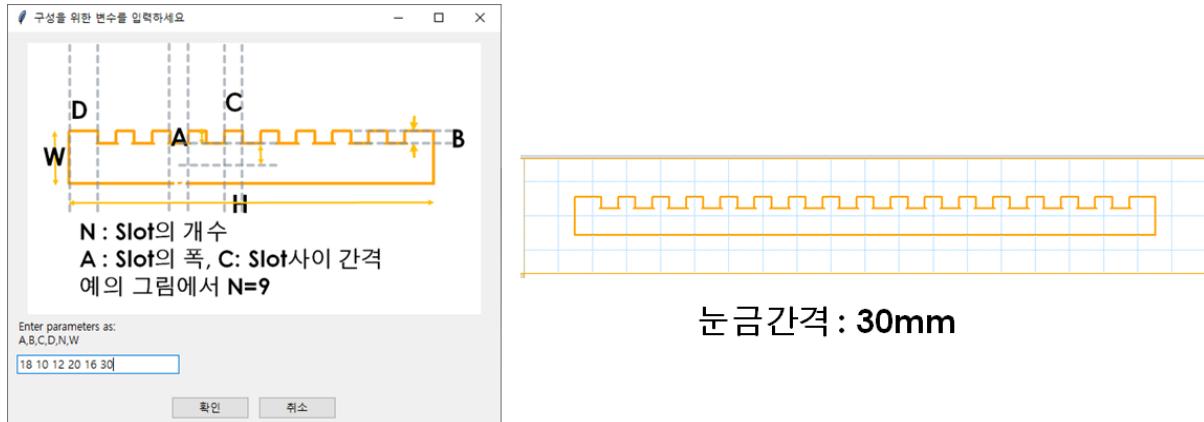


그림 28. 직선형 구조물을 선택한 경우의 입력창 (왼쪽) 및 설계된 구조물(오른쪽)의 예.

그림 29는 원형의 구조물을 설계하는 예이다. 원주 위에 같은 간격으로 48개의 slot이 배치되어 있는 예이다.

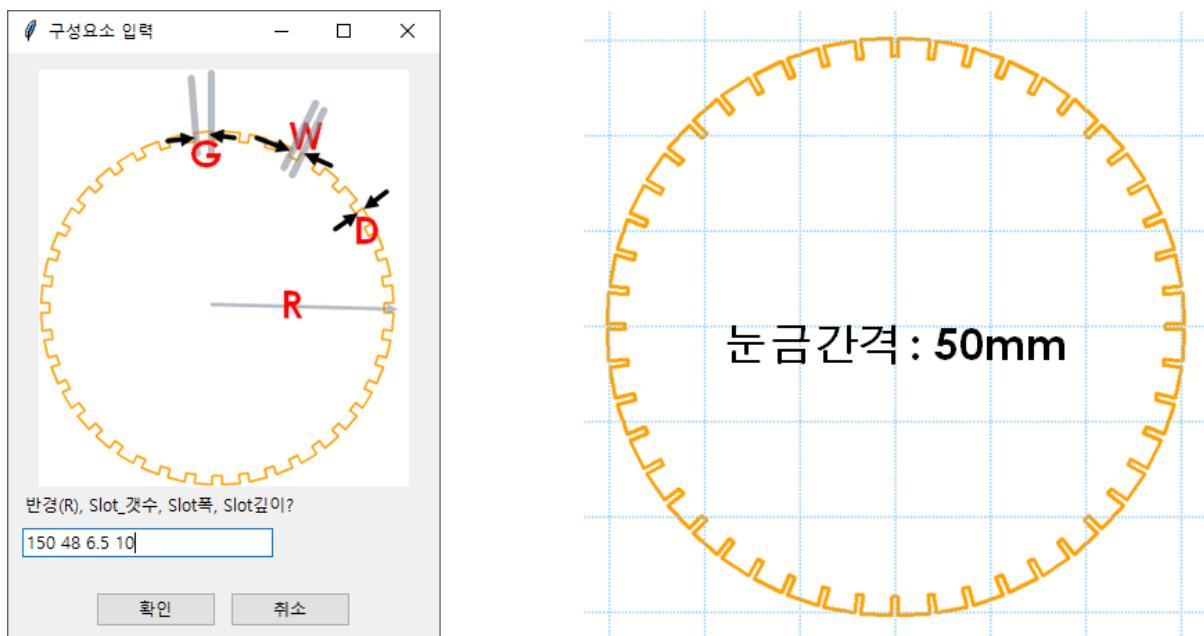


그림 29. 원형 구조물 설계를 위한 입력 창(왼쪽) 및 생성된 구조물의 모습(오른쪽)

그림 30은 타원형의 조립용 구조물을 설계하는 예로 96개의 slot을 같은 간격으로 배치한 예이다.

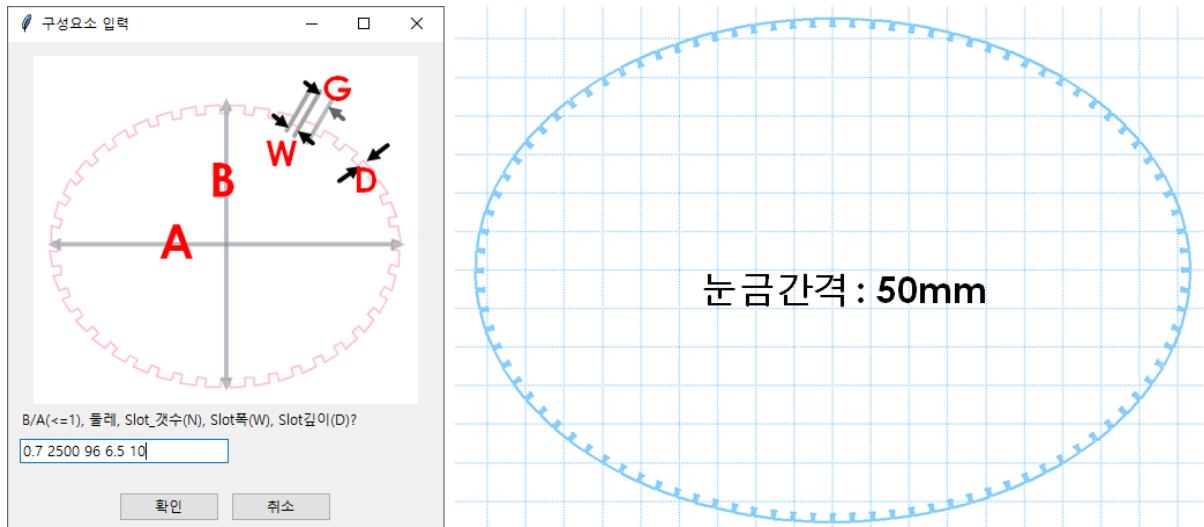


그림 30. 타원형 구조물 설계를 위한 입력 창(왼쪽) 및 구조물의 모습(오른쪽)

생성된 구조물은 Point Sequence File (.pts File)로 저장되며 Woodmeister에서 "CURVE()->*.pts 도형 가져오기"로 가져올 수 있다.

- **Dither Image 생성.**

Dither Image 모사 및 이를 위한 G-Code 생성

Dither는 Image를 표현하는 방법중의 하나이며 본 프로그램에서는 입력된 Image의 화소들을 여러개씩 둑어서 대표되는 값을 만들고 이를 CNC를 통하여 표현할 수 있도록 G-Code를 생성해준다. 그림 31은 Dither Image를 생성하기 위하여 읽어 들인 Image의 예이다.

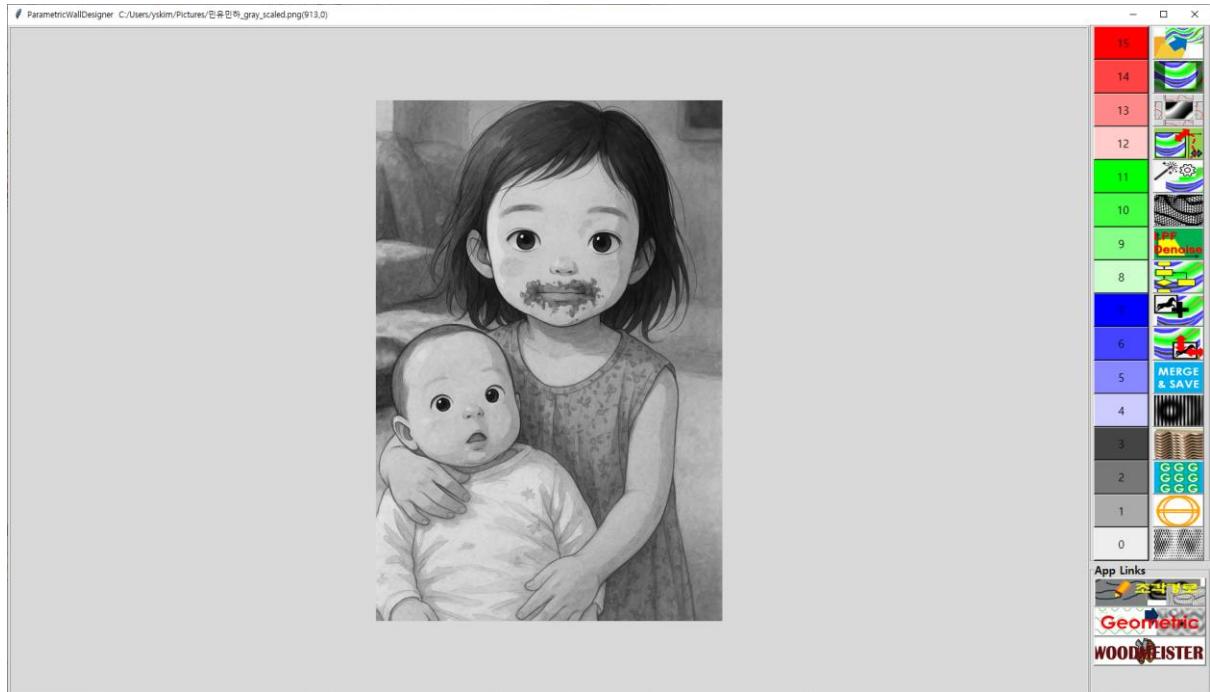


그림 31. Dither Image를 만들기 위해 입력 Image를 읽어 들인 경우

그림 32의 왼쪽 그림은 원본 Image를 Dither Image로 변환하는 과정을 보여주는 창이다. 원본 이미지(Original)와 함께 6 가지 다른 화소 패턴이 표시된다: 1. Halftone Pattern, 2. White Stripe/Black Canvas, 3. Black Stripe/White Canvas, 4. Dot Pattern, 5. White Stripe/Black Canvas, 6. Black Stripe/White Canvas. 각 패턴은 화소의 크기와 배치에 따라 다르게 표시된다. 그림 32의 오른쪽 그림은 화소 형성 방식을 설명하는 다이어그램이다. 0. Drilling 단계에서 두 개의 90도의 드릴링 헤드가 화소의 두께(D)를 만든다. 1. Drill 단계에서 두 개의 60도의 드릴링 헤드가 화소의 두께(D)를 만든다. 2. Circular Through Hole 단계에서 두 개의 원형 관통 홀이 화소의 두께(D)를 만든다.

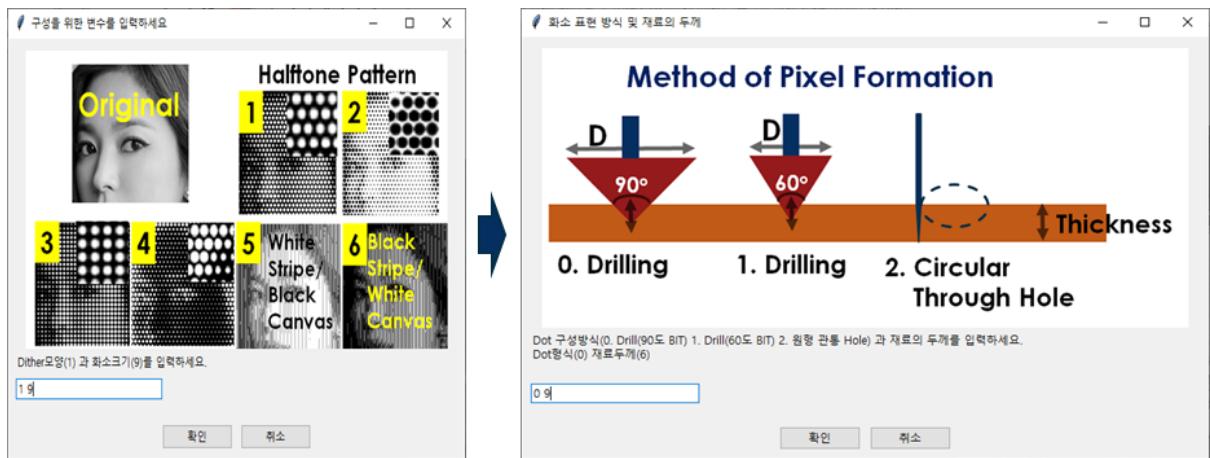


그림 32. Dither Image 생성을 위한 입력 창의 순서 및 내용. Black Canvas에 White dot로 화소를 구성함. 화소의 구성을 위하여 원 Image에서 9x9의 화소를 뮤어 1 개의 Dither화소를 만든다. Dither 화소의 크기는 화소값에 비례한다.

그림 33의 왼쪽 그림은 81개(9x9)의 화소를 한 개의 새로운 화소로 표현한 경우로 화소의 개수는 원 화소의 대략 1/81로 줄어든다.

