

# **Automatyczne generowanie obrazów typu Thread Art z zastosowaniem wybranych algorytmów sztucznej inteligencji**

Diplomant: inż. Michał Suliborski (239713)

Promotor: dr inż. Witold Marańda

Politechnika Łódzka, Instytut Informatyki  
Wydział Fizyki Technicznej, Informatyki i Matematyki Stosowanej  
Kierunek: Informatyka Stosowana  
Specjalizacja: Inżynieria Oprogramowania i Uczenie Maszynowe

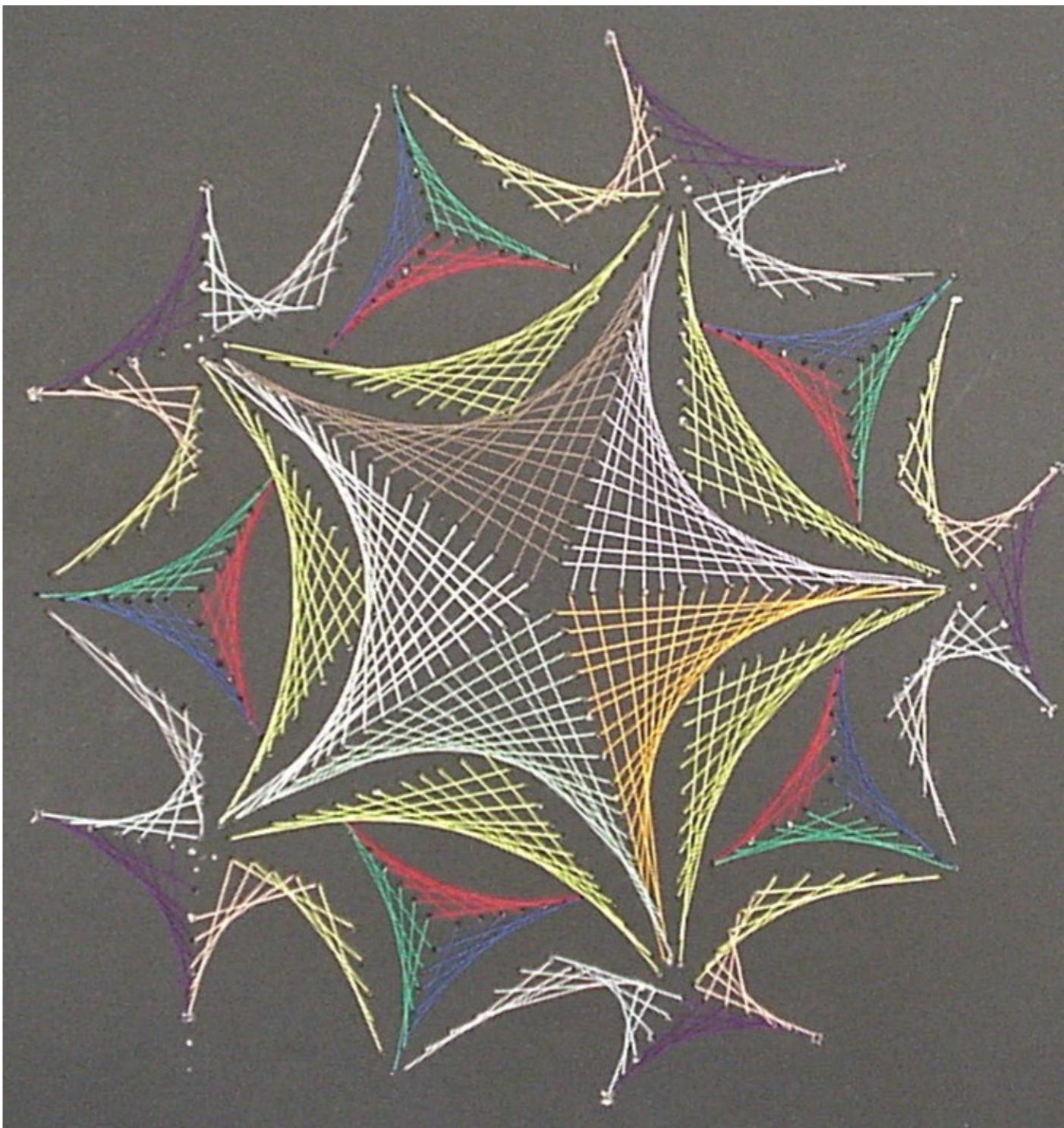
Praca dyplomowa magisterska

# Thread Art

Inaczej również  
String Art lub Pin and Thread Art

Technika powstała pod koniec XIX wieku.

