

Agate Evolver

Wojciech Ciszewski Alicja Dutkiewicz
Stanisław Kurdziałek

Opiekun: Piotr Szymczak

19.02.2019

1 Wstęp

Niniejszy projekt powstał w ramach Hackathonu 2.0 Wydziału Fizyki UW. Jego celem jest przetestowanie prostego modelu powstawania agatów poprzez zasymulowanie w oparciu o niego powstawania kolejnych warstw minerału.

Końcowym efektem naszej pracy jest *Agate evolver*, który produkuje strukturę warstw dla danej początkowej geometrii agatu w dwóch i, w niektórych przypadkach, trzech wymiarach. Przetestowaliśmy go na rzeczywistych agatach, aby sprawdzić, czy numerycznie wygenerowana struktura pasm pokrywa się z rzeczywistą.

2 Opis modelu

Agaty to minerały złożone z mikrokryystalicznej krzemionki ułożonej w charakterystyczne warstwy. Agaty wypełniają zwykle pustki po pęcherzach gazowych uwiezionych w stygnącej lawie. Jedna z teorii ich powstawania mówi, że kolejne warstwy krzemionki osadzały się na ściankach takiej pustki, a kolorowe paski są odzwierciedleniem różnych w składzie chemicznym roztworu na przestrzeni setek tysięcy lat, w czasie których tworzył się agat.

Szukaliśmy kształtów poszczególnych warstw agatu zakładając, że wzrost (w kierunku od ścianek do środka buły agatowej) odbywa się ze stałą prędkością (czyli nowa warstwa składa się z punktów będących w ustalonej odległości od warstwy poprzedniej). Taki wzrost ma naturalną tendencję do generowania „ostrych kantów”. Takie punkty z nieciągłą pochodną są często widoczne w agatach, co jest argumentem za tym, że ten prosty model wzrostu może być w tym przypadku właściwy.

2.1 Model 2D

W modelu 2D traktowaliśmy kolejne warstwy jako płaskie krzywe zamknięte, którą można przybliżyć wielokątami. Wykorzystaliśmy klasy `Polygon` i `MultiPolygon` z modułu `geometry` biblioteki `shapely`.

Stworzyliśmy funkcję `evolve_agat` pozwalającą wygenerować przekrój agatu o zadanym konturze. Funkcja przyjmuje następujące argumenty:

```
first_layer
    lista wierzchołków zewnętrznej warstwy [(x1, y1), (x2, y2), ...]

layer_width
    odległość pomiędzy kolejnymi warstwami. Domyślnie 0.05

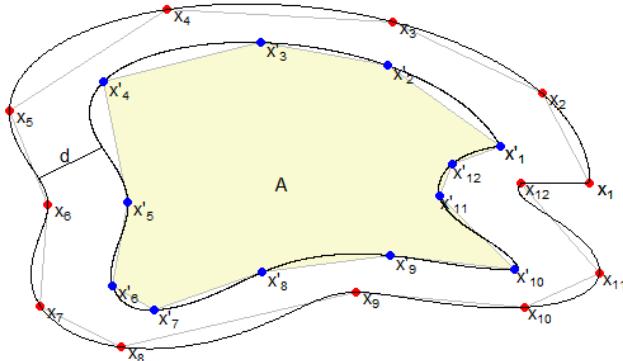
min_area
    powierzchnia wnętrza agatu. Domyślnie 0.001

N_apexes
    maksymalna liczba wierzchołków wielokątów reprezentujących kolejne warstwy. Jeśli równa -1 (domyślnie), to tyle samo co w first_layer

fig
    obiekt klasy Figure z biblioteki matplotlib, na którym ma zostać narysowany wynik symulacji. Domyślnie przyjmuje wartość None i rysunek jest wyświetlany po wywołaniu funkcji
```

