yskim57@gmail.com

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

Parametric Wall DesIgner

멋진 예술공간을 만들어주는 Parametric Wall 전용 Design Program

목차

[1. Parametric Wall이란? 2](#_Toc168058173)

[2. Image 기반의 Parametric Wall Design Program 4](#_Toc168058174)

[2.1. 화면 Layout 4](#_Toc168058175)

[2.2. Button 기능 5](#_Toc168058176)

[3. Reference 17](#_Toc168058177)

[3.1. Parametric Wall 설계 및 Panel의 생성에 이르는 과정은 아래의 Video에서 참조할 수 있다. 17](#_Toc168058178)

# Parametric Wall이란?

Parametric wall 은 예술성과 음향 효과를 모두 갖춘 최상급의 Interior의 용도에 많이 사용하고 있습니다. 다양한 재질로 이를 구성할 수 있으나 일반적으로 목재 또는 이와 비슷한 성질의 재료를 많이 사용하고 있습니다. 아래의 그림은 pinterest 에서 parametric wall 을 입력하여 나열한 예로써 매우 다양한 pattern 을 사용하여 만들 수 있으며 그 용도에 따라 pattern의 design 방법도 서로 달리할 필요가 엿보인다.

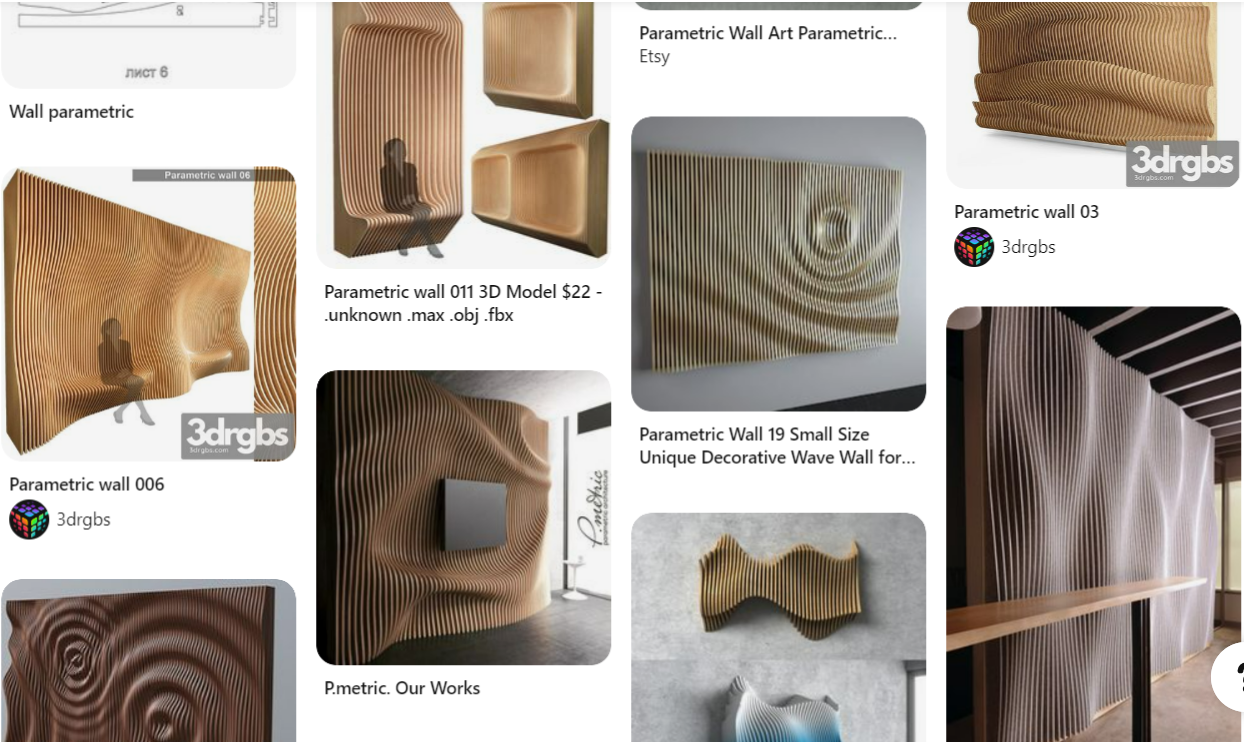


그림 1. Parametric Wall의 예

본 문서에서는 Image를 기반으로 만드는 Parametric Wall 설계 방법 및 이를 구현한 전용 Design Program에 대한 설명을 하고자 한다.

아래의 그림은 이의 전형적인 화면의 하나로 기본이 되는 Image 를 열어 놓은 상태의 화면을 보여준다.

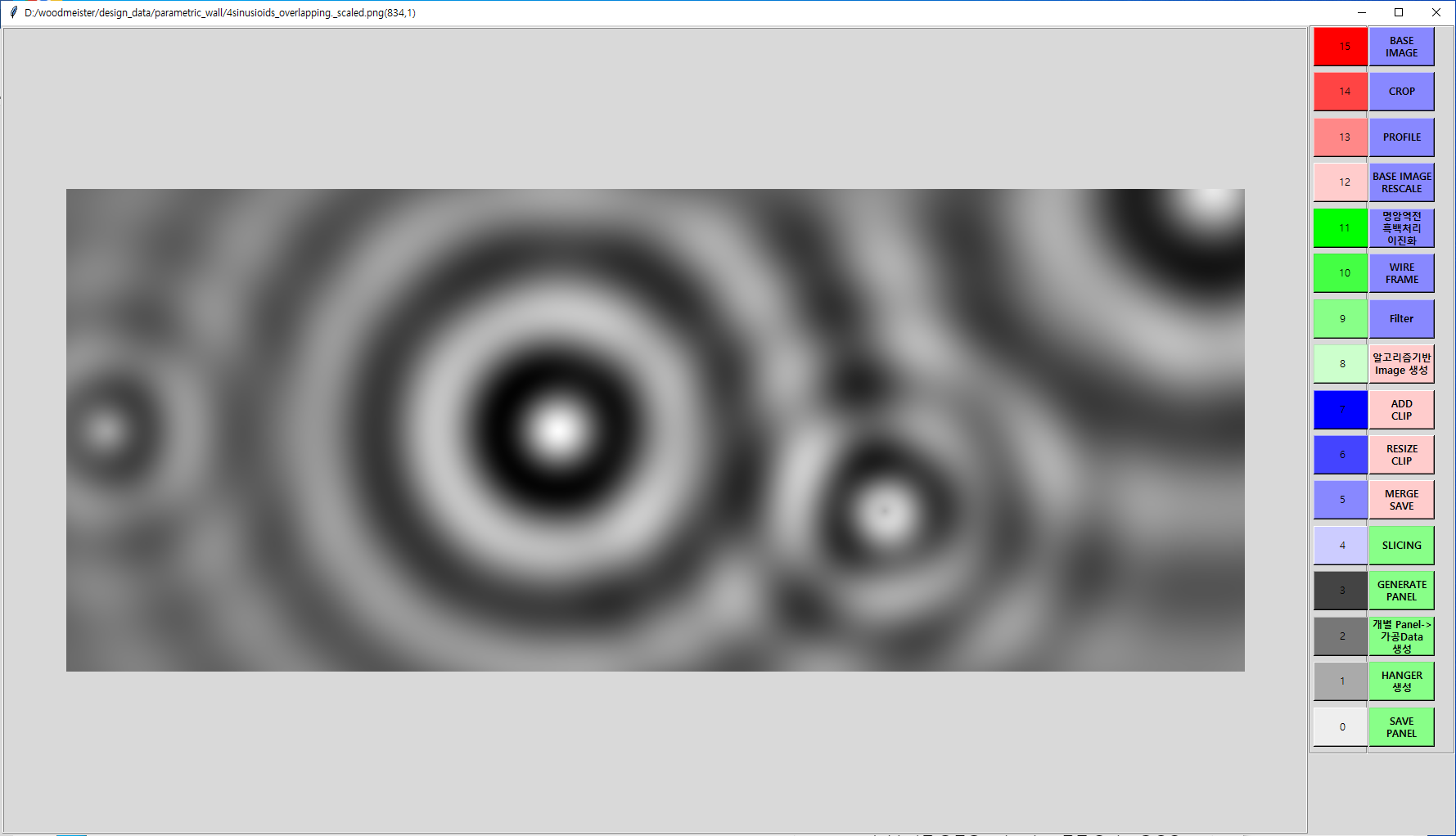


그림 2. Parametric Wall Design Program 의 실행 화면 예

이 Image 를 처리하여 1440mm x 600mm 의 크기로 제작한 Parametric Wall 의 Sample이 그림 3이다.