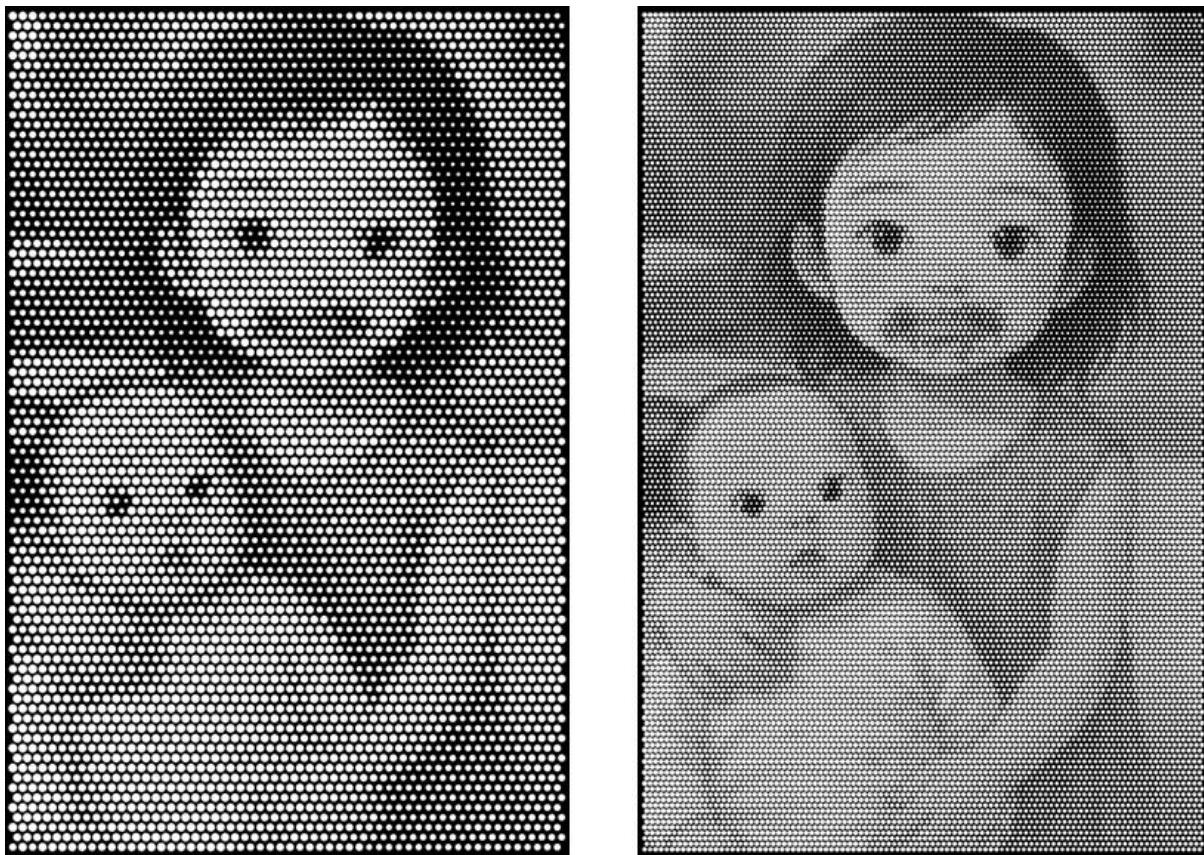


그림 33. 그림 32와 같은 입력을 한 경우 Simulation Image. 화소의 구성 : 9x9(왼쪽). 화소를 5x5로 구성한 경우의 Simulation Image(오른쪽)

그림 33의 오른쪽 그림은 5x5의 화소로 Dither Image를 새로 만든 경우로 9x9에 비하여 상당히 세밀한 표현을 할 수 있다.

해당되는 Dither Image를 CNC로 생성하기 위하여서는 다음의 입력창에서 크기 조절을 위한 변수를 입력하고 G-Code를 저장할 File을 지정하여 .NC File을 만든다.

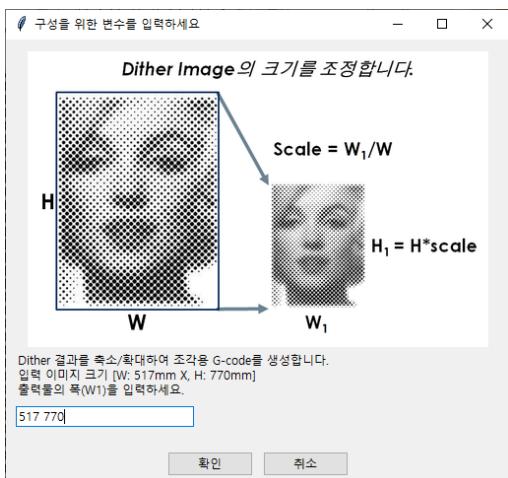


그림 34. G-Code를 생성하기 위한 입력 창

생성된 G-code를 CNC에 전달하여 가공하면 Simulation으로 생성된 Image에 가까운 Dither Image를 만들어 낼 수 있다.

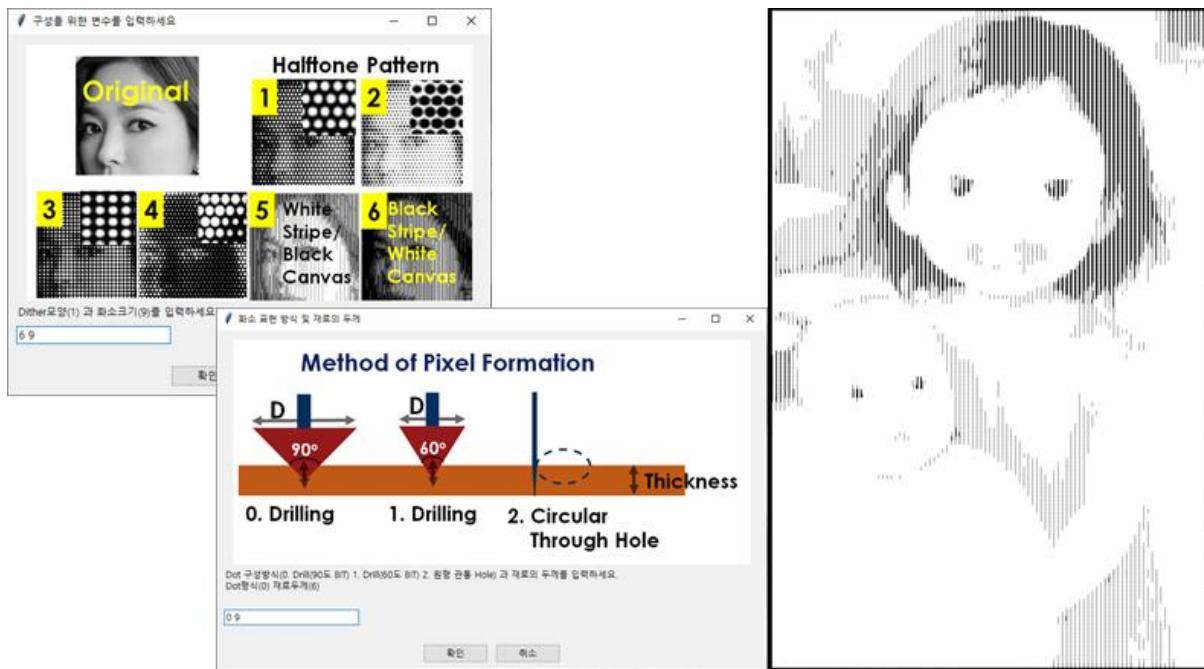


그림 35. Stripe 형태로 Dither Image를 만드는 예. 화소의 폭 표현의 한계로 Simulation Image(오른쪽)가 원본을 충실히 보여주지는 않으나. 실제 가공은 이보다 매우 충실히 표현이 된다.

2.4. Application Program launcher

Image 처리를 위한 main Program과 연동하여 이에 소요되는 부가적인 작업을 하기 위한 별도의 Application Program을 수행하기 위한 Launcher Button이다.



PathMaker를 시작한다. PathMaker는 자유 곡선에 기반한 조각 또는 Image생성을 위한 경로 Data를 생성하는 Application Program이다.



GeoLaD를 시작한다. GeoLaD는 여러 종류의 기하학적인 Pattern의 배치 및 이를 통한 Image 생성을 하는 Application Program이다.



Woodmeister Program을 시작한다.

3. 조각 경로 작성 Tool (PathMaker)



PathMaker는 자유 곡선에 기반한 조각 또는 Image생성을 위한 경로 Data를 생성하고 이로부터 Image를 생성하거나 조각을 위한 CNC 경로의 생성 등을 할 수 있도록 제작된 전용 Application Program이다. 예를 들어 아래의 그림과 같은 자유곡선들로 이루어진 Pattern을 설계할 수 있다.

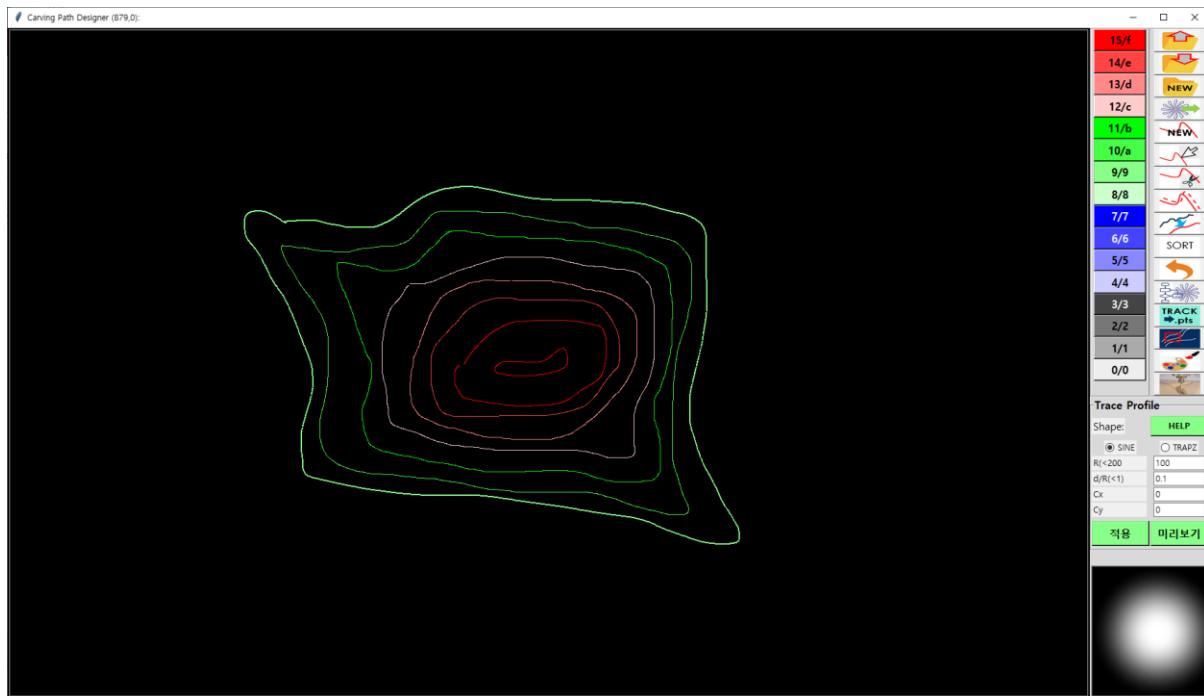


그림 36. PathMaker의 화면 Layout

3.1. 화면 Layout

그림 36과 같이 왼쪽 부분은 그림을 그리기 위한 Canvas 영역으로 되어 있으며 오른쪽에 Level을 설정하는 Color Button과 여러 개의 동작을 지시하는 Button들이 있다. 왼쪽의 Canvas 영역에는 여러 개의 선들이 서로 다른 색상으로 그려져 있는 예이다. 서로 다른 색상은 서로 다른 Level을 뜻하며 색상에 따른 Level 값들이 Button 내부에 표기되어 있다.

PathMaker에서 생성/저장/관리하는 File은 type이 ".trk"로 지정되는 Track File이다. 한 개의 Track File은 한 개 이상의 Trace로 구성되어 있으며 위의 그림에서 서로 다른 색으로 표기된 선들이 각각 Trace이다. 위의 그림은 8개의 Trace를 가지는 Track File을 표기한 것이다.

3.2. Button

3.2.1. Level

숫자 표기된 Button들은 Trace의 Level을 지정하는 Button이다. Level은 Trace를 따라 Image를 생성하거나 조각을 하는 경우 이의 높이 또는 깊이를 나타내는 값이다. Trace를 새로 시작하거나 기존 Trace의 Level을 변경하고자 할 때 유용하다. Level은 16개까지 지정할 수 있다. Trace를 새로 시작하거나 선택한 Trace의 Level을 변경하고자 할 때 해당하는 Level Button을 누르면 Level 값이 변경된다.

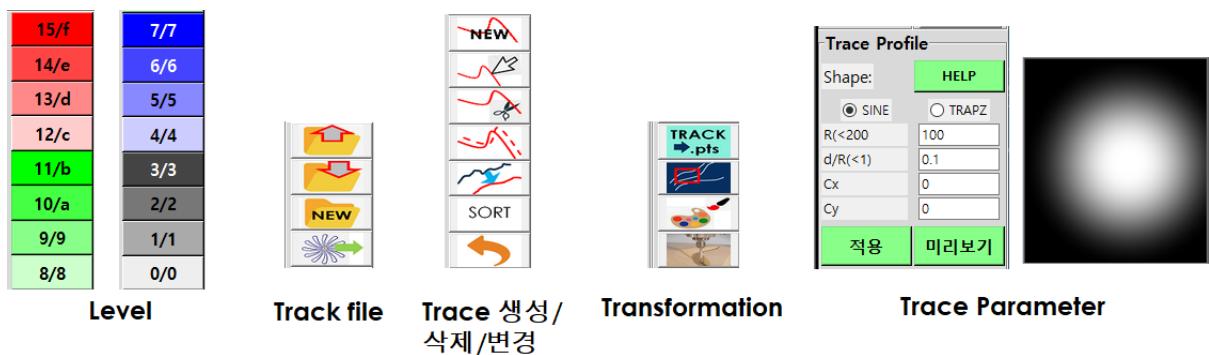


그림 37. PathMaker의 Button 및 Parameter

3.2.2. Track File 관리



- **Track File Open.**

Track File을 읽어 온다. 이 Button을 누르면 배경이 되는 그림을 읽어올지 묻는 창이 떠오른다. 배경 그림을 가져오는 경우 이에 기반하여 Trace들을 그리는데 도움이 될 수 있다. 배경 그림의 선택 이후 Track File을 읽어서 이를 창 위에 표기 할 수 있도록 유도하여 준다.



- **Track File 저장.**

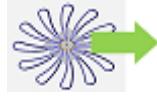
화면의 내용을 Track File로 저장한다. 배경 그림을 제외한 Trace들만 저장한다.

- 새로운 Track File 만들기.