Ewolucję schematycznie przedstawia rysunek 1. W każdym kroku ewolucji za pomocą metody `buffer` znajdowany był zbiór $N = N_{\text{apexes}}$ punktów oddległych o `layer_width` od aktualnej warstwy $\{[(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)]\}$ i znajdujących się wewnątrz wielokąta — te punkty tworzyły kolejną warstwę $\{[(x'_1, y'_1), (x'_2, y'_2), \dots, (x'_N, y'_N)]\}$. Jej wnętrze A kolorowano na jeden z losowo wybranych kolorów. Następnie przypisywano $x_i \leftarrow x'_i$, $y_i \leftarrow y'_i$ i rozpoczynał się kolejny krok. Ewolucję kontynuowano, aż powierzchnia wielokąta A reprezentująca aktualną warstwę była mniejsza niż `min_area`.

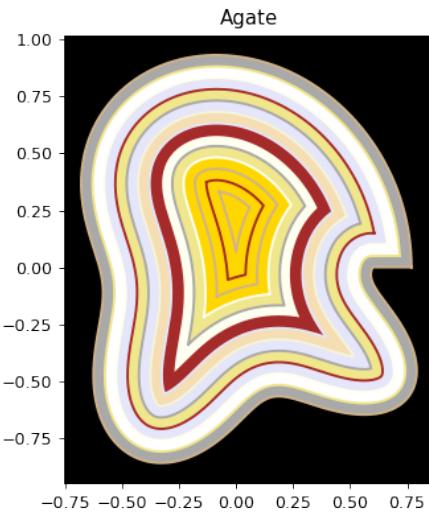
Rysunek 1: Ilustracja kroku ewolucji 2D



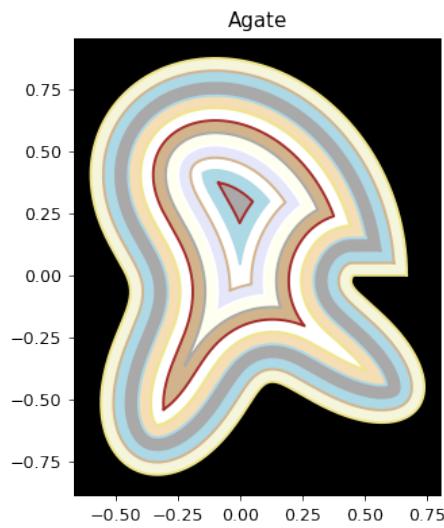
2.1.1 Generowanie konturów

Sztuczne agaty generowaliśmy za pomocą funkcji $r(\varphi) = \frac{1}{1+\alpha}(1 + \frac{\alpha}{1+\alpha} \sin(\beta\varphi)^2)$. Przykłady można zobaczyć na rysunkach 2, 3 i 4.

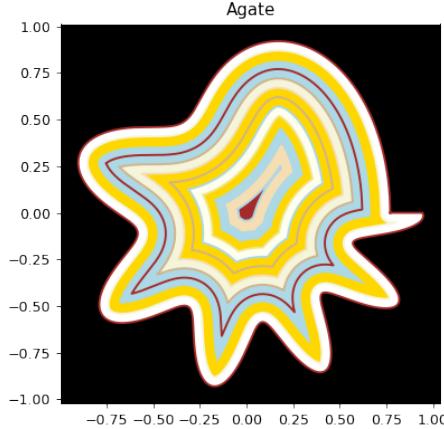
Rysunek 2: `evolve_agat(polar_curve(modulation = 0.3, freq = 2/3))`