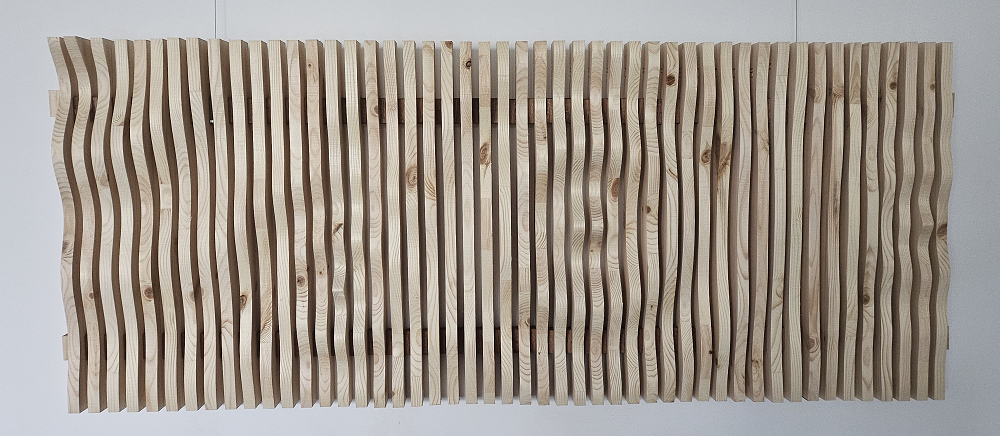


그림 3. 그림 2의 Image에 기반하여 만든 제작 sample (크기 : 1440mm x 600mm)

# Image 기반의 Parametric Wall Design Program

pametric\_wall\_image는 Python으로 작성한 프로그램으로 Image 의 처리를 통하여 parametric wall을 Design하고 이에 소요되는 여러 개의 가공품의 도면을 Woodmeister에 호환되는 Format으로 생성해낼 수 있는 프로그램이다. 이에는 실제 가공에 바로 적용하는 최적의 배치를 포함하고 있으며 이에 기인하여 소재의 낭비가 없이 빠른 속도로 가공할 수 있다. 그림 4는 이 프로그램을 처음 수행하였을 때의 화면이다.

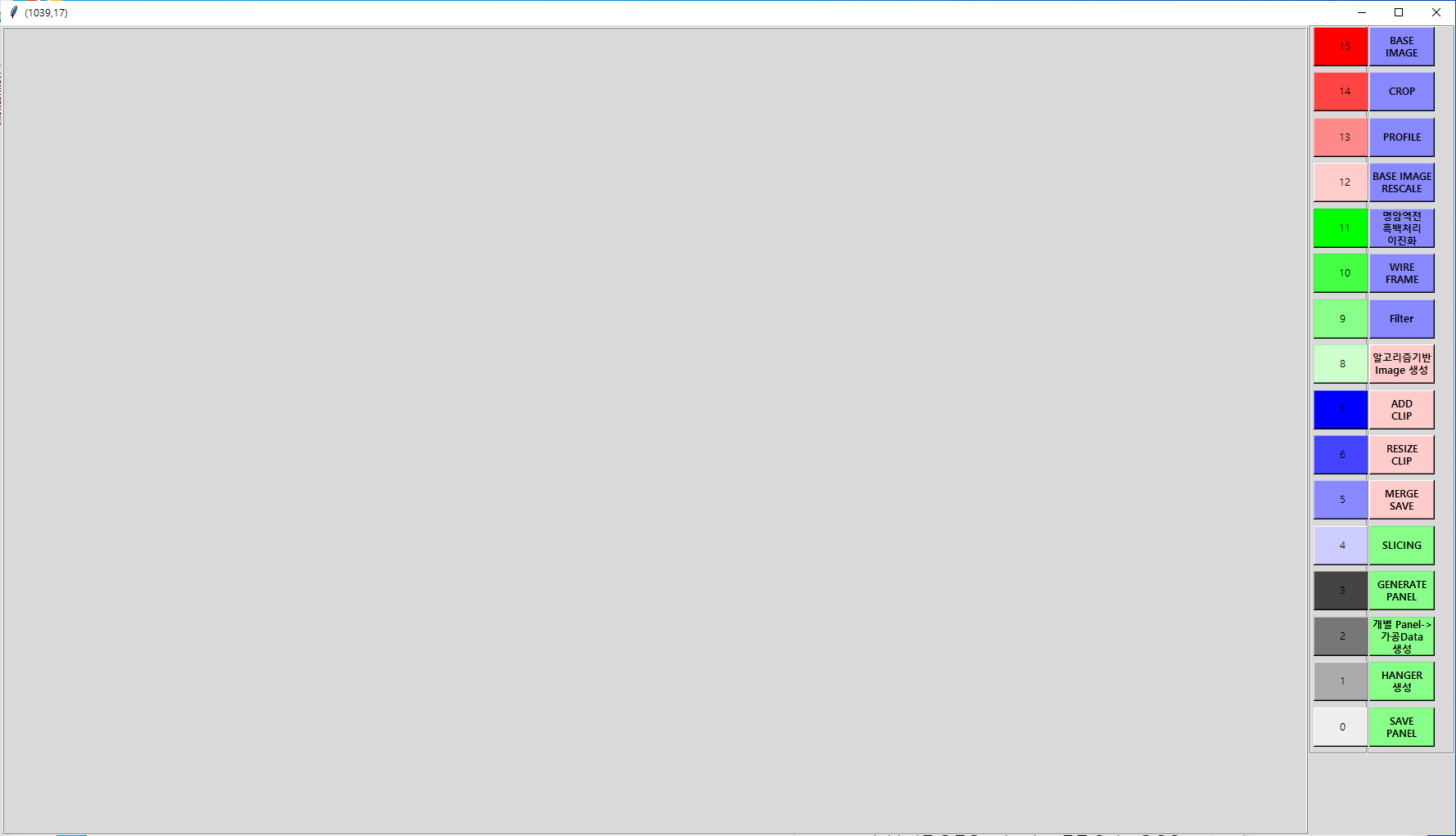


그림 4. Parametric\_wall\_image의 수행 후 첫 화면

## 화면 Layout

그림 4의 오른쪽에 16가지의 대표적인 Level 을 나타내는 색상을 가지는 color (level) button 들과 그 오른쪽에 Image의 조작 등에 활용하는 Button 들이 나열되어 있다.

현재 Level 에 관한 Button 은 사용하지 않고 있으며 그 오른쪽에 위치한 Button들에만 기능이 부여되어 있다.

Screen 의 대부분을 차지하는 왼쪽 부분이 Design 화면으로써 여기에 처리하고자 하는 Image 와 이에 부수되는 정보들이 표기될 수 있다.

기능이 부여된 Button들은 크게 3가지의 부류로 나뉘어 있으며 이들은 각각 다른 색으로 표기되어 있다. 3개의 부류는 ‘Image 처리, Image 생성/합성, CAD data 생성’에 해당하는 버튼들이다.