새로운 Track File 작성은 시작한다. 이를 위하여 디자인 화면에 표기된 모든 내용을 지우고 다시 시작하도록 한다. 시작하면서 배경화면을 가져올지 묻는 창이 먼저 떠오른다.

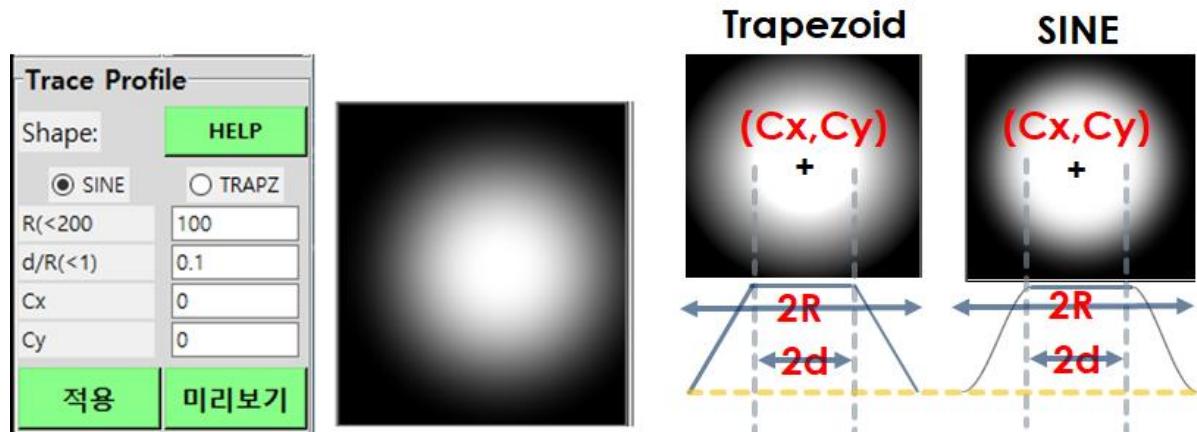
- G-Code File로부터 Track File 만들기



G-Code File을 읽어 들여 Track File로 변환하여 화면에 표기한다. 다른 수단으로 만든 G-Code File을 이용하여 Image Pattern을 만드는데 활용할 수 있다.

3.2.3. Trace의 생성/삭제/변경

Trace를 생성/삭제/변경하는 여러가지 일들을 할 수 있는 기능이 부여된 Button들이다. Trace는 선과 Level로 표현되는 도형적인 특성 이외에 각각의 Trace가 Image 생성시 어떻게 활용되는지를 지정하는 Parameter를 가질 수 있다. 이를 요약하여 표현하는 것이 Trace Parameter 창이다.



Trace Parameter(왼쪽) 및 이에 따른 Tool Shape (오른쪽)

그림 38. Trace Profile을 표기하는 Parameter 창과 Profile shape

Trace Profile Parameters:

Image 생성을 할 때 각각의 Trace를 따라 Tool이 움직이며 Tool의 형태 및 궤적에 따라 생성되는 Image의 모양을 만들어 내기 위한 것이다. 즉 이 Tool의 모양에 따라 만들어지는 단면의 형태가 달라질 수 있도록 하는 것이다. 이는 Image의 생성에 매우 유용하다. 그림 39는 Trace Parameter의 변경에 따른 Tool의 모형에 대한 여러가지의 예이다. 실제로 가공을 할 때 이와 같은 단면을 가지는 Tool을 사용하기는 어려울 것이다. 여기서의 가공이란 가상적인 것으로 각 Trace가 만들어

내는 Image의 형태를 만들기 위한 것임을 알아두기 바란다. 실제로 고속으로 회전하는 Tool을 사용하는 CNC 가공에 center가 (0,0)이 아닌 Tool을 사용할 수는 없다. 다만 이 같은 Tool이 가상의 Image 형성에는 매우 유용하게 사용될 수 있는 것이다.

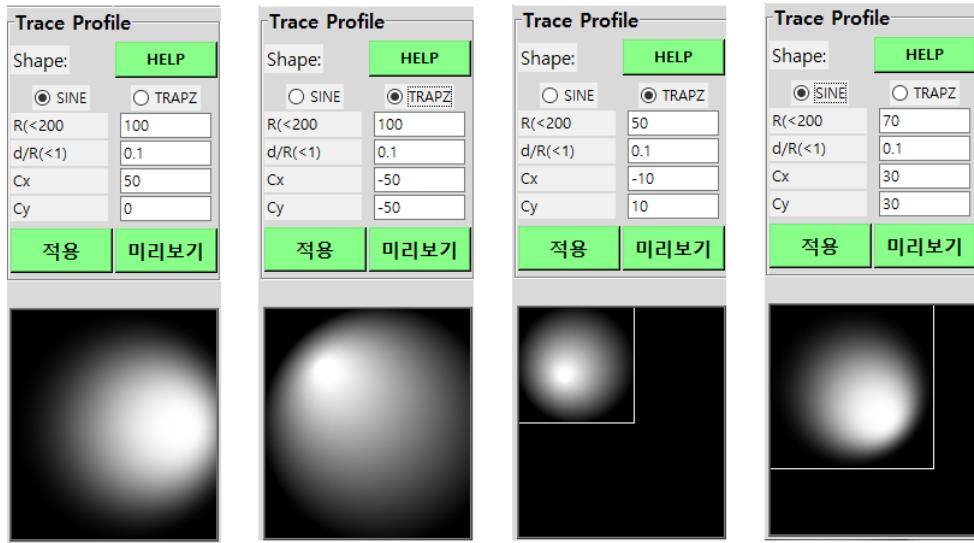


그림 39. Trace Parameter 변경에 따른 가공 부위의 형상

그림 40은 같은 모양의 Trace 3 개를 가지며 각각의 Trace Parameter가 서로 다른 경우에 생성된 Image의 모양과 각 Trace의 Profile을 같이 표시하여 놓은 그림이다.

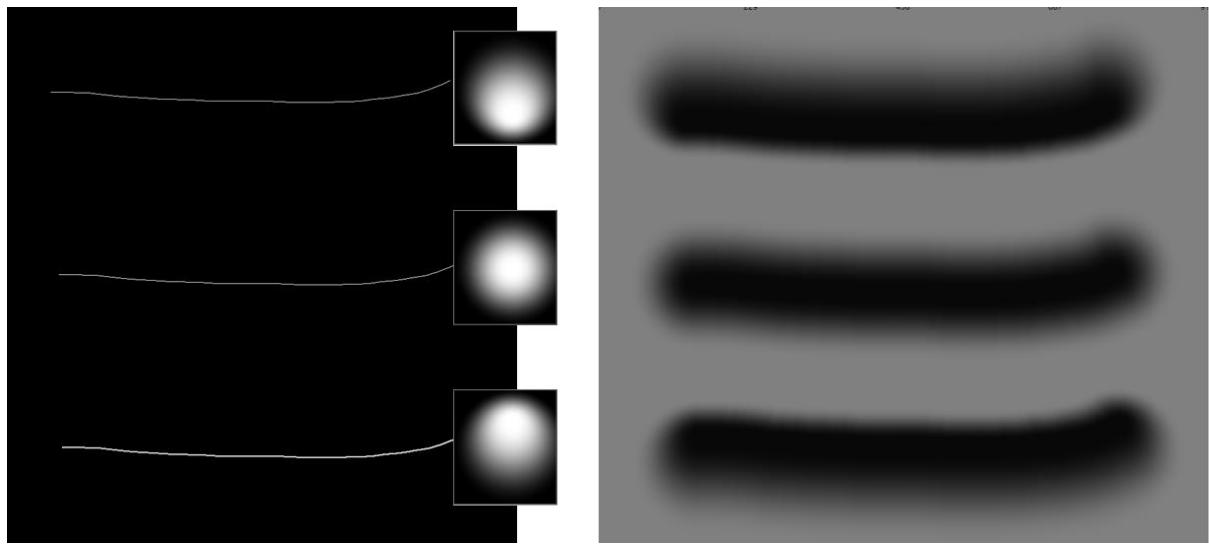


그림 40. Trace Parameter의 변화에 따른 생성 Image의 변화를 보여주는 예.

- 새로운 Trace의 시작.

NEW

새로운 Trace를 시작하기 위하여 누른다. 이후 Canvas에서 마우스 왼쪽Button을 누르고 drag하여 Trace를 그리고 끝내기 위하여 Button을 Release한다. Trace의 Level을 지정하기 위하여 Level Button을 누르면 해당 Level로 Trace의 색이 바뀐다. 이후 Trace의 Profile 항목 내의 각 값을 변경한 후 **적용** Button을 누르면 해당 Trace의 Profile로 적용이 된다. **미리보기** Button을 누르면 해당 Profile의 모양을 아래의 작은 창에 보여준다.



이 Button은 Canvas 위에 있는 Trace를 선택하기 전에 누른다. 이후 해당 Trace를 왼쪽 마우스 Button으로 click하여 선택한다. 선택된 Trace는 왼쪽 Button을 누른 상태로 이동할 수 있으며 삭제 Button ()으로 삭제하거나 복제 Button ()으로 복제하거나 Smooth Button ()으로 경로를 매끄럽게 할 수 있다.



선택된 Trace를 삭제한다.



선택된 Trace를 복제한다. 복제된 Trace가 추가되면 이것이 선택된 상태가 된다.



Trace 생성 시 포함된 거친 움직임 부위를 부드러운 형태로 바꾸어 해준다.

SORT

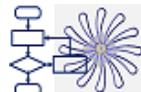


가공 시간을 최적화할 수 있도록 Trace의 순서를 재정렬하여 준다.



- 되돌리기.

Trace를 지우거나 복제한 행위를 없던 것으로 되돌린다.



- 기하학적 Trace 생성.

사전 정의된 알고리즘으로 Track을 생성한다. 이는 개발용으로 사용하는 곳이며 일반 사용자들의 사용은 제한한다.



- Track -> .pts 변환.

Track File로부터 .pts File을 생성한다. pts File은 Woodmeister를 이용한 Design 등에서 다양한 용도로 사용할 수 있다.



- 처리 영역 지정.

조각 또는 Image 생성을 위한 ROI (Region of Interest)를 지정한다. 이 Button을 누르면 아래와 같이 관련 항목들을 입력할 수 있는 창이 떠오른다. 여기에 생성 영역에 대한 정보를 입력하고 “확인” Button을 누른다.

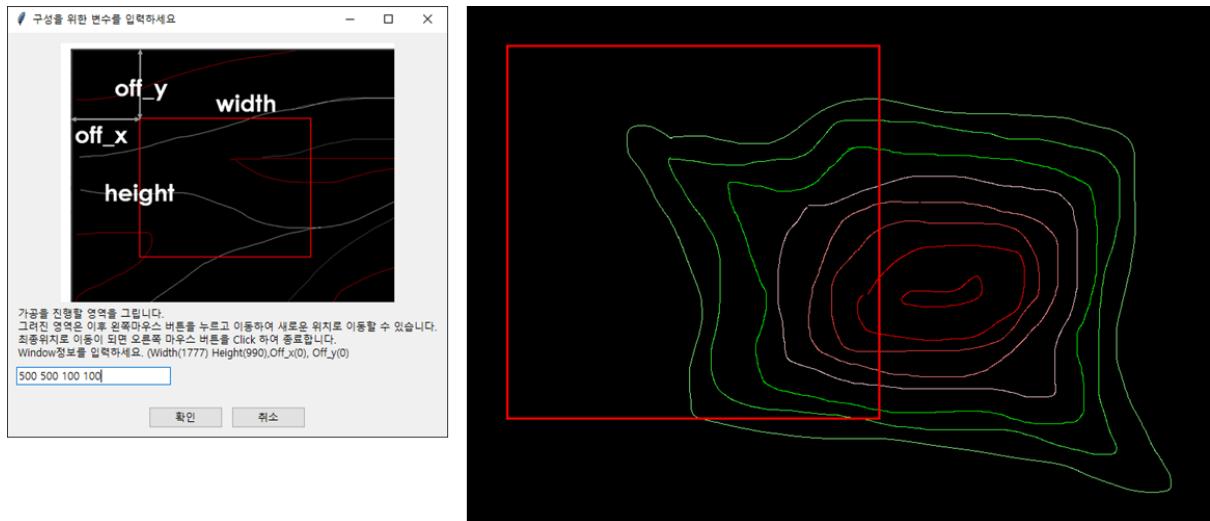
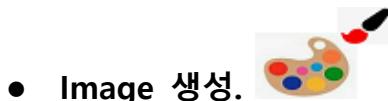


그림 41. 처리 영역 지정을 위한 입력 창 (왼쪽) 및 확인 Button을 누른 후의 화면 (오른쪽). 입력된 내용에 해당하는 영역이 붉은색 사각형으로 표기된다. 이 사각형은 마우스 왼쪽 Button을 누르고 Drag 후 Release함으로써 이동할 수 있다. 이후 이미지 생성 또는 조각 경로 생성 Button Click 시 해당 영역의 Image를 생성하거나 조각하기 위한 Code를 생성한다.



화면에 표기된 Trace들과 ROI를 기반으로 Image를 생성한다. 이 과정을 그림 42에 요약하였다.

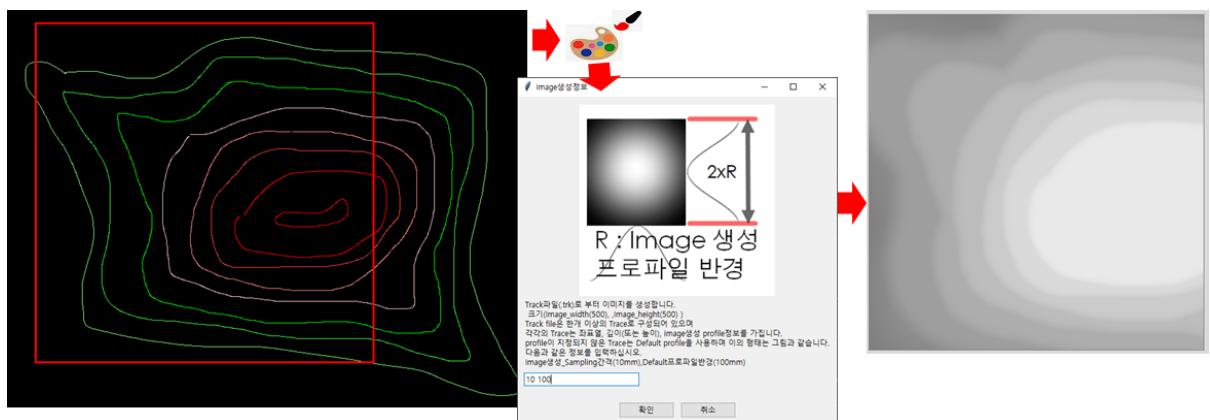


그림 42. ROI가 설정된 후 Image 생성Button을 누르고 해당 영역의 가공을 위한 변수를 입력하면 해당 영역으로부터 생성된 Simulation Image가 생성된다. 생성된 Image는 약간의 처리를 거쳐 Parametric Wall의 제작 등에 활용할 수 있다.