Rysunek 3: `evolve_agat(polar_curve(modulation = 0.5, freq = 2/3))`



Rysunek 4: `evolve_agat(polar_curve(modulation = 0.3, freq = 1))`



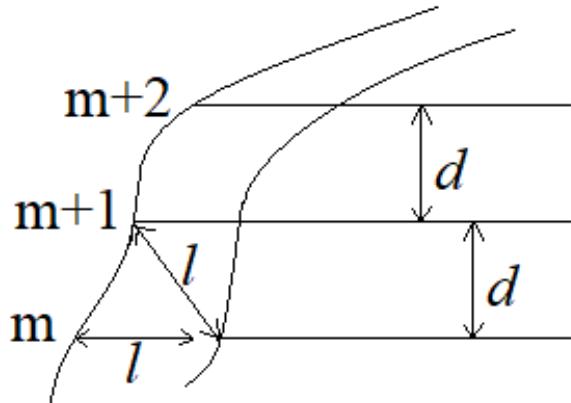
2.2 Model 3D

Rzeczywiste agaty należy traktować jak trójwymiarowe bryły — ewolucja dwuwymiarowych warstw odpowiada rzeczywiście jedynie w przypadku, gdy agat jest wystarczająco długi, a jego przekrój poprzeczny — wolno zmienny. Rozważmy dowolną zamkniętą, dwuwymiarową powierzchnię wypukłą Γ . W przybliżeniu możemy taką powierzchnię traktować jak zbiór dwuwymiarowych zamkniętych konturów będących kolejnymi przekrojami poprzecznymi danej powierzchni oddalonymi od siebie o pewną stałą odległość d . Przekroje takie można z kolei przybliżać wielokątami, jak w przypadku 2D. Pojedyncza warstwa trójwymiarowego agatu jest więc przechowywana w pamięci programu jako lista obiektów klasy `MultiPolygon`. Dodatkowo przechowywana jest odległość d między kolejnymi przekrojami poprzecznymi.

Funkcję generującą kolejną warstwę trójwymiarowego agatu stworzyliśmy w oparciu o dwuwymiarową wersję. Kluczowym było spostrzeżenie, że obszar o jaki urośnie dany przekrój poprzeczny jest sumą wkładów od jego własnego przyrostu i przyrostu sąsiednich przekrojów. Niech l będzie odlegością o jaką narasta warstwa w pojedynczym kroku (`layer_width` w programie). Założymy, że w pamięci przechowywanych jest n przekrojów każdej warstwy numerowanych liczbami całkowitymi $0, 1, 2, \dots, n - 1$. Aby znaleźć kolejną warstwę należy znaleźć jej wszystkie przekroje. Wkład przekroju o numerze k danej warstwy do przekroju o tym samym numerze kolejnej warstwy wyznaczamy używając stosownej funkcji dla przypadku dwuwymiarowego. Wkład przekroju o numerze k do przekroju o numerze m wyznaczamy ewoluując przekrój k o odpowiednio zmniejszoną odległość l' . Przekroje o numerach k i m oddalone są od siebie o $D = |k - m| \cdot d$. Stąd jeśli punkt X należy do przekroju k to zbiór punktów leżących w przekroju m oddalonych od X o nie więcej niż l jest zbiorem punktów oddalonych od rzutu punktu X na płaszczyznę przekroju m o najwyżej $\sqrt{l^2 - D^2}$ (sytuację tę ilustruje rysunek 5). Oznacza to, że biorąc $l' = \sqrt{l^2 - D^2}$

otrzymujemy odległość o jaką należy przeewoluować przekrój o numerze k (używając odpowiedniej funkcji z przypadku 2D) aby uzyskać wkład do przekroju o numerze m . Jeśli $l^2 - D^2 < 0$, to dwa przekroje są od siebie na tyle odległe, że żaden z nich nie ma wpływu na wzrost drugiego w pojedynczym kroku.