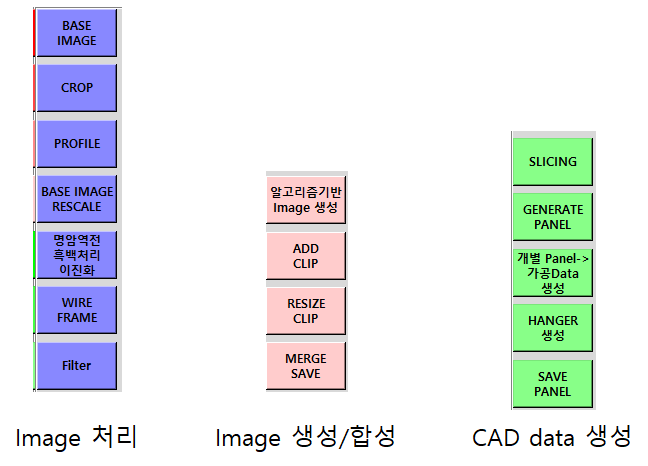


그림 5. 기능이 부여된 Button 들의 분류

## Button 기능

Button 상에 표기된 글짜를 버튼의 이름으로 정하고 설명을 진행하도록 한다.

: 기반이 되는 Image 를 열어서 화면에 띄워 준다. 여기에 사용하는 Image 는 가로/세로의 크기가 각각 화소수로 1599 x 990이 넘지 않도록 하여준다. 만일 크기가 이보다 크다면 다른 편집 수단을 사용하여 이 크기 이하로 줄여서 처리하는 것을 권장한다. 또한 Image의 각 화소는 16bit 화소 값을 가지는 것이 바람직하다. 이를 위하여 가급적 16bit PNG image를 기반으로 하는 것을 권장한다.

: 선택된 영역을 잘라내어 다른 이름으로 저장하도록 한다. 영역을 선택을 위하여 왼쪽최상단점에서 Mouse 왼쪽버튼을 누른 상태로 오른쪽 하단까지 Drag한 후 Release 한다. 선택된 영역이 초록색 점선으로 표기된 사각형으로 표기된다. 이 후 CROP button 을 눌러서 선택된 영역을 새로운 Image 로 저장할 수 있다. 저장은 16bit PNG file 로 이루어 진다.

: Base Image 의 단면을 표기하여 준다. 그림 6과 같이 표기가 된다. Image의 네개의 변에 표기되는 Graph중 붉은색으로 표기된 것은 해당 변의 Gray level 의 분포를 0~255 사이의 값으로 표기한 값이며 녹색으로 표기된 그래프는 해당 Row 또는 Column 상의 최대치를 표기한다.

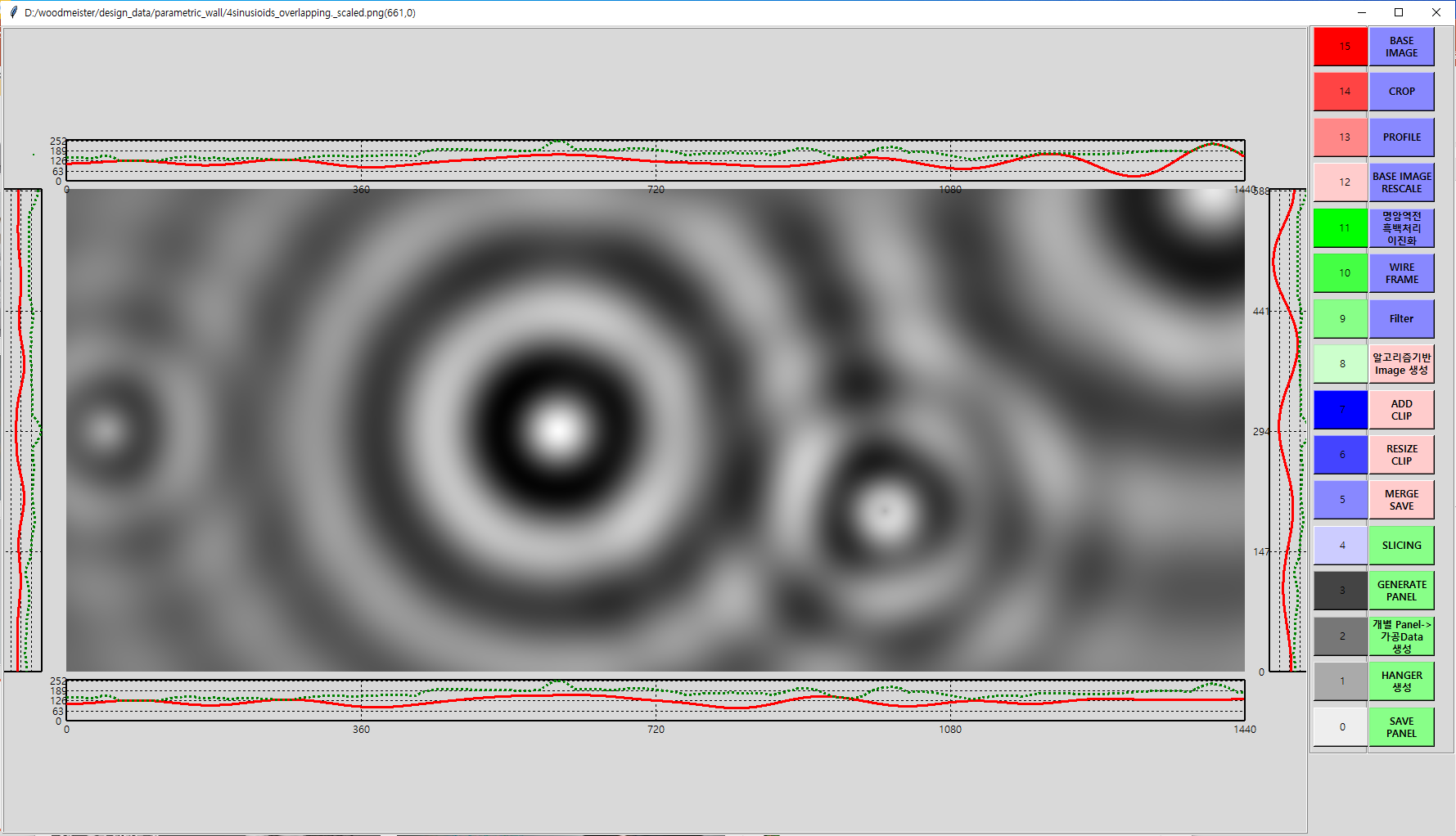


그림 6. Profile button 을 누른 후의 화면. Image 의 개략적인 Profile 을 표기하여 준다. 붉은색 선: 해당 변의 Pixel 값을 0~255의 값으로 Normalize 한 값을 표기. 녹색선:해당하는 row 또는 column 상의 최대치 값을 0~255의 값으로 Normalize 한 값을 표기.

: Base Image 의 크기 및 Gray level를 조절하여 준다.