Image 생성시의 Level mapping : 각 Trace에 할당된 Level 값은 해당 Trace로 Image를 생성할 때 Pixel 값의 최대치(쌓아올리는 경우) 또는 최소치(파내려가는 경우)를 지정하는 것으로 그림 43과 같이 mapping 된다.

Level 8은 높이 0로 변환되며 Level 0 이 가장 깊은 Level로, Level 15가 가장 높은 값으로 mapping 된다. 각각의 Level 사이는 15씩 차이가 나도록 하고 있다. 즉 Level이 9 이상인 경우 둑을 만들고 Level 이 7이하인 경우 도랑을 파내는 것과 유사하게 Image를 생성한다. 둑을 만드는 경우 Level 값이 클수록 높은 둑이 되며 도랑을 파내는 경우 Level 값이 작을수록 깊은 도랑으로 만들어진다.

그림 44는 Track File로부터 생성하는 또다른 Image의 예이다. 각각의 Trace들에 서로 다른 Profile 을 적용하여 도랑과 둑을 형성하는 모습을 보여주고 있다.

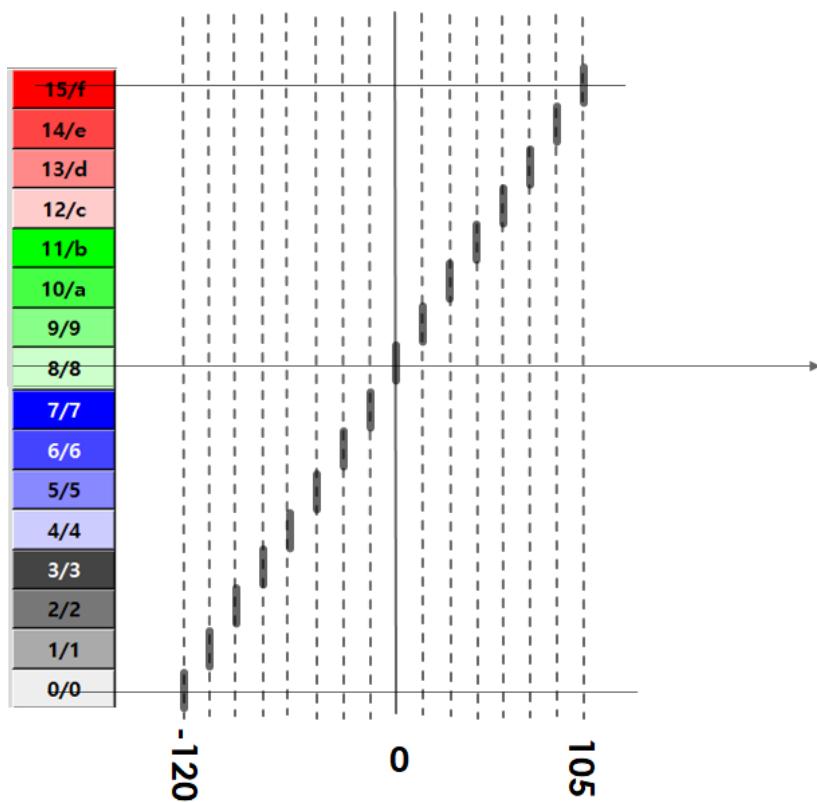


그림 43. Image 생성시 Level 과 Pixel값의 관계

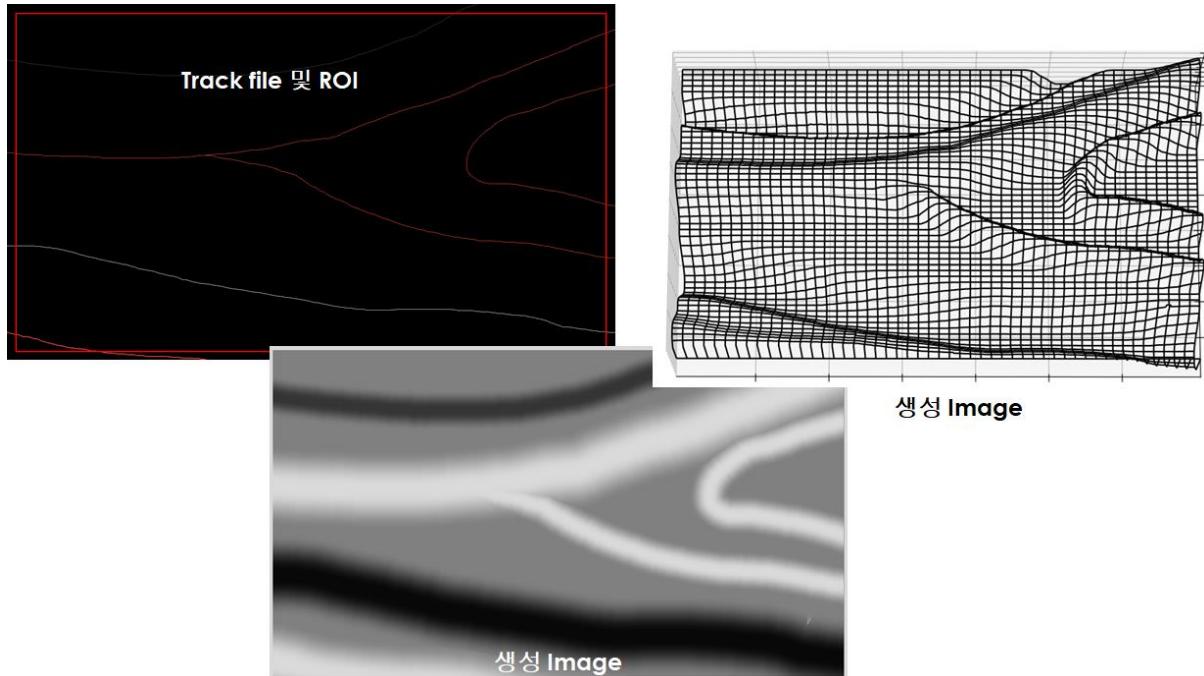


그림 44. Track File에서 생성한 Image의 예



Track File에 기반한 조각 경로를 생성하여 G-Code File로 저장한다. 여기에서 조각의 의미는 CNC를 사용하여 가공을 하는 것을 의미한다. 조각 경로를 만드는 것과 아울러 이에 의해 조각되는 형상의 Simulation Image를 생성하여 보여준다. 이에 반영되는 ROI는 앞에서 기술한 것과 같은 방식으로 정하여 진다.

CNC를 이용한 가공은 재료를 깎아내는 가공이다. 따라서 이때 Trace의 Level값은 앞의 Image 생성과는 조금 다른 의미를 가진다. Level 값이 15인 경우 최소한의 깊이로 Level 값이 0인 경우 최대한의 깊이로 가공을하도록 한다. 아래의 그림은 CNC 조각을 위하여 생성하는 G-Code File과 이에 의하여 생성되는 Image를 simulation 한 것이다. 조각 시 Level값 1의 차이는 1mm에 해당하여 깊이 1mm에서 15mm 까지의 깊이로 제어할 수 있도록 하였다.

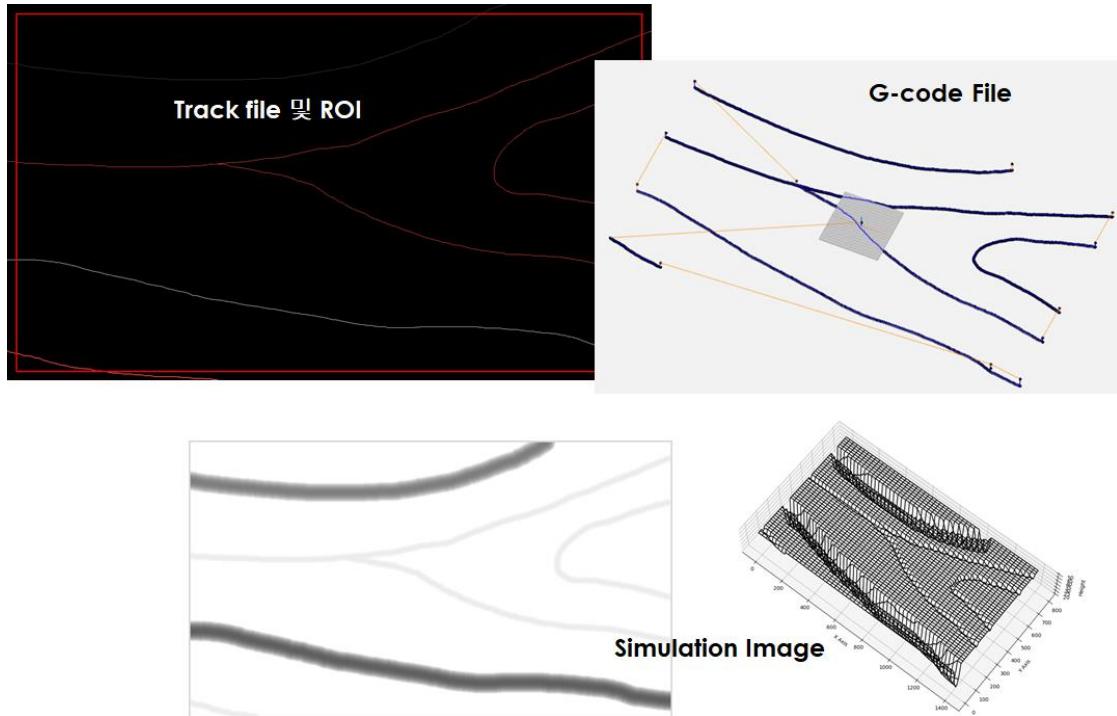


그림 45. Track File로 CNC 조각을 하는 경우 생성되는 G-Code와 simulation Image.

CNC로 조각을 하는 경우에는 사용되는 Tool의 형상과 Tool을 가공물로 내려 보내거나 들어올릴 때의 경로 설정, 가공 배율과 같은 정보를 필요로 한다. 이들 정보는 그림 46과 같은 창에서 입력한다.

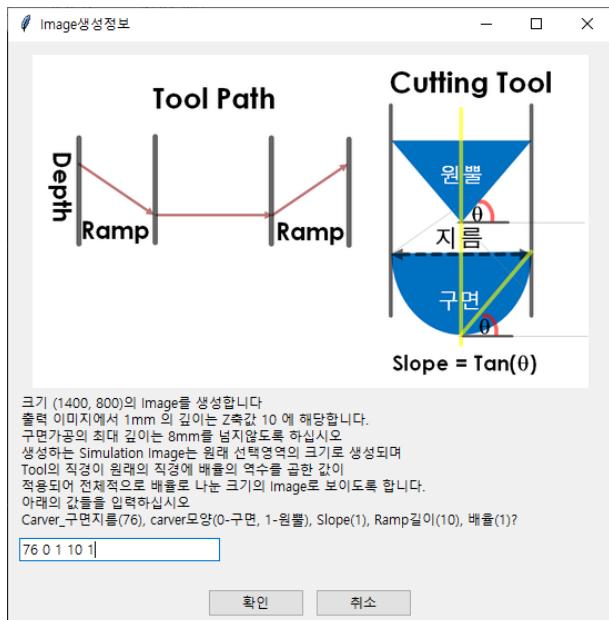


그림 46. CNC 가공을 위한 Tool 정보 및 Path 관련 정보 입력 창. 해당 예에서는 Tool 의 직경 76mm인 구면, $\theta = 45^\circ$, Ramp 길이 10mm, 1:1 배율 가공에 해당하도록 입력함.

그림 45는 그림 46에서 보여준 예로 가공을 하는 경우의 결과물이다.

4. Geometric Layout Designer (GeoLaD)



GeoLaD는 물결 등 자연 현상 또는 구조물을 모사할 수 있는 기하학적 모델을 자유로이 배치하고 배치된 모델들로부터 이미지를 생성하여 Parametric Wall 등의 제작에 활용할 수 있도록 하는 GUI 기반의 Layout tool이다.

그림 47은 GeoLaD 수행 직후의 화면의 모습이다.

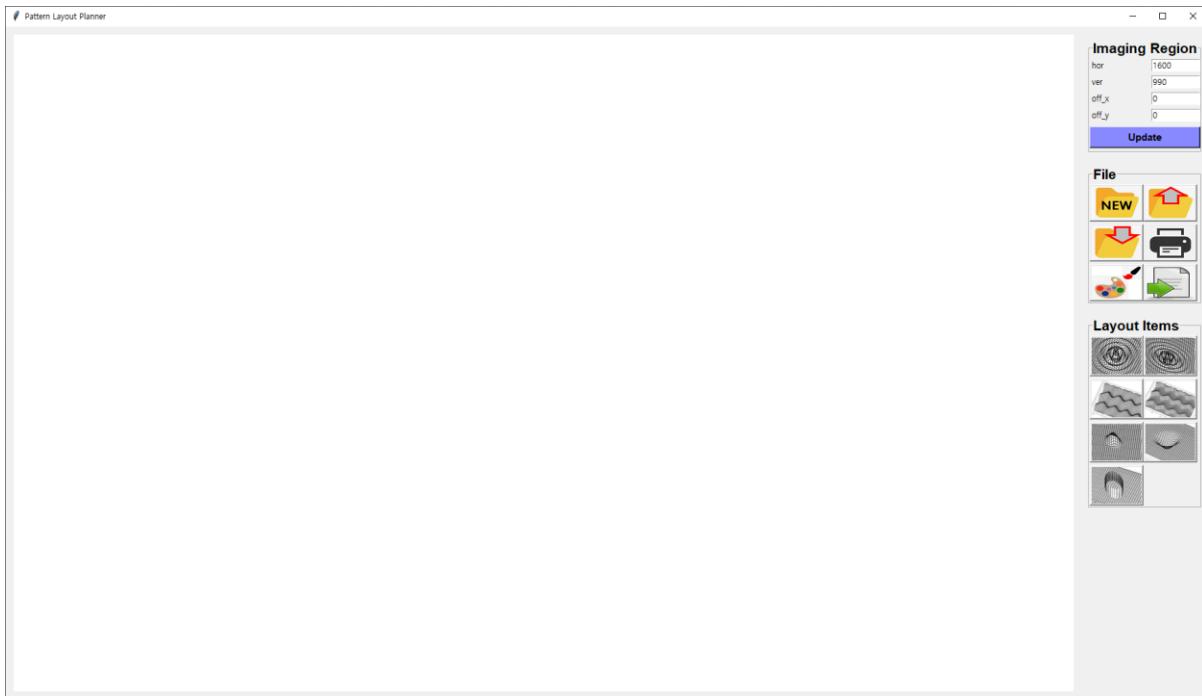


그림 47. GeoLaD의 실행 초기화면. 왼쪽의 큰 Design Window와 오른쪽의 Control Button 및 Parameter 영역으로 나뉘어 있다.