Rysunek 5: Wkłady od sąsiednich przekrojów



Pojedynczy krok ewolucji trójwymiarowej warstwy realizowany jest przez funkcję `make_new_layer_3D`. Funkcja ta przyjmuje następujące argumenty:

```
old_layer
    lista obiektów klasy MultiPolygon — przekroje warstwy, która ma być
    ewoluowana

layer_width
    głębokość narastania warstwy w jednym kroku

l_distance
    odległość między dwoma sąsiadującymi przekrojami
```

Funkcja ta zwraca obiekt `new_layer` — listę przekrojów nowej warstwy. Każdy z tych przekrojów uzyskiwany jest jako przecięcie odpowiednio przeewoluowanych sąsiednich przekrojów przy użyciu metody `intersection`, która zwraca przecięcie obiektu klasy `MultiPolygon` z innym obiektem tej klasy przekazywanym jako argument. Na bazie tej funkcji stworzona została funkcja `make_N_layers_3D`, która poprzez wielokrotne wywołanie funkcji `new_layer` tworzy N nowych warstw. Funkcja ta zwraca listę utworzonych warstw. Do wizualizacji wzrostu agatów służy funkcja `draw_crossection`, która rysuje przekrój o zadanym numerze agatu przekazywanego do funkcji jako lista warstw.

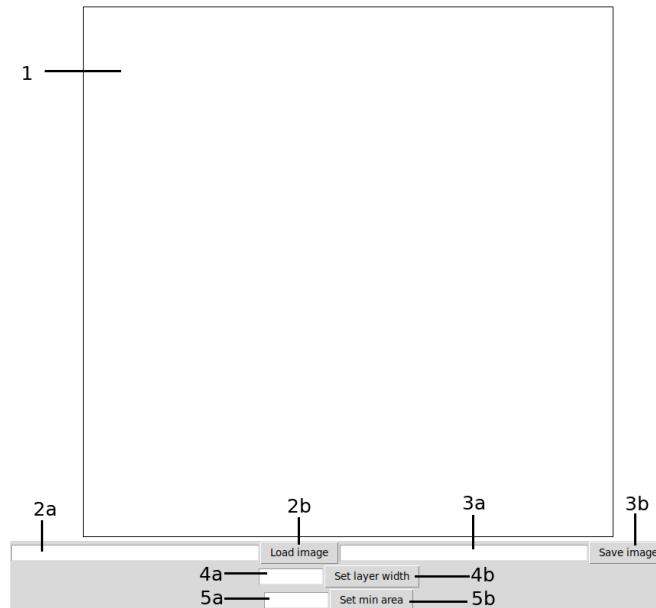
3 Interfejs użytkownika

3.1 Część 2D

Okno programu przedstawione jest na rysunku 6.

Rysunek 6: Okno programu - wersja 2D

Agate evolver

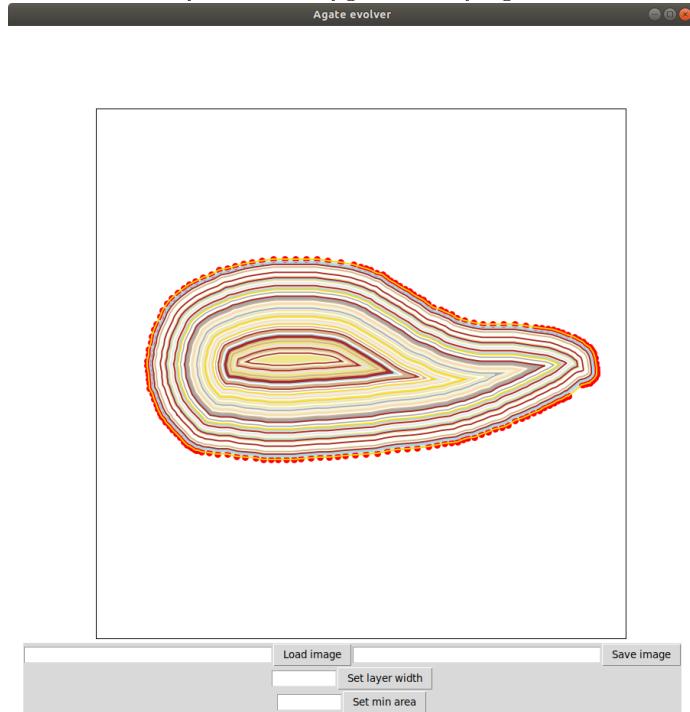


Składa się z następujących części:

1. Obszar rysowania

Użytkownik może zaznaczyć kontur agatu przytrzymując lewy przycisk myszy. W momencie puszczenia przycisku zostanie wygenerowany agat o zaznaczonym konturze. Gdy użytkownik zaczyna rysować nowy kontur, poprzedni rysunek jest usuwany. Przykład wygenerowanego agatu znajduje się na rysunku 7.

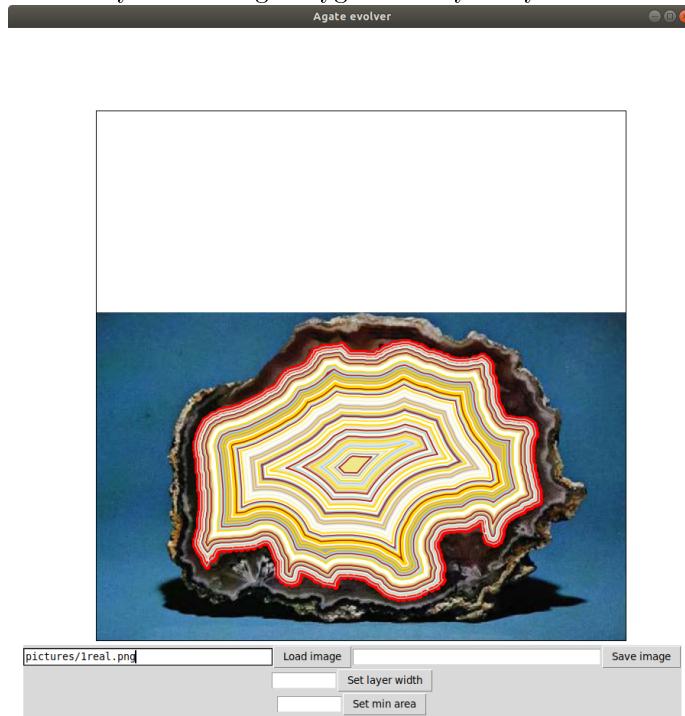
Rysunek 7: Wygenerowany agat



2. Wczytywanie obrazu

Program umożliwia wczytanie obrazu w celu porównania wyników modelu z prawdziwymi agatami. Należy podać w polu **2a** ścieżkę do pliku z obrazem w formacie PNG (można pominąć rozszerzenie), a następnie nacisnąć przycisk **2b**. Obraz zostanie wyświetlony w obszarze rysowania. Następnie można zaznaczyć w dowolnym miejscu kontur, a wygenerowany agat zostanie wyświetlony na wczytanym obrazie. Przykład takiego użycia znajduje się na rysunku 8.

Rysunek 8: Agat wygenerowany na rysunku



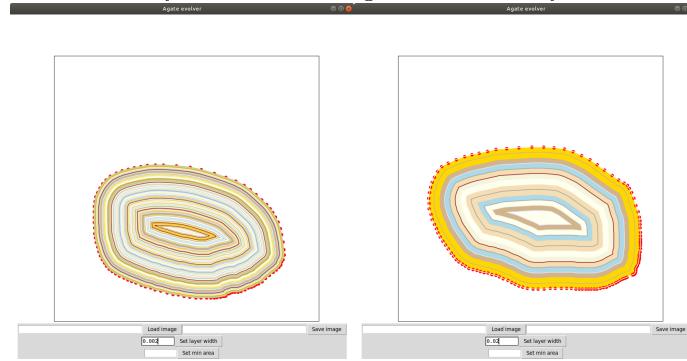
3. Zapisywanie obrazu

Wygenerowany agat można zapisać jako obraz PNG. W tym celu należy podać w polu **3a** nazwę wynikowego pliku, a następnie nacisnąć przycisk **3b**.