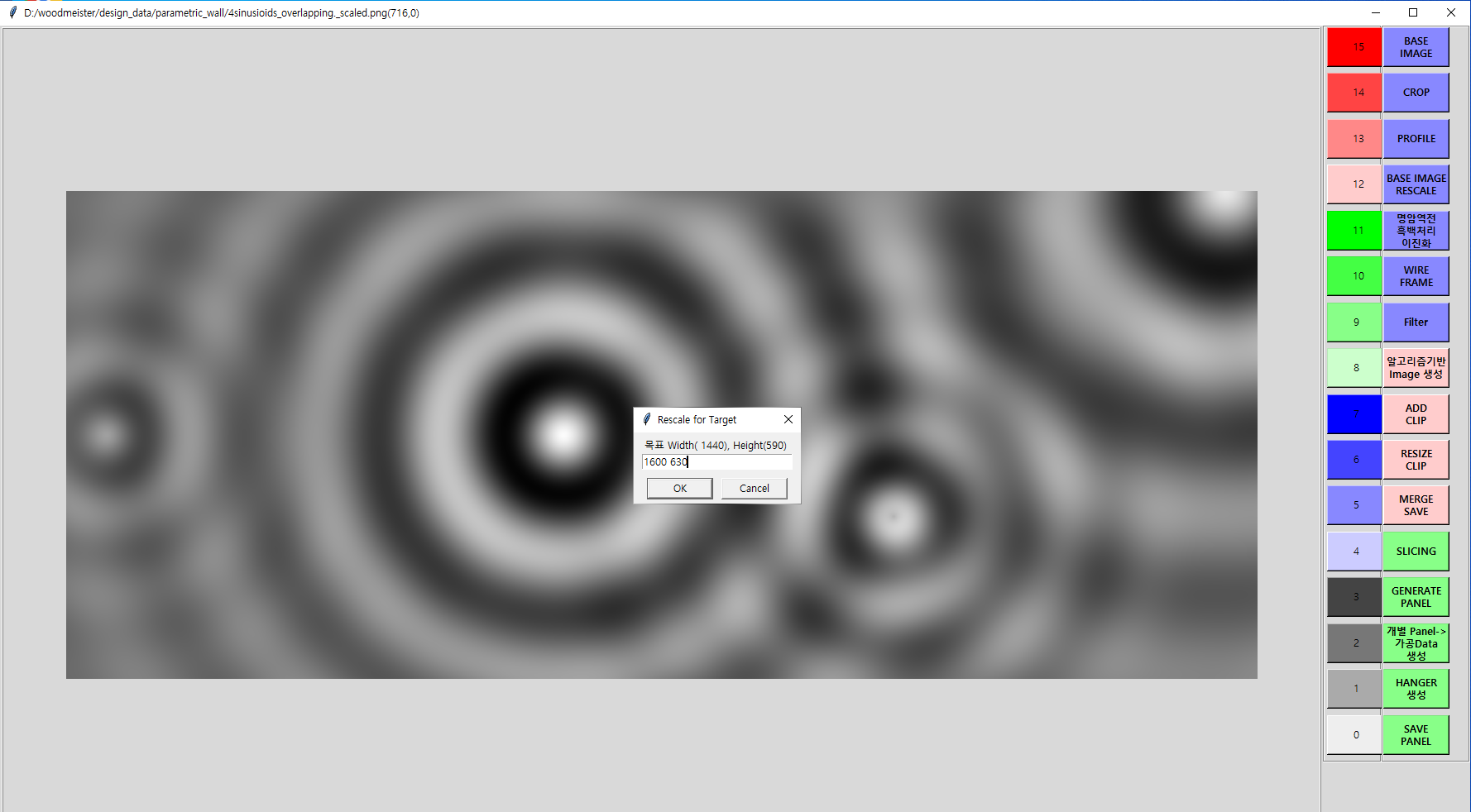
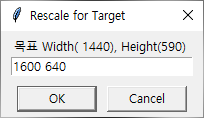
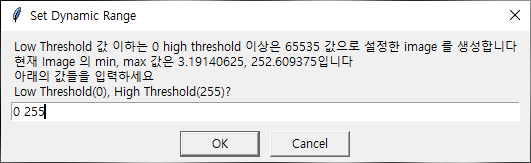


그림 7. BASE IMAGE RESCLAE 화면.

그림 7의 화면에서 수평 수직의 크기를 원하는 값으로 설정한다. 단위는 mm 이다. 크기를 1600 x 640으로 변환하고 싶으면 아래와 같이 값을 입력 후 “OK”버튼을 누른다.



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



완료가 된 후 Image 는 기본image file 이름+ “\_scaled.png”로 저장된다. 작업이 끝난 후 생성된 image 를 열어서 화면에 띄워주며 이것이 Base Image 로 취급된다.

: 화면에 표기된 image 의 화소값을 변경하여 새로운 Image로 저장하고 화면에 띄워준다.

* Invert : 화소의 값을 역전시킨다. 최대값->최소값, 최소값->최대값

저장 Image 는 기본image file 이름+ “\_inverted.png”로 저장된다.

* 이진화 (Binarization) : 임계치 이상을 high, 이하는 low 값으로 변환한다.

저장 Image 는 기본image file 이름+ “\_binary.png”로 저장된다.

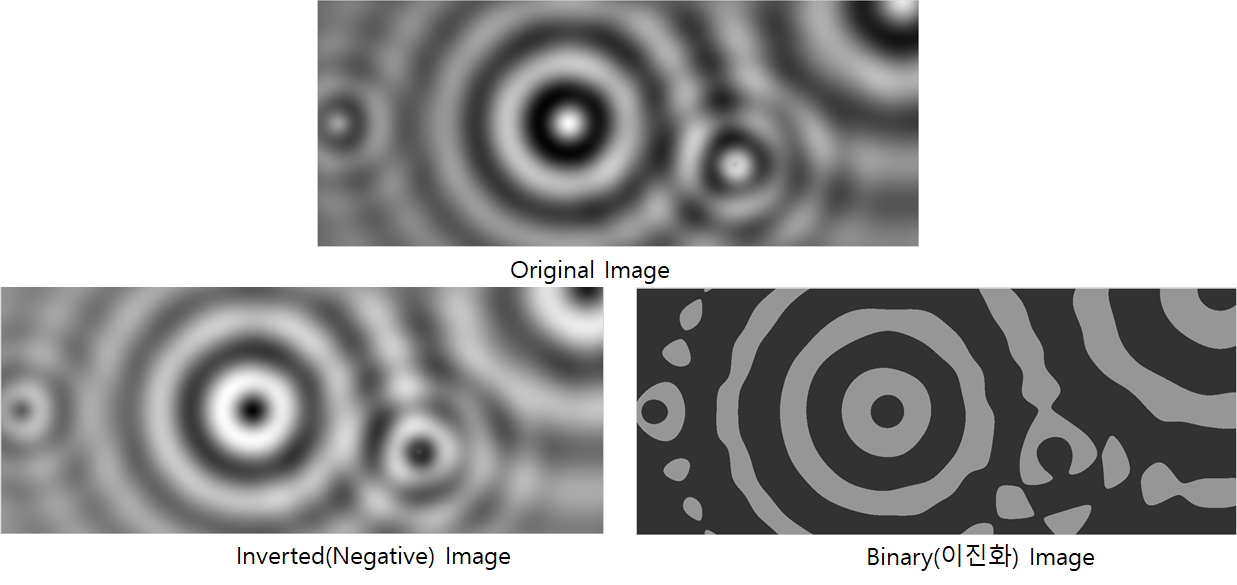


그림 8. 원래의 Image(위)와 Inverted(왼쪽 아래), 이진화(오른쪽 아래) Image

* Grayscale : Color Image 를 흑백Image 로 변환시킨다.

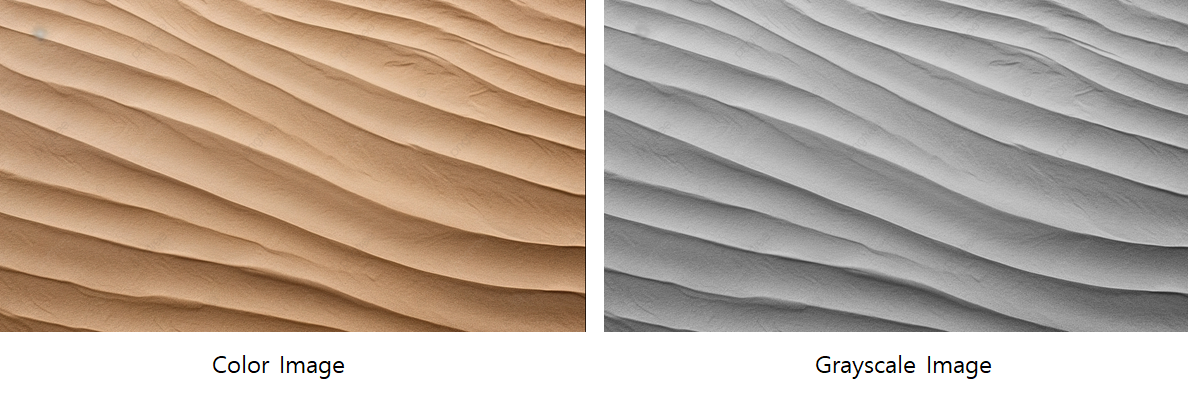


그림 9. Color image의 Grayscale image 변환

* 컬러지형도 변환 : Color 로 표기된 지형도를 흑백지형도로 변환한다.