4.1. 화면 Layout

화면의 왼쪽 부분은 디자인 창을 보여주며 오른쪽 부분은 Control 영역으로 여러 기능들의 실행을 위한 Button과 디자인 요소를 선택하고 이들에 대한 Parameter 입력을 할 수 있는 Field들을 가지고 있다. 오른쪽의 Control 영역에 배치되어 있는 요소들은 아래의 그림과 같다.



그림 48. Control 영역의 Button과 Parameter 입력창.

4.2. Button 및 Parameter

4.2.1. Imaging Region Parameter와 Button:

Image 생성에 반영되는 영역(ROI)을 지정하는 값을 입력한다. 아래의 그림과 같이 입력하고 **Update** Button을 누르면 때 갈색 Box로 Image 생성을 위한 영역이 표시된다. 이후 Image 생성 작업을 하는 경우 해당 영역 내에서 작업을 진행한다. 그림은 1400 X 800의 크기를 가지는 영역을 지정하고 있으며 이의 왼쪽 상단 좌표는 (100, 100) 라는 것을 표현한다.

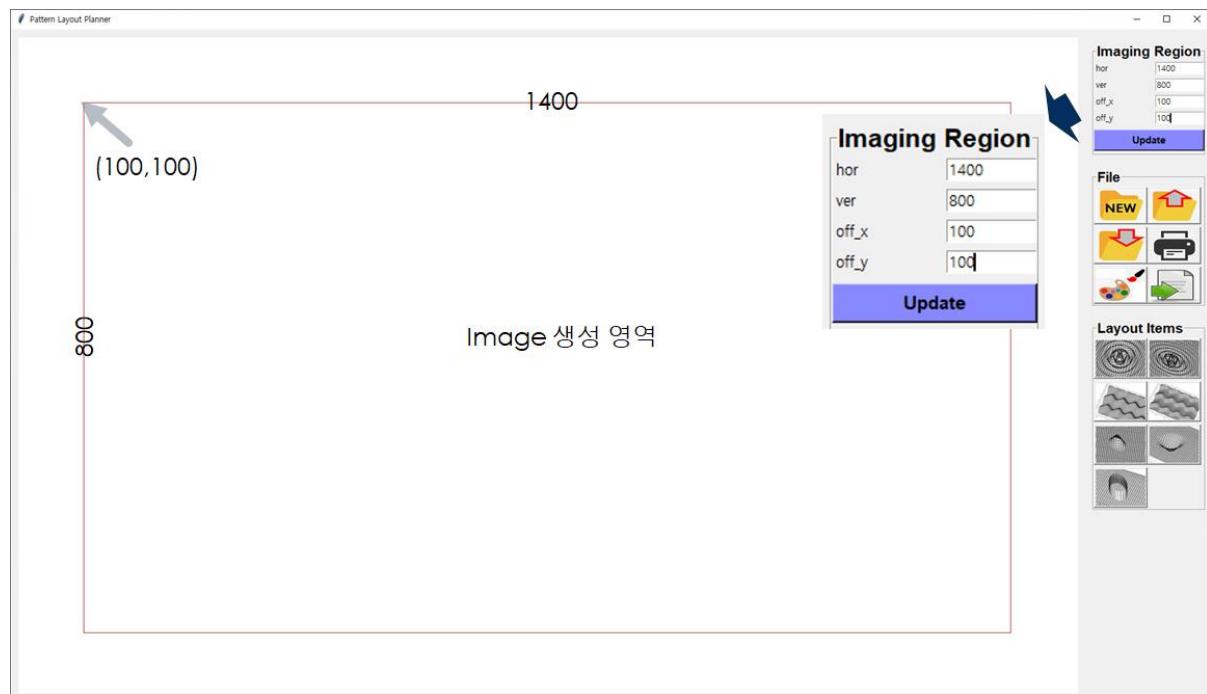


그림 49. Image 생성 영역의 지정. Imaging Region의 각 항목을 필요한 값으로 채운 후 Update Button을 누르면 갈색 Box로 영역이 표기된다.

4.2.2. File 관리

여러 종류의 File 관리 관련 Button들을 가지고 있다.

- 새로운 Layout File 작성하기.



Layout을 새로이 작성한다. 왼쪽 Design 화면의 모든 내용이 삭제되어 새로운 Layout을 작성할 수 있도록 한다.

- Layout File 가져오기.



Layout을 저장한 File을 읽어온다. Layout 저장 File의 Type은 ".lay" 이다.

- Layout File로 저장하기.



Design Window에 배치된 내용을 Layout File로 저장한다.

- Layout 상태를 Text로 보여주기.



현재의 Design을 Text로 화면에 표시해 준다. 아래의 그림을 참조하기 바란다. 디자인은 2개의 요소를 포함하고 있으며 이들의 모양 및 배치를 Text로 표현하여 준다. 이 기능은 참조를 위한 것으로 일반적인 사용은 거의 필요 없는 기능이다.

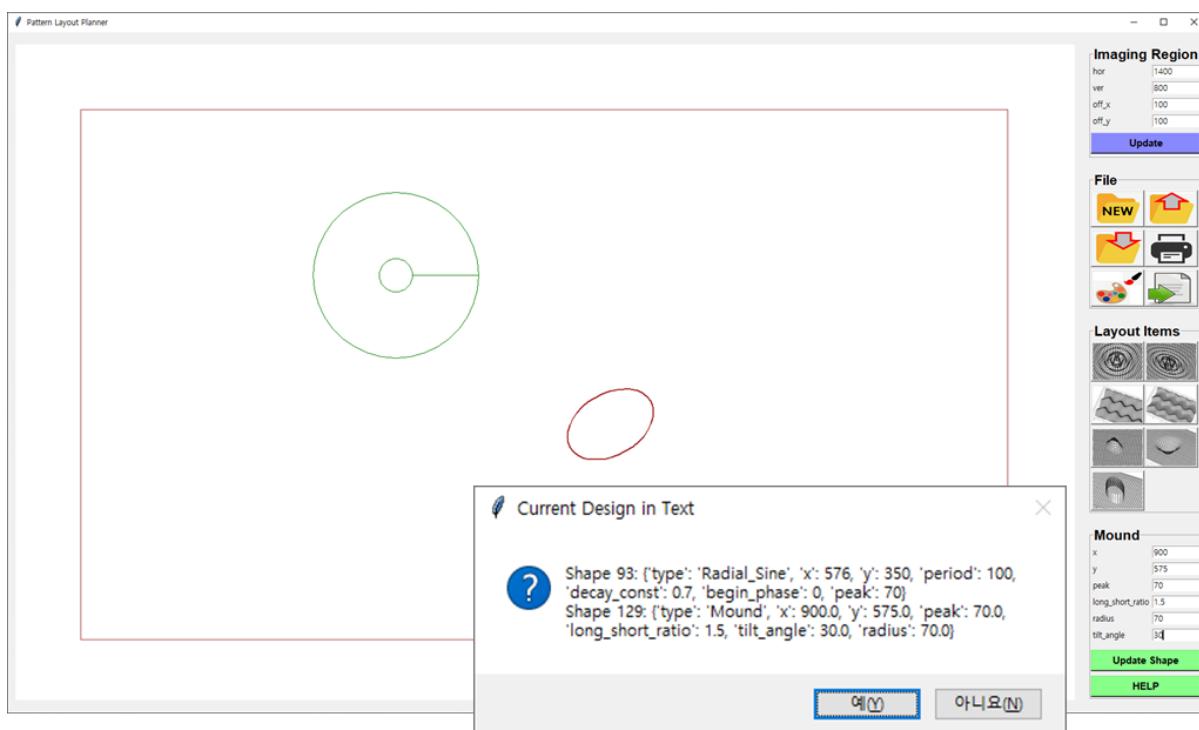


그림 50. Design화면과 Text로 표기된 Design

- **Image 생성.**



Layout 정보를 바탕으로 Image를 생성하여 저장한다. 생성 영역에 둘러싸인 부분의 Image가 생성되며 ".png" File로 저장된다. 그림 51은 Layout 과 이것으로부터 생성된 Image를 보여준다.



그림 51. Layout (왼쪽)과 생성된 Image(오른쪽)

- **Track File ("*.trk")로 저장.**



Layout을 기반으로 Track File을 생성하여 저장한다. Track File을 구성하는 Trace들은 파동을 나타내는 Item에서는 Local peak인 마루와 일치되게 생성되며 단순한 구조의 Layout Item의 경우 외곽선을 따라 생성된다. 그림 52는 그림 50의 Layout에서 생성된 Image와 Track File을 중첩하여 표기한 것이다. 중앙에서 동심원으로 퍼져나가는 파형의 마루와 초록색으로 표현된 Trace들이 일치하는 것을 알 수 있다.

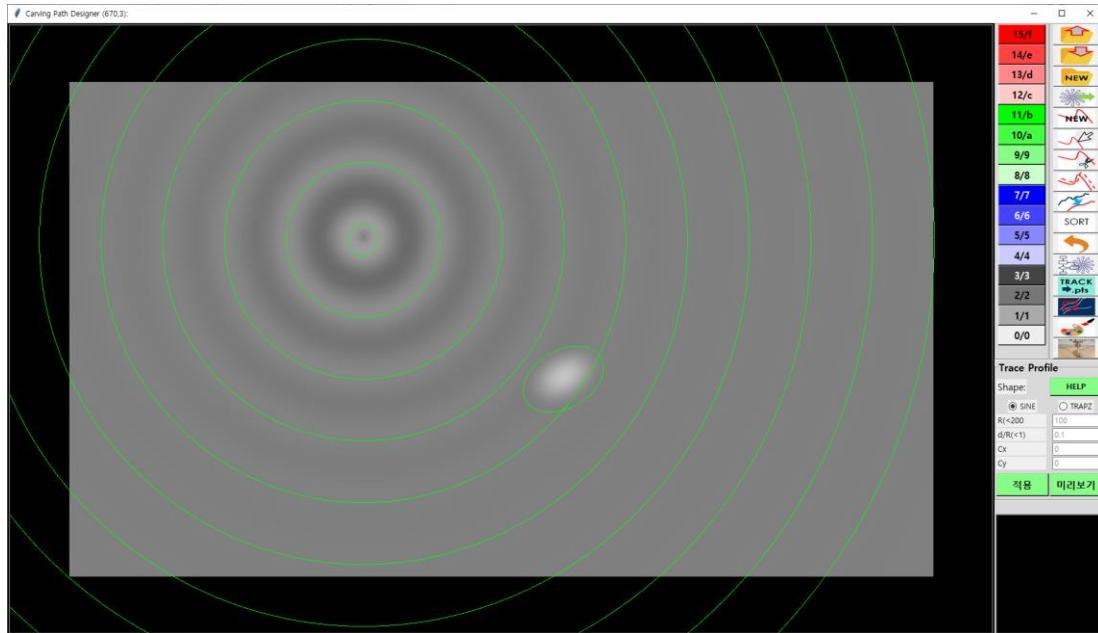


그림 52. Layout으로부터 생성된 Track File을 생성 Image와 중첩 표기한 모습. 초록색으로 표기된 Trace는 Layout object의 마루 또는 외곽선에 위치하고 있음을 보여준다.

Track File로 변환을 하여 저장하는 과정은 아래의 그림 53에 정리하였다.

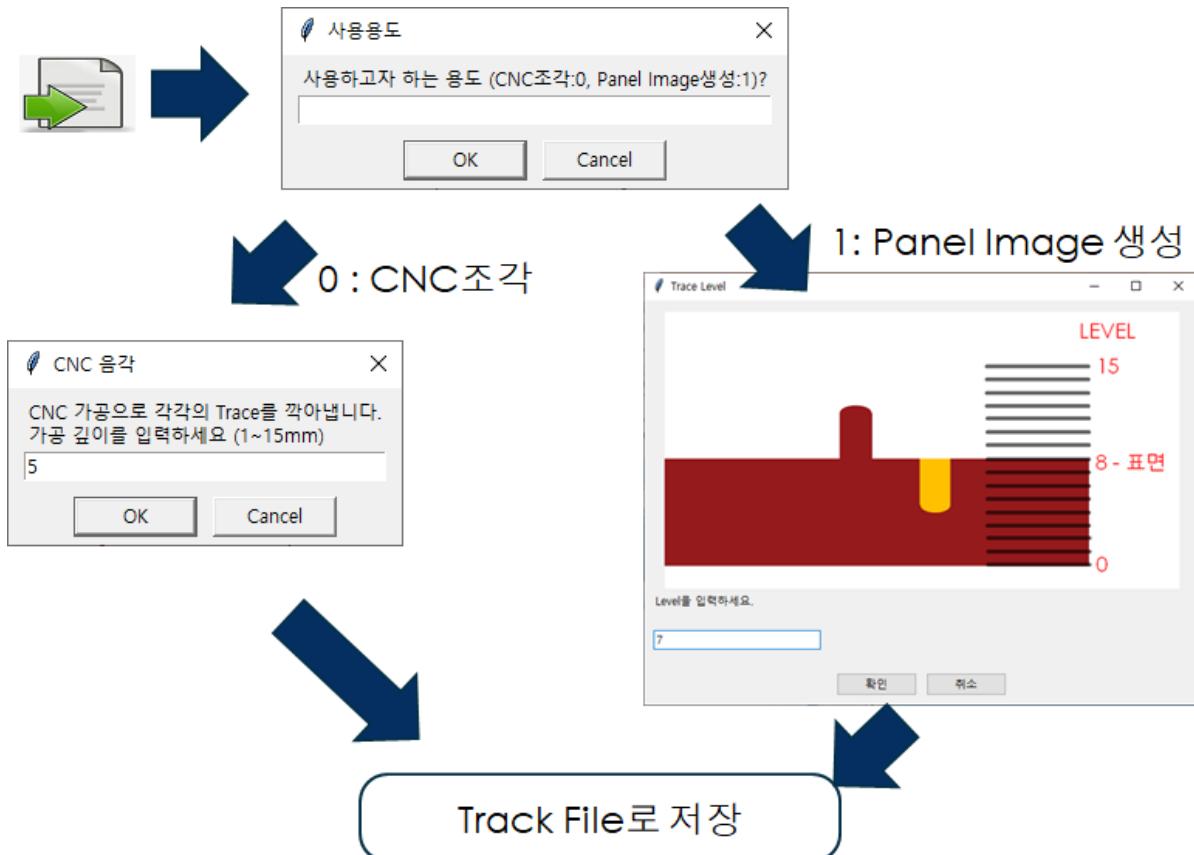


그림 53. Layout으로부터 Track File을 생성 저장하는 과정. 용도에 따라 Track File의 각 Trace에 표기되는 Level 값들의 default 값을 지정할 수 있도록 하고 있다. 이후 Track File은 PathMaker

를 이용하여 편집/활용할 수 있다.