4. Grubość warstwy

Użytkownik może ustawić grubość warstwy generowanego agatu, wpisując wartość w polu **4a**, a następnie naciskając przycisk **4b** (dla odniesienia obszaru rysowania ma wymiary 1x1). Domyślna wartość: 0.005. Przykład dwóch różnych grubości znajduje się na rysunku 9.

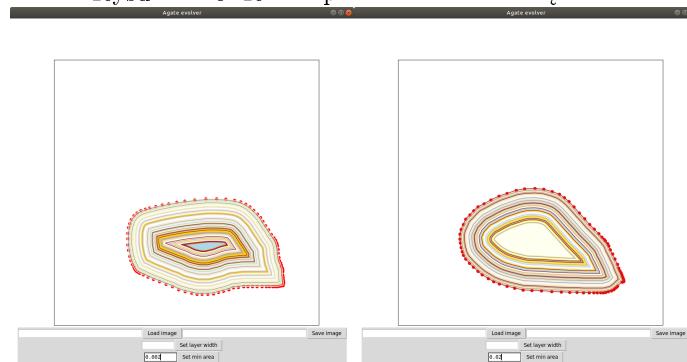
Rysunek 9: Różne grubości warstwy



5. Powierzchnia wewnętrzna

Użytkownik może ustawić minimalną powierzchnię, dla której będą generowane warstwy, wpisując wartość w polu **5a**, a następnie naciskając przycisk **5b** (dla odniesienia obszar rysowania ma wymiary 1x1). Domyślna wartość: 0.001. Parametr ten określa jak duża powinna być jednolita powierzchnia w środku generowanego agatu. Przykład dwóch różnych powierzchni znajduje się na rysunku 10.

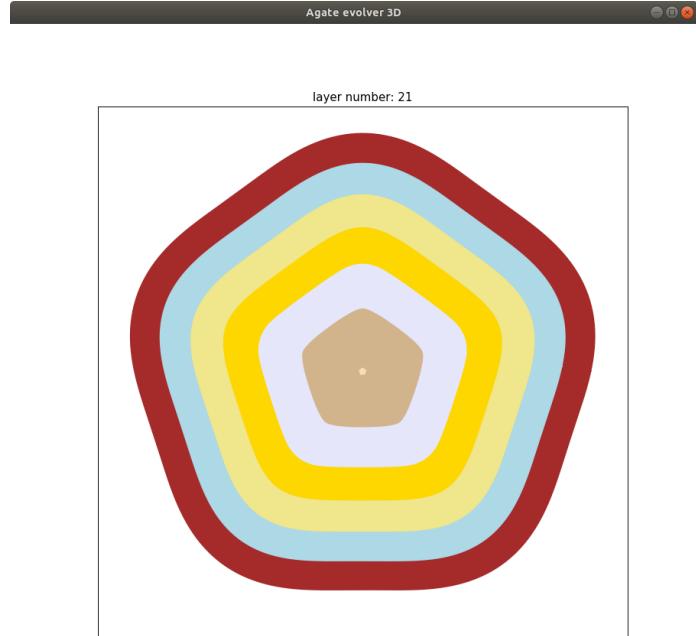
Rysunek 10: Różne powierzchnie wewnętrzne



3.2 Część 3D

Okno programu jest przedstawione na rysunku 11.

Rysunek 11: Okno programu - wersja 3D



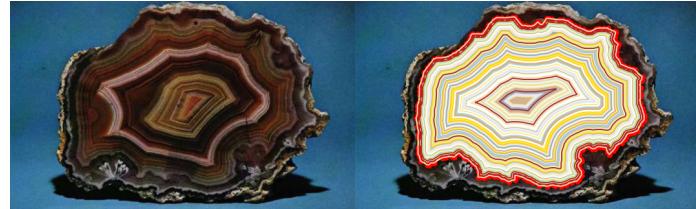
Zmieniać wyświetlaną warstwę można za pomocą kółka myszy.