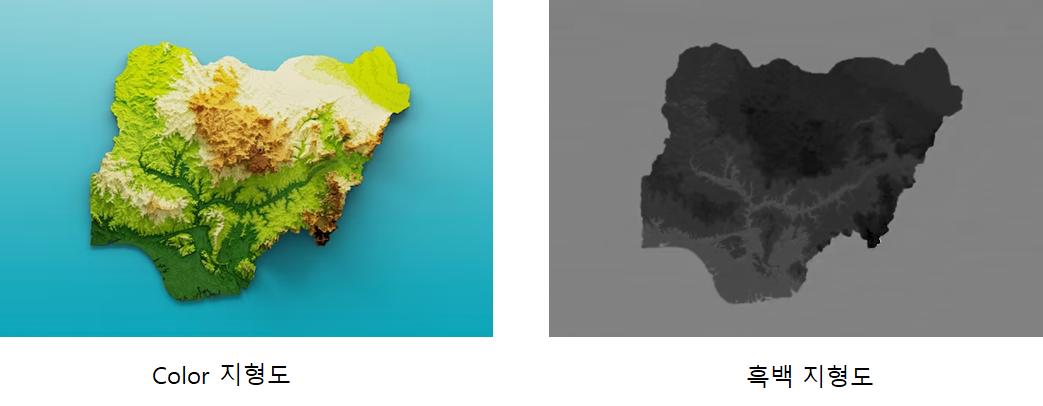


그림 10. Color지형도의 흑백 지형도 변환

 : Image 의 각 화소값을 높이로 간주하는 3차원 View를 보여준다. View는 wire frame 형식으로 표현한다.

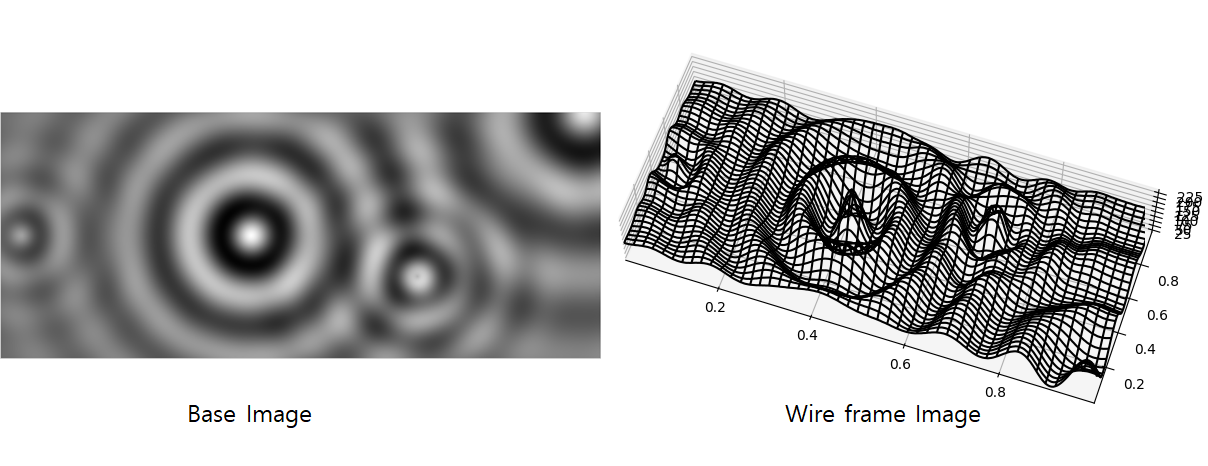
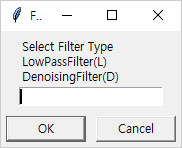
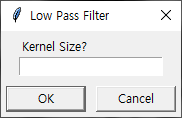
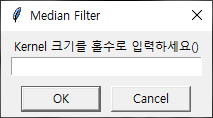


그림 11. Base Image와 이것을 Wire frame으로 표현한 결과. Wire frame 은 제작 및 설치가 완료된 후 대략의 형태를 3차원으로 보여준다.

: Base image에 가공 및 표현이 부적절한 지나치게 세세한 부분 또는 Noisy pixel (pepper noise 등)을 포함하고 있을 때 이를 완화시키기 위한 저역 filter(Low Pass Filter) 또는 잡음제거 filter(Denoising filter) 를 적용하도록 한다.



Low pass filter와 Denoising filter의 경우 각각 아래와 같은 화면에서 kernel 의 크기를 입력한다. Kernel은 처리를 위한 연산을 할 때 한 개의 pixel 에 대한 처리를 할 때 포함하는 인접화소의 갯수를 나타낸다.

 또는  와 같은 화면에서 kernel 의 크기를 입력한다.

* Denoising의 경우 base image 이름 + “\_denoise.png”로 저장된다.
* Low pass filter 의 경우 base image 이름 + “\_lpf.png”로 저장된다.

 : 수학적인 방법으로 다양한 형태의 Image 를 생성한다. 이를 실행하면 여러가지의 Image 생성 방법을 사용할 수 있다. 이를 생성하는 프로그램은 다양한 Needs 를 즉시 반영할 수 있도록 Source program 으로 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 Image 로 표기하였다.

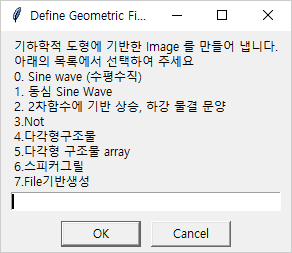


그림 12. “Algorithm 기반 Image 생성” 메뉴 실행 첫 화면의 예

* Sine wave (수평 수직)