4.2.3. Layout Item

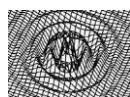
디자인에 활용될 수 있는 Item들이며 현재 7 종류의 Item을 정의하여 사용할 수 있다. 이들 각각의 Item들은 Layout Items창 안에 정렬된 7개의 Button 중 한 개로 지정할 수 있다.

- Item 추가

새로운 Item을 추가하려면 원하는 Button을 Click한다. 삽입이 된 Item에 관련된 파라미터들은 Layout Items창 아래에 표기되는 Parameter Field의 값을 통하여 변경을 하고 Update Shape Button을 눌러 형상 변경을 한다. 위치의 변경은 해당 Object 위에서 왼쪽 Button Press->Drag->Release의 과정으로 실행한다.

- Item 삭제

Item을 삭제하려면 원하는 Item 위에서 왼쪽 마우스 버튼을 Click하여 선택한 후 "Del" Key를 눌러 삭제한다.



- Radial Sine.

동심원으로 퍼져나가는 Sine파를 모사하기 위한 것으로 퍼져 나가면서 지수함수적으로 그 진폭이 감쇄해 나가는 파형을 만들어 낸다.

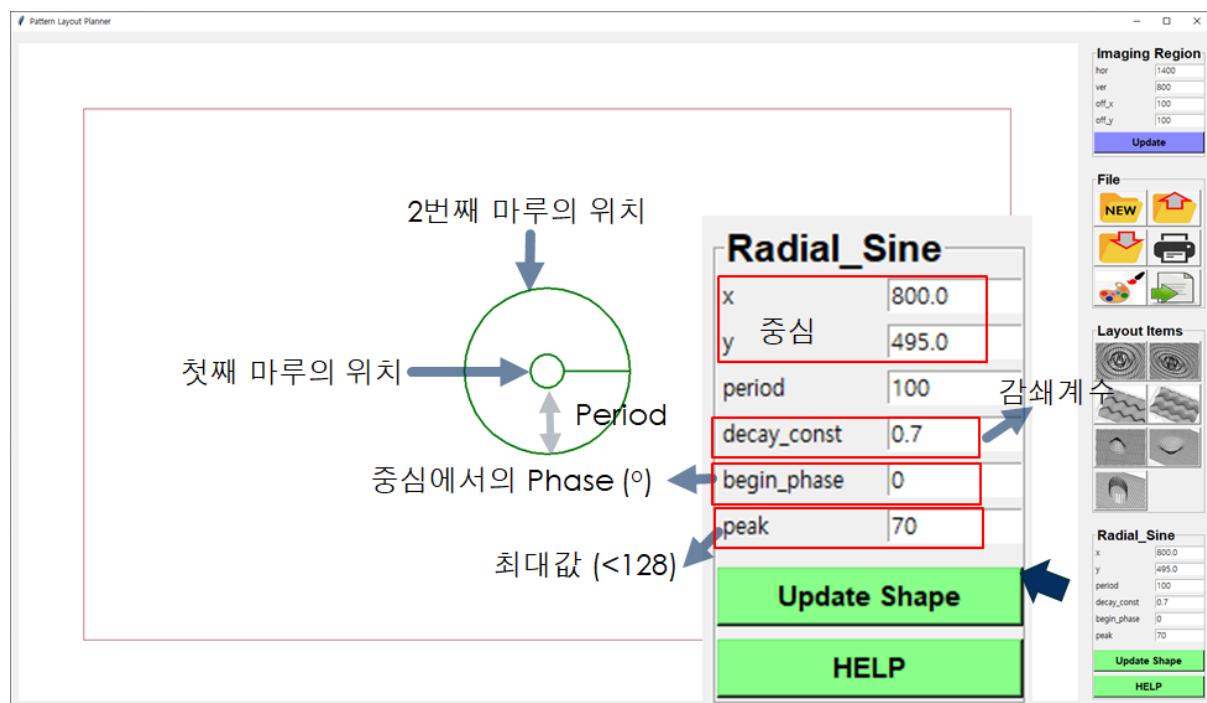


그림 54. Radial Sine파형의 Parameter. begin_phase는 중심에서의 각도이며 '0~360도 사이의 값으로 설정한다.

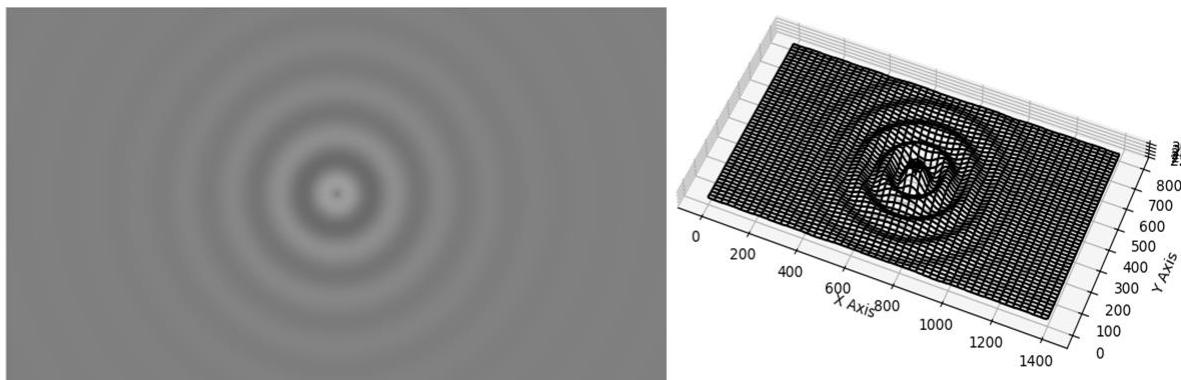


그림 55. Radial Sine파형의 생성 Image



중심에서 타원으로 퍼져나가는 Sine파를 모사하기 위한 것으로 퍼져 나가면서 지수함수적으로 그 진폭이 감쇄해 나가는 파형을 만들어 낸다.

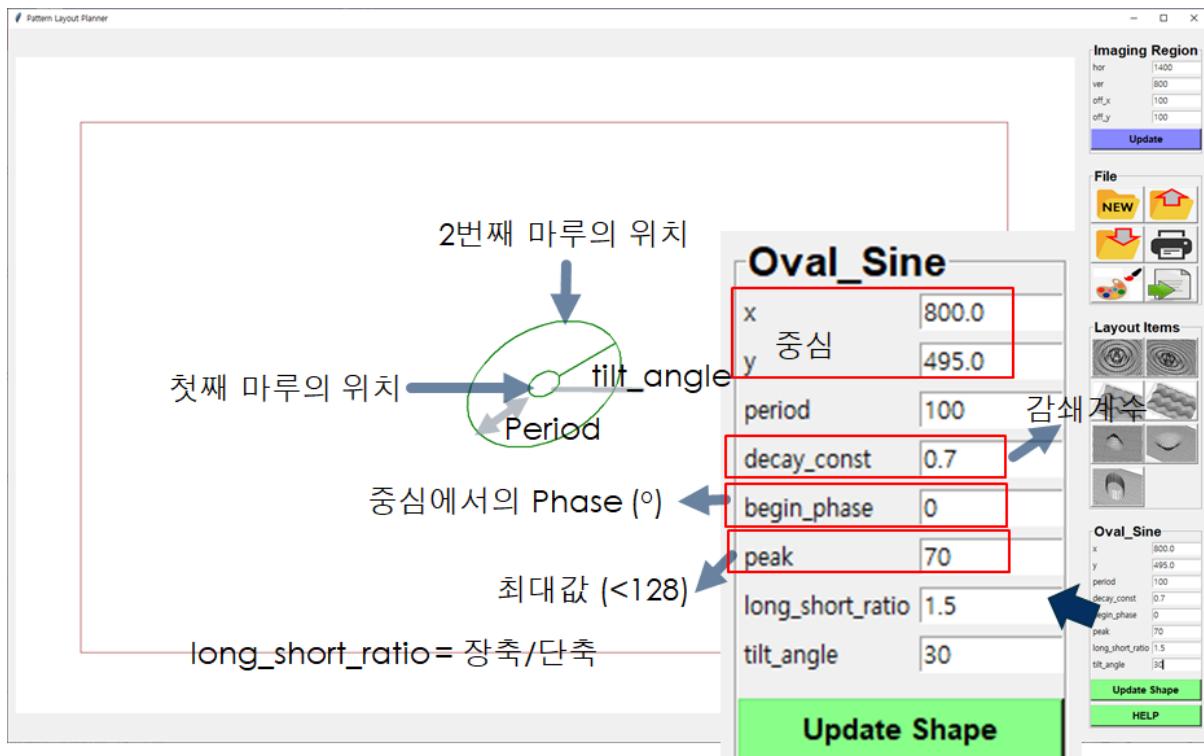


그림 56. Oval Sine파형의 Parameter. begin_phase 및 tilt_angle은 0~360도 사이의 값으로 설정 한다.

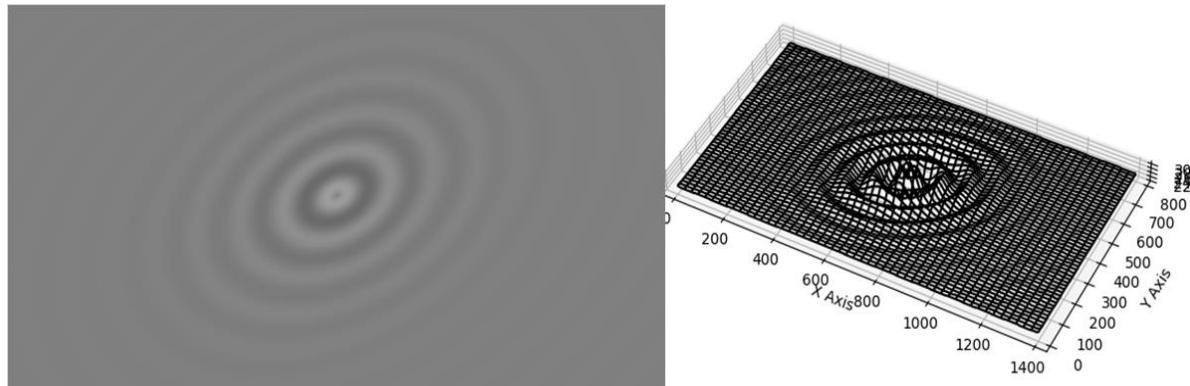


그림 57. Oval Sine파형의 생성 Image



마루의 위치가 SINE파에 의거하여 변화하는 파형이 수직방향으로 감쇄하지 않고 전파되는 Wave를 모사한다.