4 Podsumowanie

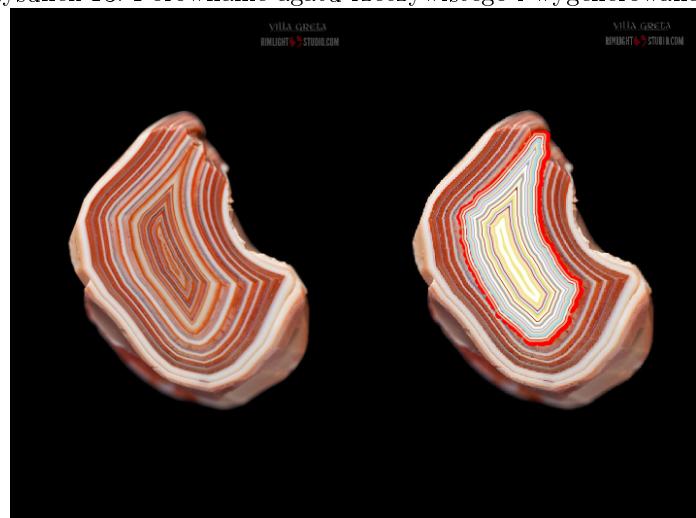
Mimo bardzo prostych założeń, warstwy wygenerowane w modelu 2D na podstawie konturów przekrojów prawdziwych agatów wyglądają realistycznie i w wielu przypadkach są bardzo podobne do rzeczywistych, co widać na rysunkach 12, 13, 14, 15 i 16. W sytuacjach, w których przewidywanie modelu odbiegają od budowy agatu, różnice mogą mieć dwie główne przyczyny:

- Przekrój wykonano w pobliżu brzegu agatu, przez co warstwy narastające na nim spowodowały zmiany nieobejmowane przez model 2D
- Przestrzeń, którą agat wypełniał podczas powstawania, nie miała postaci pustej bańki, lecz zawierała wewnętrzne dodatkowe struktury, a jej przekrój nie był jednospójny, przez co warstwy narastały jednocześnie z wielu stron

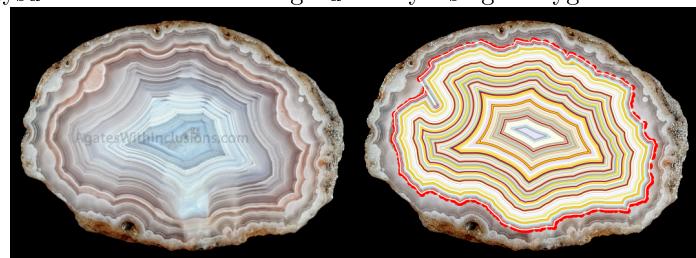
Rysunek 12: Porównanie agatu rzeczywistego i wygenerowanego



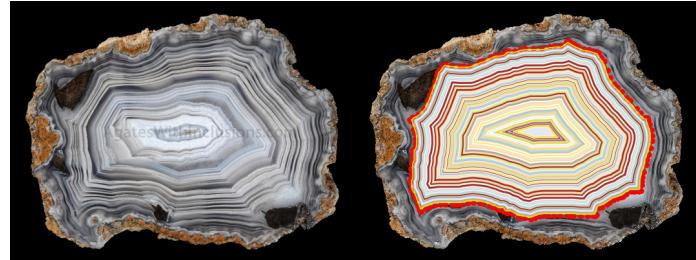
Rysunek 13: Porównanie agatu rzeczywistego i wygenerowanego



Rysunek 14: Porównanie agatu rzeczywistego i wygenerowanego



Rysunek 15: Porównanie agatu rzeczywistego i wygenerowanego



Rysunek 16: Porównanie agatu rzeczywistego i wygenerowanego