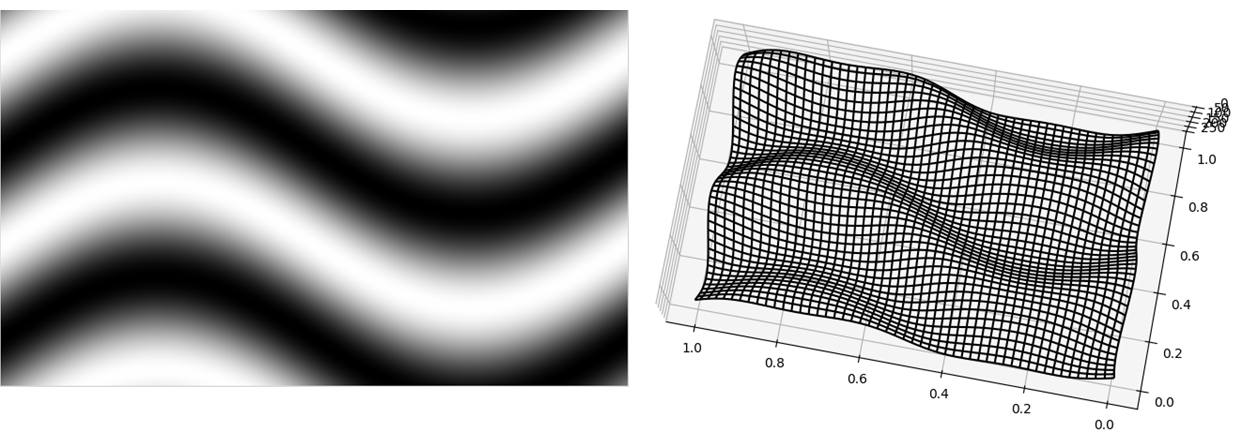


그림 13. 수평 수직 Sine Wave

* 동심 Sine Wave`

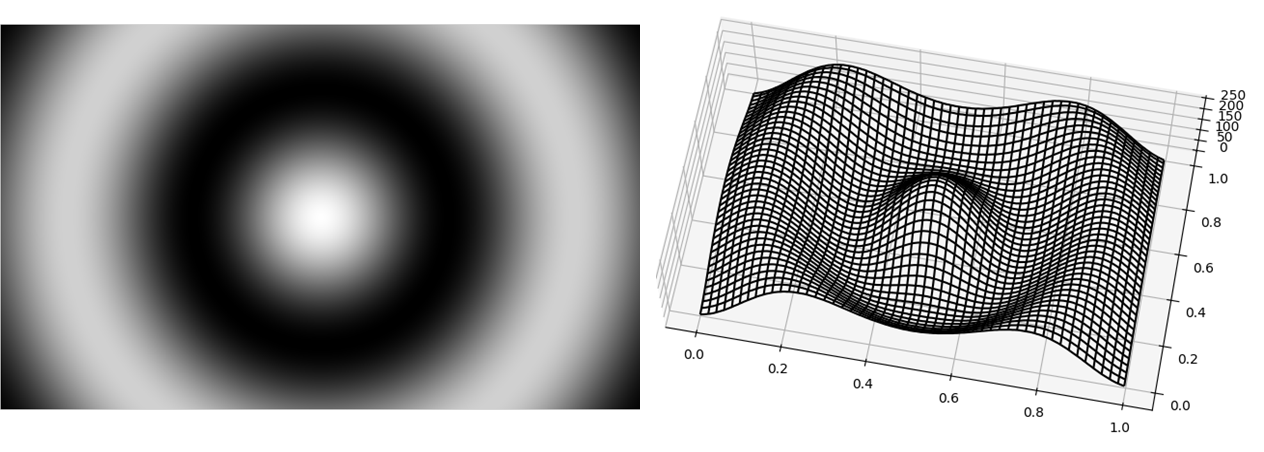


그림 14. 동심 Sine Wave

* 2차함수 기반 상승, 하강 물결 문양

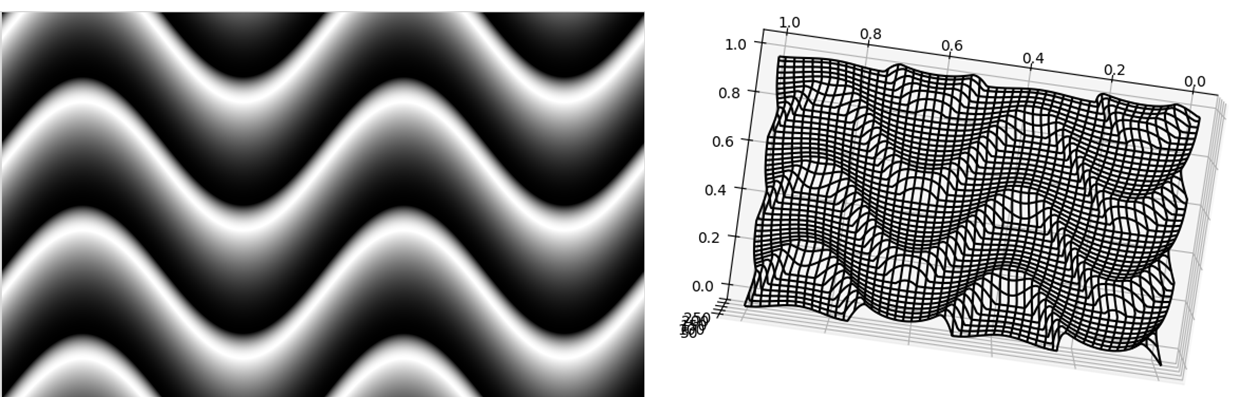


그림 15. 2차 함수 기반 상승, 하강 물결 문양

* 다각형 구조물

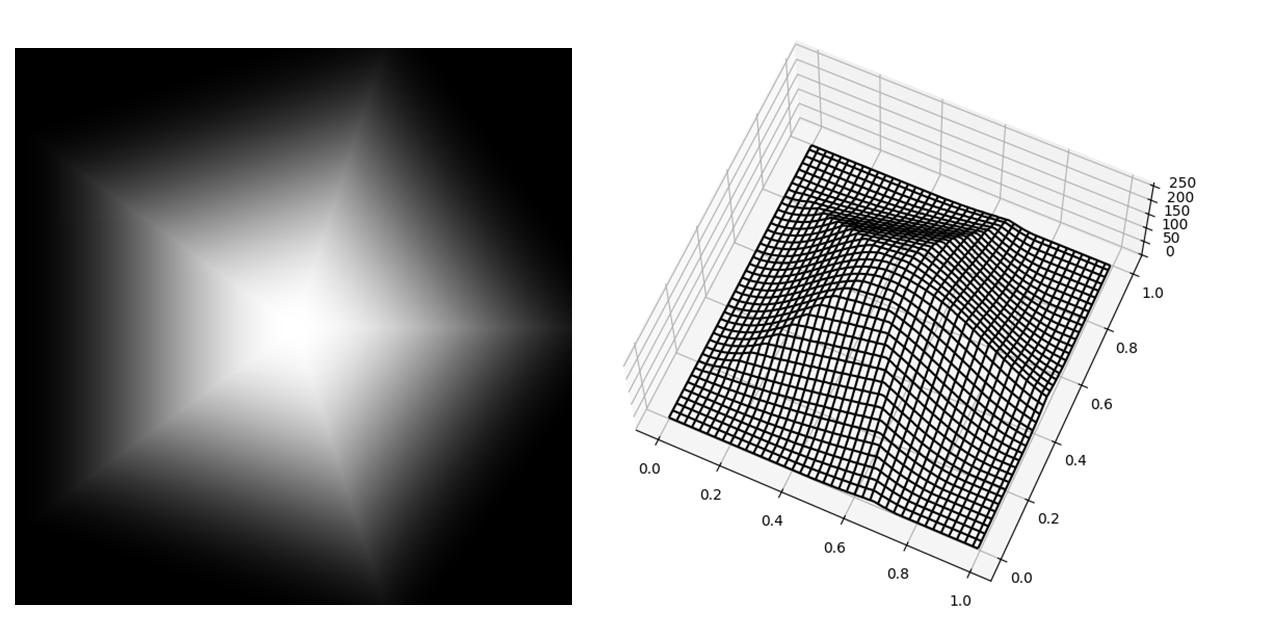


그림 16. 다각형 구조물의 예. 기본 도형: 5 각형, Image profile : Sine

* 다각형 구조물 Array

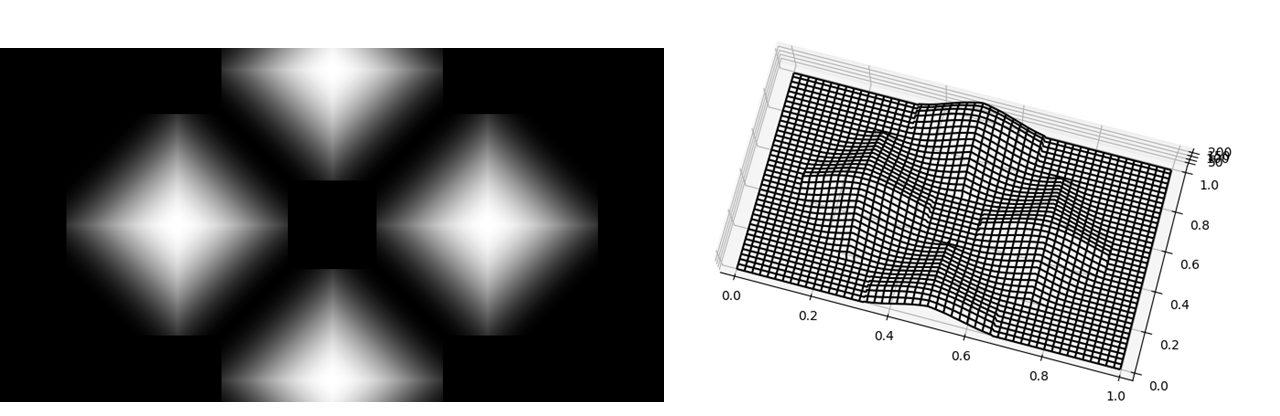


그림 17. 다각형 구조물 Array

* File 기반 생성

생성용 File 기반으로 Image 를 합성하는 방법이다. 아래의 그림은 생성 용 File 의 예이다(4\_overlapping\_sinusoid.gen). File 의 Type 은 “.gen”이다.