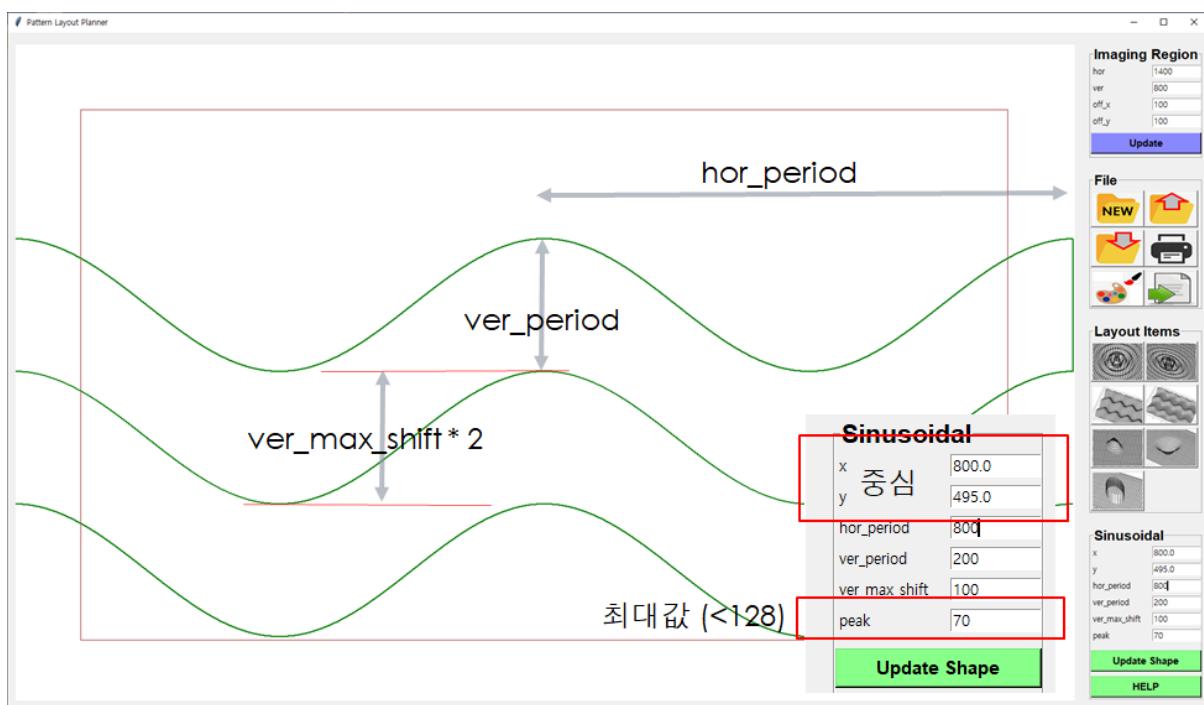


그림 58. Sinusoidal파형의 Parameter

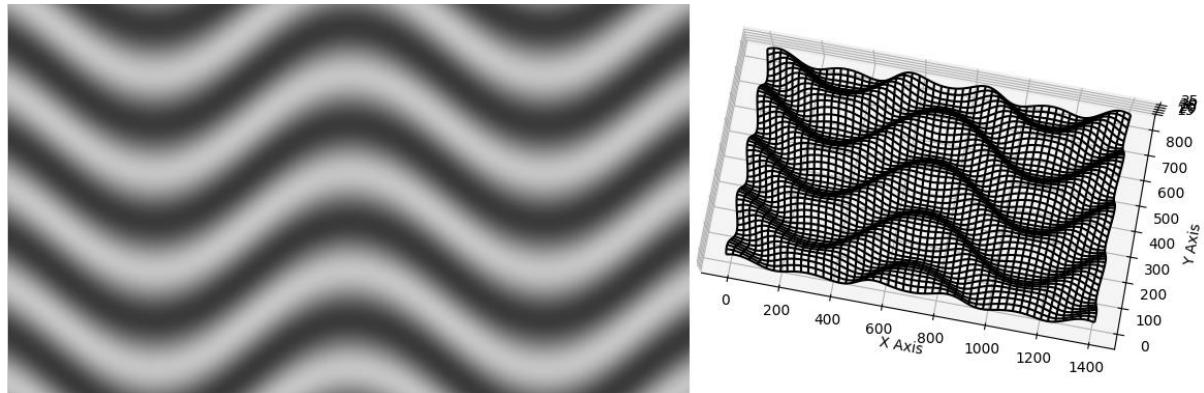


그림 59. Sinusoidal파형의 생성 Image



바람에 의하여 형성된 사막의 모래 표면을 형상화 한 것으로 오르막 구간과 내리막 구간이 반복되는 형상을 만들어 낸다. 오르막 및 내리막 형상은 2차함수를 따른다. 마루의 위치는 Sinusoidal 파형과 동일한 형태로 배치한다.

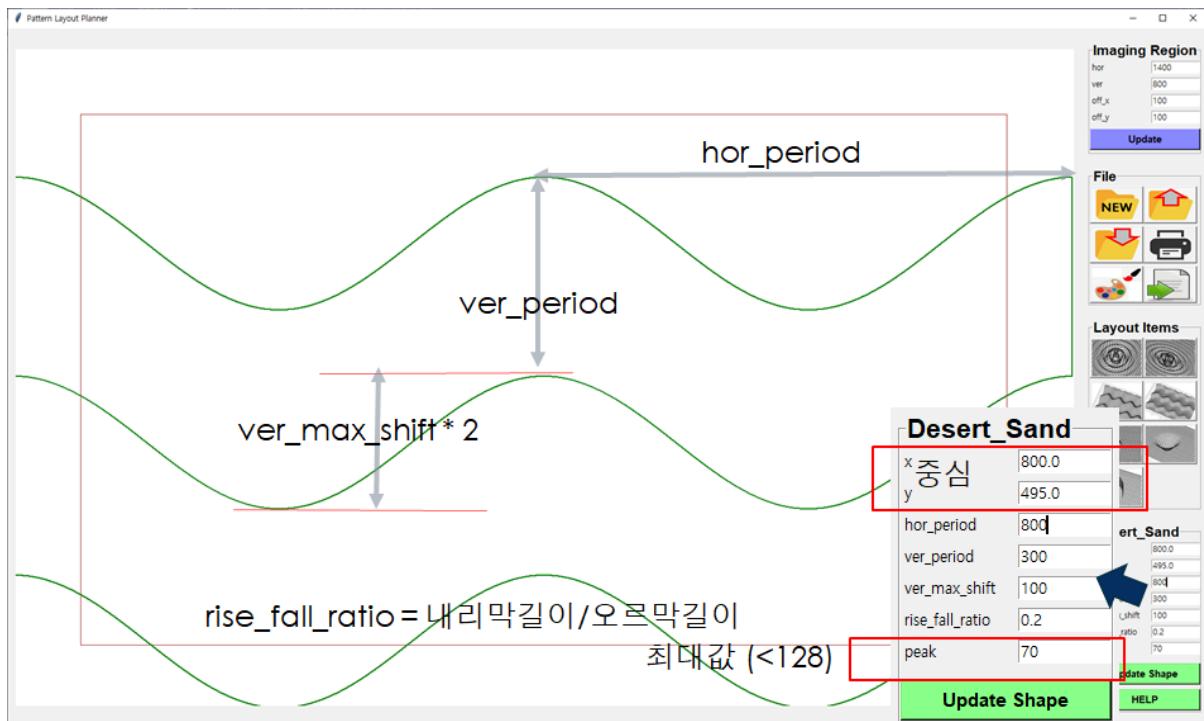


그림 60. Desert Sand파형의 Parameter

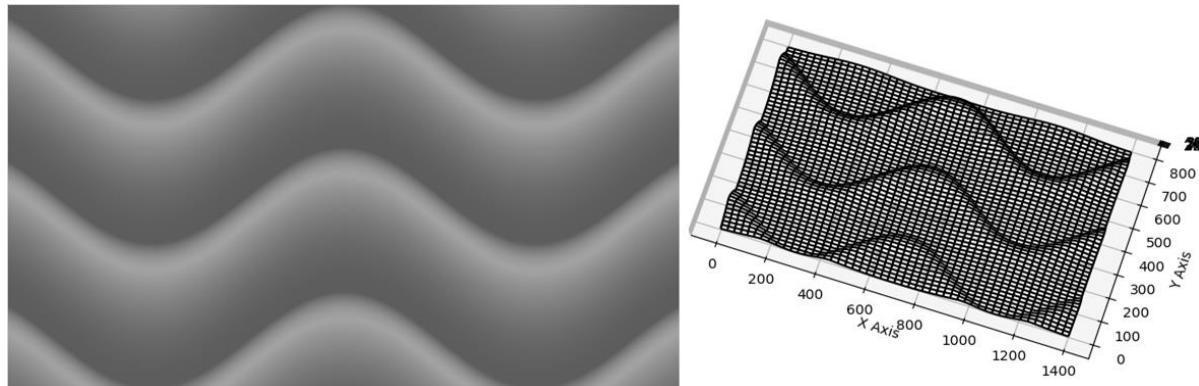


그림 61. Desert Sand파형의 생성 Image



바닥이 타원형을 가지는 모양의 봉우리를 만든다. 중심을 지나는 수직면을 따라 자른 단면의 Profile은 sine파의 -90도에서 90도까지의 반 주기 형상을 따른다.

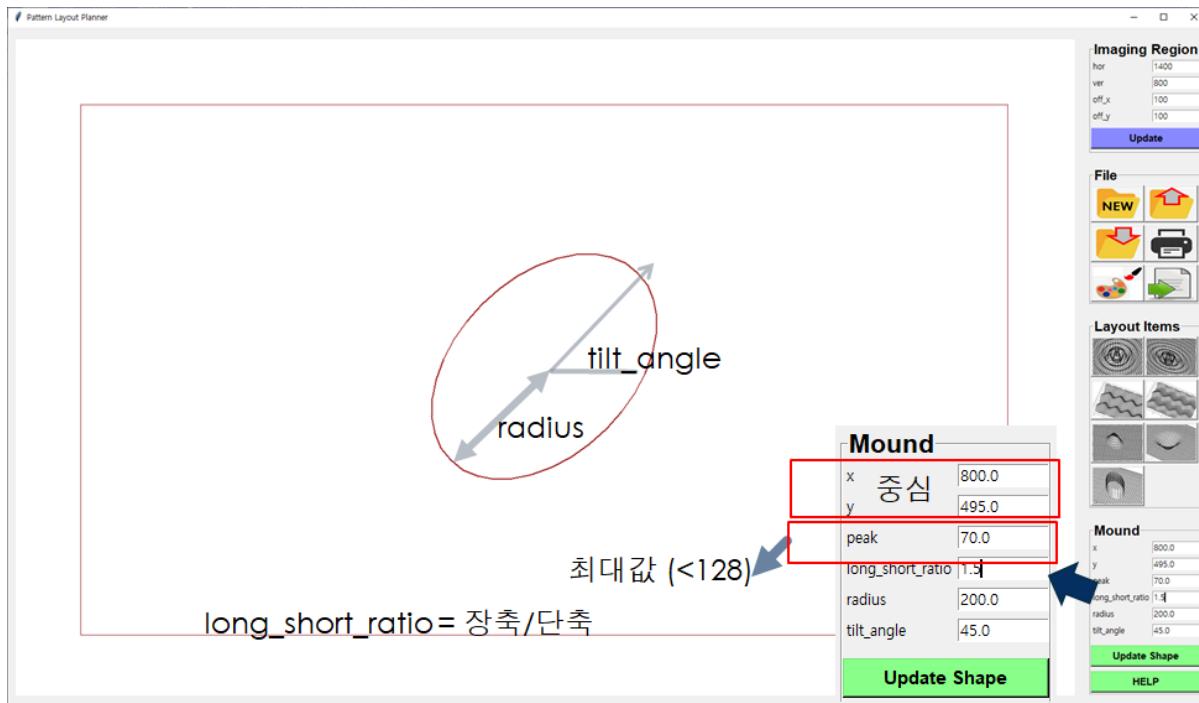


그림 62. Mound 형상 Parameter

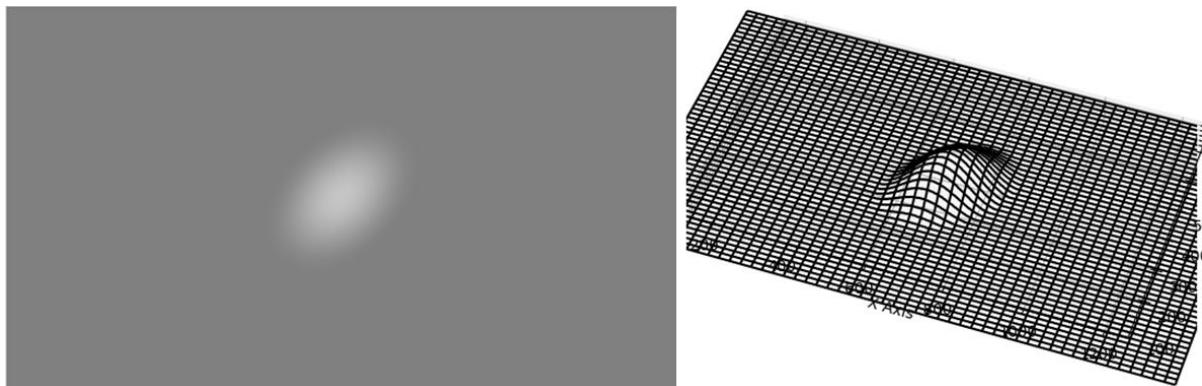


그림 63. Mound 형상의 생성 Image



Mound와 정반대로 윗면이 타원형이며 웜푹 파인 형상을 만들어낸다. 중심을 지나는 수직면을 따라 다른 단면의 Profile은 sine파의 반 주기 형상을 따른다.

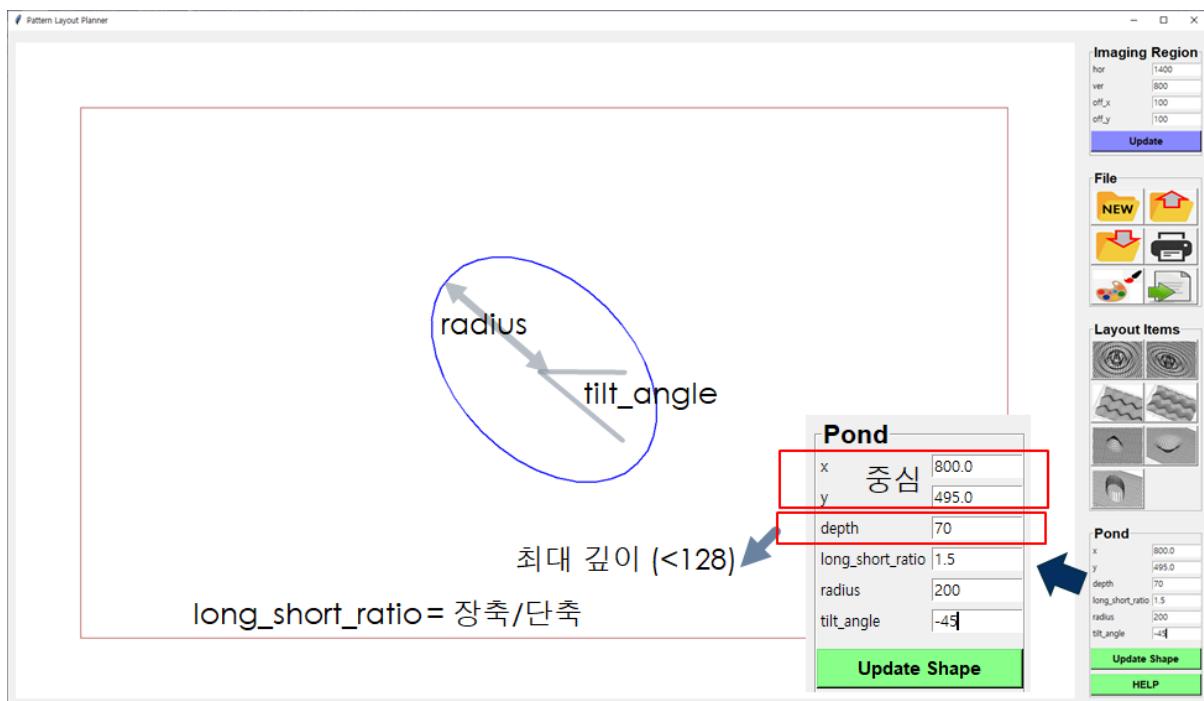


그림 64. Pond 형상 Parameter

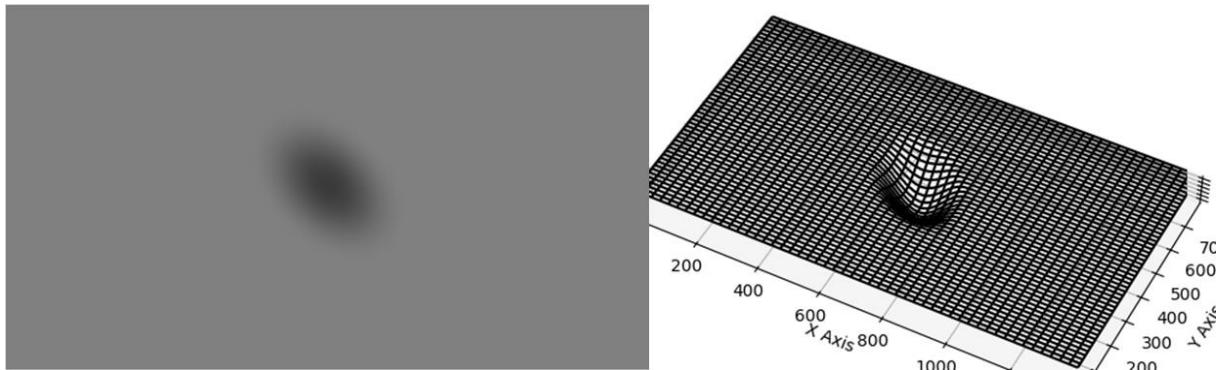


그림 65. Pond 형상으로 생성된 Image



바닥이 타원형이며 높이(깊이)가 일정한 타원형의 형상을 만든다. Height가 양수인 경우 돌출된 모양으로 음수인 경우에는 함몰된 형태로 생성된다.