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

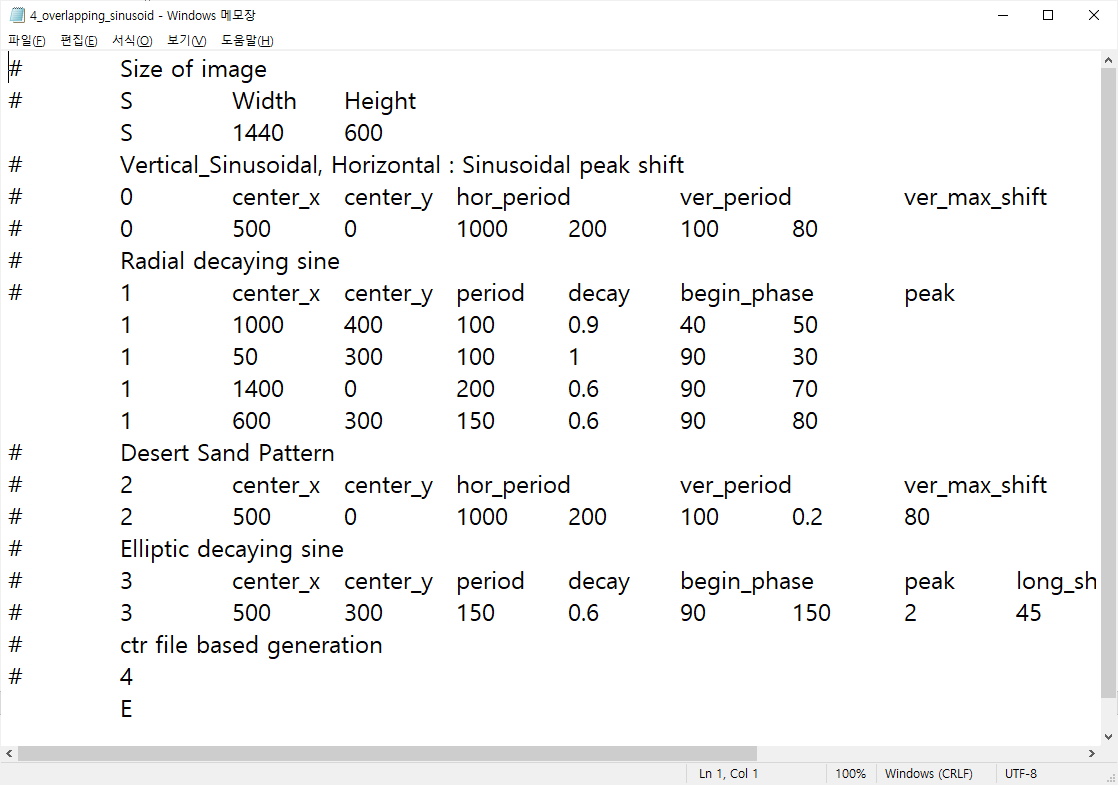


그림 18. Image 합성을 위한 생성 File 의 예

아래의 그림은 이 생성용 File 을 기반으로 생성된 Image 이다.

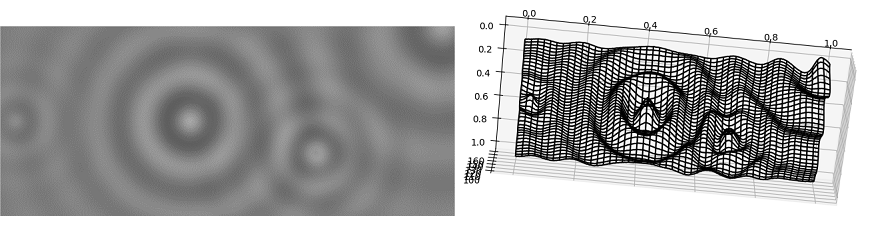


그림 19. 앞의 Image 생성 File로 생성된 Image



현재 화면의 위에 새로운 Image를 추가한다. 이를 Clip Image라고 부른다. 추가한 Clip Image는 마우스 오른쪽 버튼을 누른 채 Drag 하여 위치를 이동할 수 있다.



그림 20. 클립 Image를 더한 Image의 예



Clip Image 의 크기를 변경한다.



현재의 화면을 한 개의 Image 로 저장한다. Image 저장시 각 화소에 중첩된 Image 의 값을 모두 더하여 새로운 화소값을 생성한다. 각 화소는 16 bit이며 32768(215)을 level 0 로 하고 65535(216 – 1) 이 최대 값, 0 이 최소값 (-32768)에 해당하도록 Mapping 된다. Image 의 합성에 필요한 각 Clip Image 들은 이 같은 조건에서 적절한 합성이 이루어질 수 있도록 주의를 기울여야 한다.

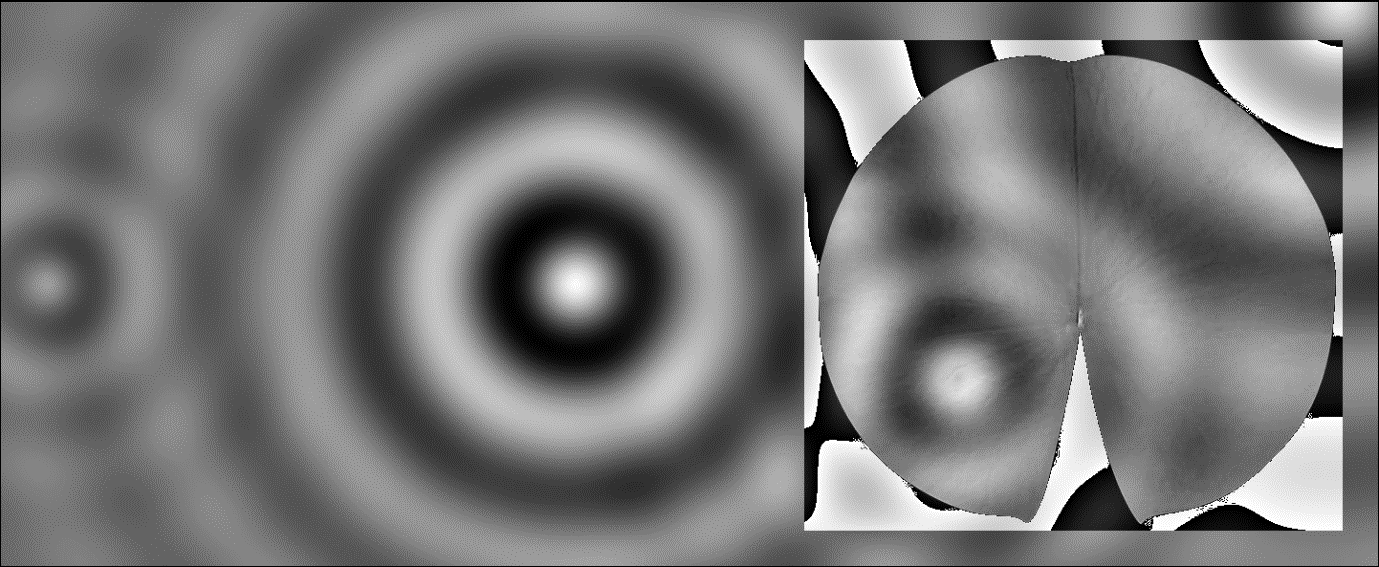


그림 22. 화면의 모든 Image 를 더하여 저장한 Image

 : Image의 수직방향으로 등 간격 절단면의 Profile을 생성해낸다.

아래와 같은 화면에서 조립용 Panel의 생성에 필요한 변수를 입력한다. 이들은 배수, 재료의 두께, Panel 간의 간격이다. 배수는 원 Image와 만들고자 하는 목표물의 크기 비율이다. 예를 들어 2로 한다면 원래의 Image에 대비하여 가로/세로를 각각 2배 크기로 만들기 위한 Panel들을 생성하도록 Profile을 조절한다.

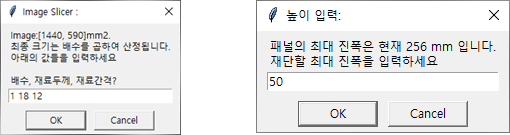


그림 23. Panel 의 Profile을 생성하기 위한 입력 창.

. : Slicing 된 단면에 기반하여 각각의 Panel의 절단용 CAD 도면을 생성한다. CAD 도면은 Woodmeister에 호환된다. 도면은 생성되는 모든 절단 부품의 개별 도면과 이들을 통합하여 가공할 수 있는 통합 도면의 2가지 종류가 생성된다. 커다란 원판을 사용하여 가공 시 후자의 도면을 사용한 가공이 편리하다.

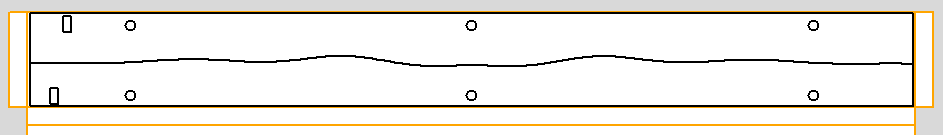


그림 24. 생성된 Panel Sample

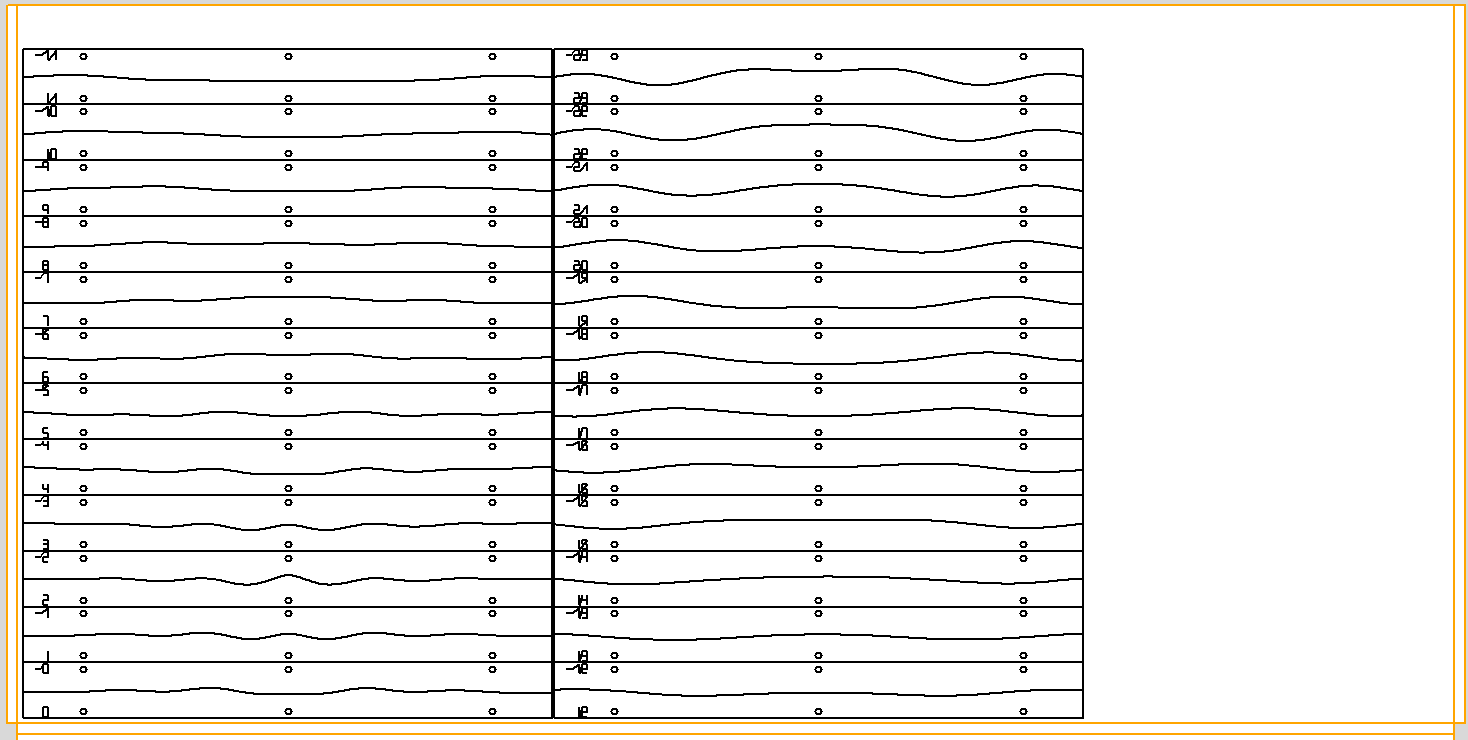


그림 25. Panel 통합 가공을 위하여 생성된 Design 도면

 : 가공된 Panel들을 등 간격으로 배열할 때 도움이 되는 Spacer 의 형태를 생성한다.

이 버튼을 누르면 아래와 같은 입력창이 떠오른다.

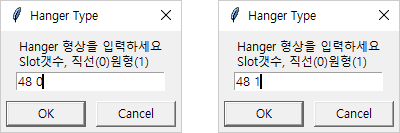
좌측의 창에서 Slot 의 개수 및 Hanger 의 형태를 입력한다. 왼쪽은 직선형의 Hanger를 지정하는 경우이며, 오른쪽은 원형의 Hanger를 지정하는 경우이다.

그림 26. Hanger 형태 입력 창. 왼쪽 : 직선형의 Hanger 지정 , 오른쪽 : 원형의 Hanger 지정

직선 형태 Spacer 의 형상은 아래의 그림과 같다. 제작된 패널들을 등 간격으로 끼워 넣을 수 있는 Slot들을 가지고 있다.



그림 27. Panel을 등 간격으로 배치하기 위한 Spacer 의 형태

Hanger 의 형상을 원형으로 지정한 경우 아래 그림의 왼쪽과 같은 입력 창이 떠오르면 여기에서 반경 및 Slot 의 폭을 입력한다. 이 같은 Data 를 기반으로 생성된 Hanger의 형상은 그림의 오른쪽과 같다. Slot의 폭은 입력한 Data를 기준으로 하며 Slot 의 깊이는 10mm로 정하여 생성된다.

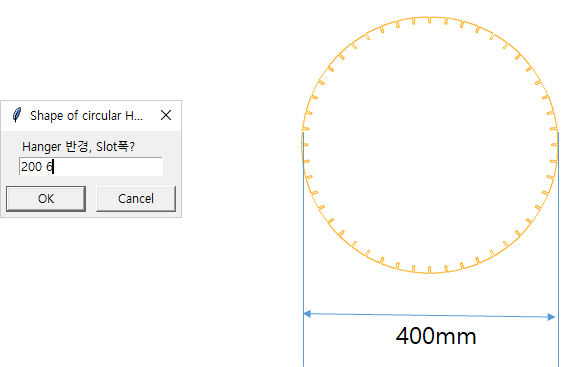


그림 28. 원형 Hanger의 Data 입력 창(왼쪽) 및 생성된 Hanger 의 모습(오른쪽)

\직선형, 원형 Hanger 모두 hanger 의 형상은 Point sequence file (“.pts”)로 저장한다.

 : Reserve된 Button

# Reference

## Video

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

[](https://www.youtube.com/watch?v=O9kDwJ-cUOQ&t=58s)

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

## Woodmeister 사용법

Manual:

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

## Youtube: [https://www.youtube.com/playlist?list=PLHzQ8JkLY6\_d3xQi7onElSgiXFULc0cMK](https://www.youtube.com/playlist?list=PLHzQ8JkLY6_d3xQi7onElSgiXFULc0cMK#)

## 목공과 Coding 의 만남

Document:

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

Youtube: <https://www.youtube.com/playlist?list=PLHzQ8JkLY6_fipJYY3eOBcDbp1lV-8dA0>

[](https://www.youtube.com/playlist?list=PLHzQ8JkLY6_fipJYY3eOBcDbp1lV-8dA0)