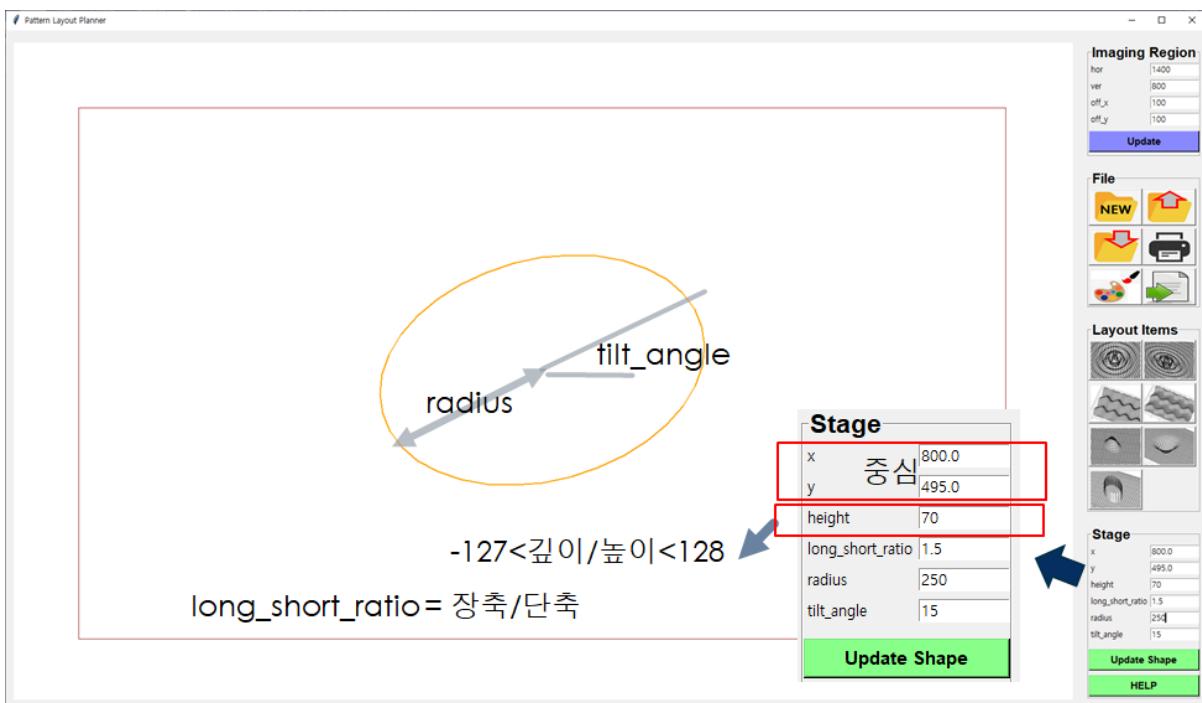


그림 66. Stage Parameter

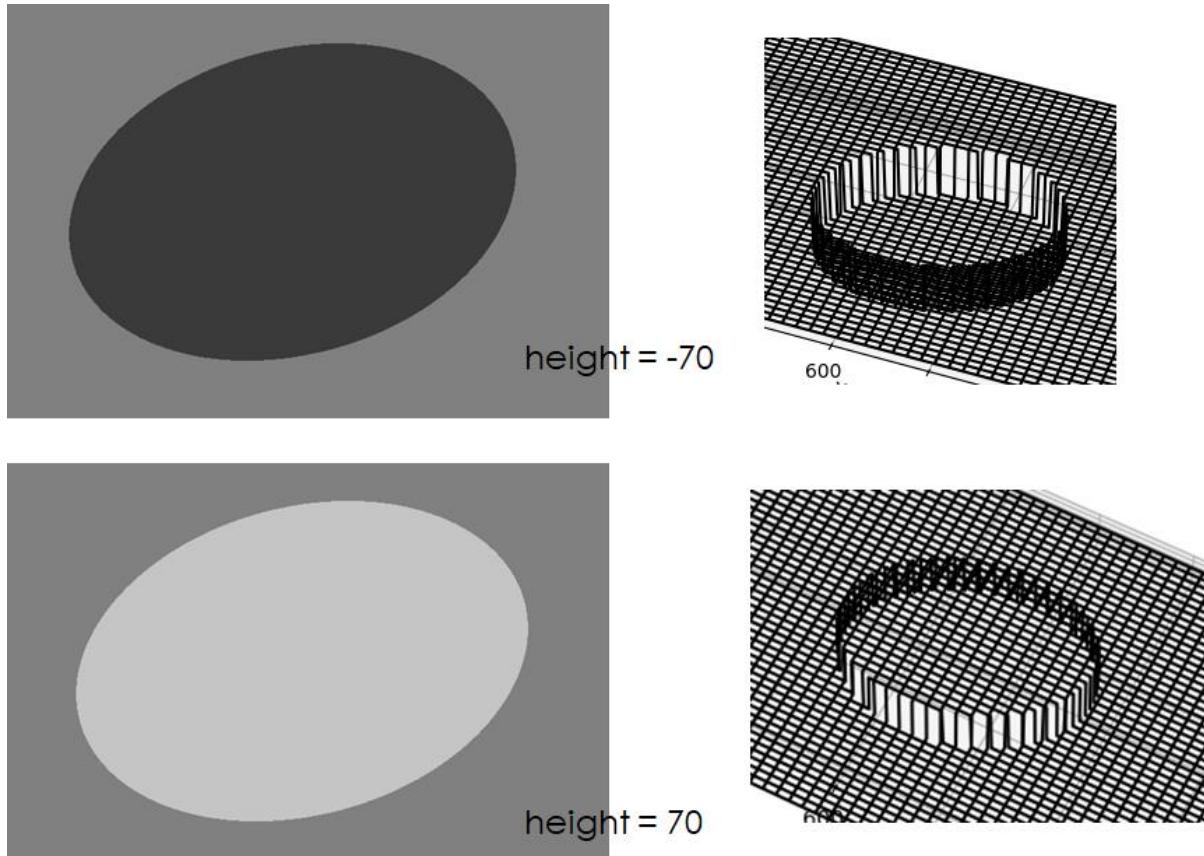


그림 67. Stage로 생성하는 Image들. 위: 함몰 Image, 아래: 돌출 Image

4.2.4. Item 중첩 Image 생성

여러 개의 Layout Item들이 배치된 경우 이것들이 중첩하여 만들어내는 Image를 생성해 낸다. 마치 연못에 여러 개의 돌을 던져 생겨나는 파동 및 이들 간의 간섭에 의한 형상 등 다양한 모양을 생성해 낼 수 있다. 각각의 Item에 의하여 형성되는 Pixel 값은 이전에 같은 위치에 있는 Pixel 값에 중첩이 되어 표기된다. 이를 통하여 파동들의 간섭 등이 표현될 수 있도록 하였다. 아래 그림 68은 4개의 Radial Sine파를 배치하고 합성한 Image를 보여준다.

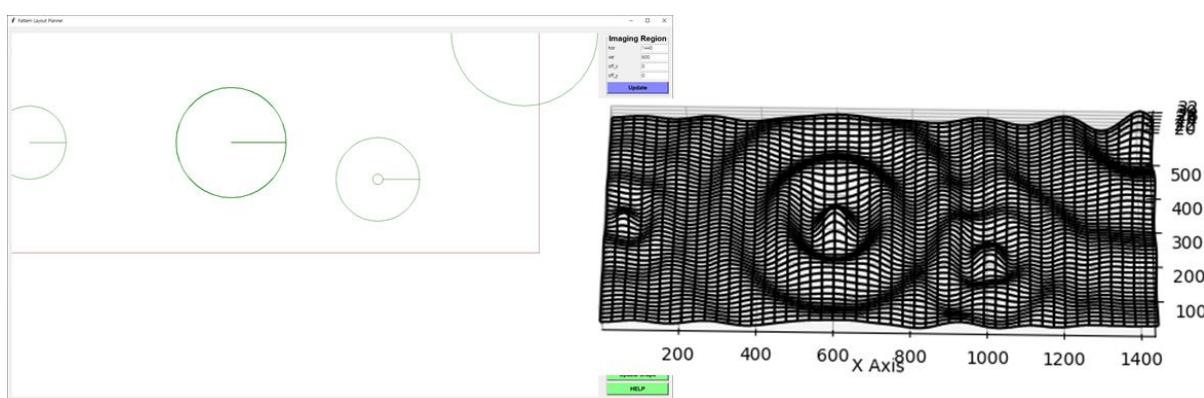


그림 68. 4 개의 Radial Sine파형을 중첩한 Image 형상. Layout Design (왼쪽), 합성 Image(오른쪽)

쪽)

그림 69는 다른 형태의 Layout에 기반한 Image 합성의 예를 보여준다.

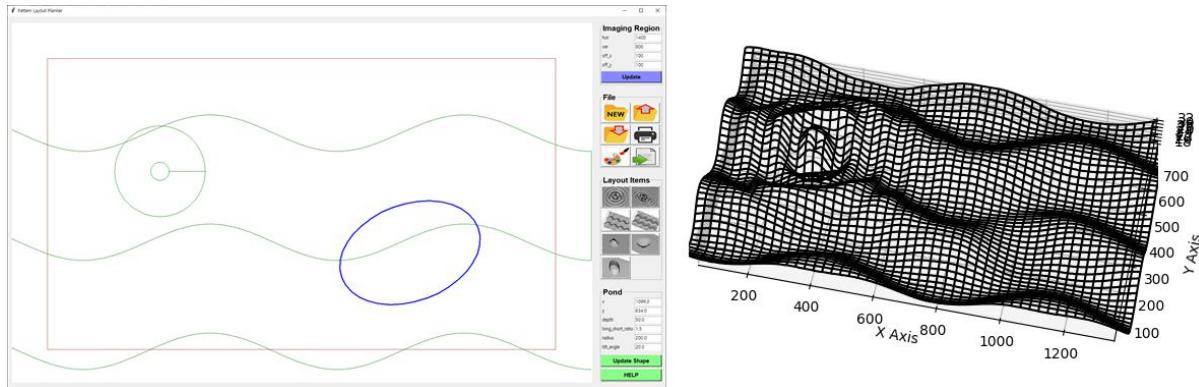


그림 69. 다양한 Layout Item을 중첩하여 합성한 Image(오른쪽). Layout Design (왼쪽)

GeoLaD에서 Image 합성을 할 때는 각각 화소의 위치에서 해당되는 Item에 의하여 발생하는 기여 값을 모두 중첩하여 새로운 Pixel 값으로 설정한다. 여기에서의 Pixel 값은 16bit integer에 기반한 값을 사용하며 경우에 따라 Saturation이 발생할 수 있다. 이를 방지하기 위한 수단으로 각 Layout Item의 Peak 값은 되도록 크지 않게 설정하고 Saturation 또는 형상의 구별이 힘들 정도의 희미한 형상이 생겨나는 경우 peak 값을 적절히 조정하여 Image를 다시 생성하는 것이 바람직하다.

5. Reference

5.1. Video

Parametric Wall 설계 및 Panel의 생성에 이르는 과정은 아래의 Video에서 참조할 수 있다.



<https://www.youtube.com/watch?v=O9kDwJ-cUOQ&t=58s>

5.2. Woodmeister 사용법

Manual:

<https://github.com/user-attachments/Files/15508661/woodmeister.docx>

Youtube:

https://www.youtube.com/playlist?list=PLHzQ8JkLY6_d3xQi7onElSgiXFULc0cMK

내막대로 Backloaded Horn Speaker 만들기

Hifi Speaker - Backloaded Horn 스피커 만들기

세상에 없는 나만의 Backloaded Horn Speaker 만들기 - tower형스피커설계 part2

세상에 없는 나만의 Backloaded Horn Speaker 만들기-tower형스피커설계 part1

Tower형 스피커 설계 및 제작

설계에서 제작에 이르는 모든 것을 지원해주는 나만의 Backloaded Horn Speaker 를 만들기 위한 solution.

세상에 없는 나만의 Backloaded Horn Speaker 만들기 - tower형스피커설계

5.3. 목공과 Coding 의 만남

Document:

<https://github.com/user-attachments/Files/15508660/Coding.docx>

Youtube: https://www.youtube.com/playlist?list=PLHzQ8JkLY6_fipJYY3eOBcDbp1IV-8dA0