Popularność zyskała w latach 60.



Przykład obrazu typu Thread Art

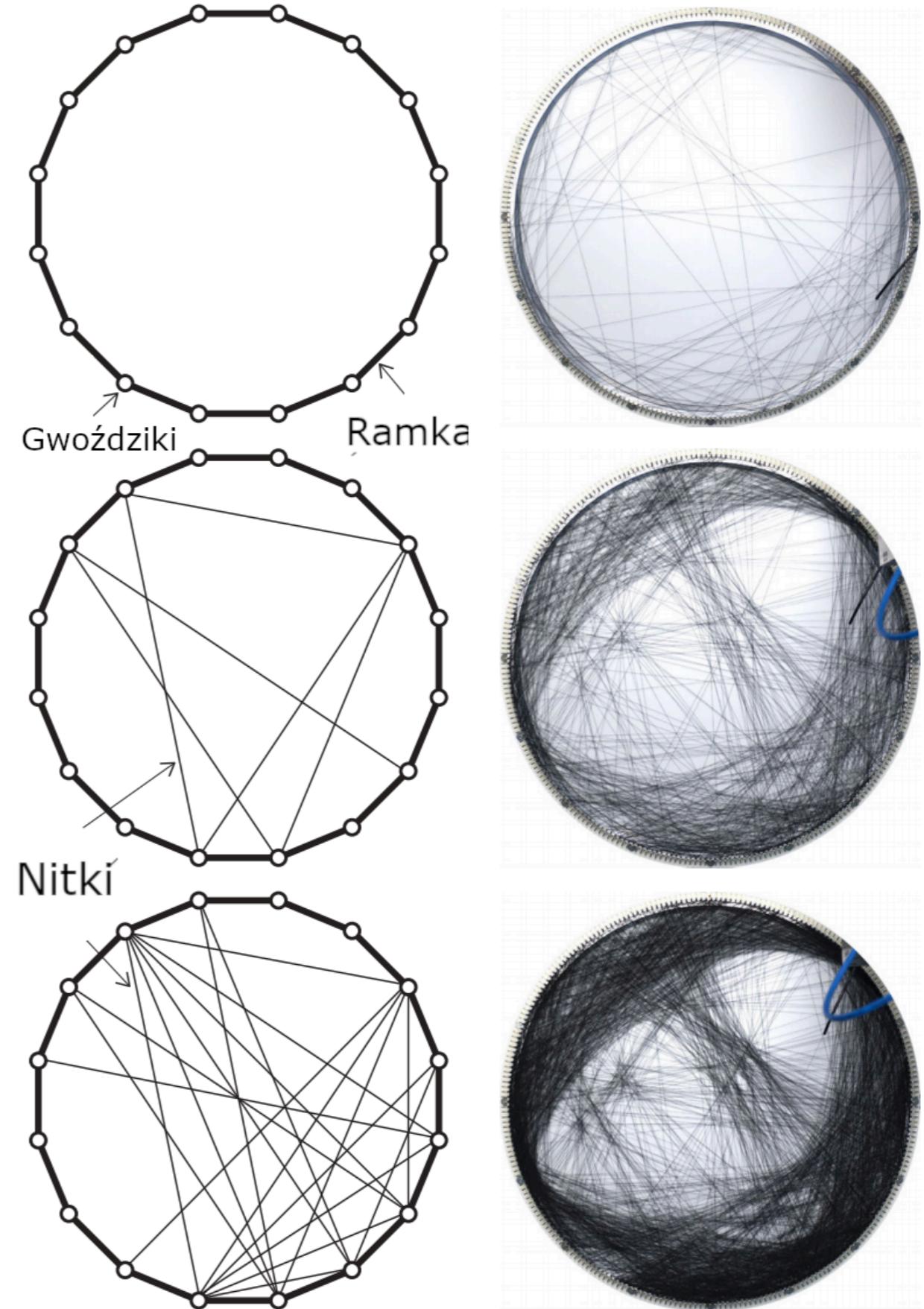
# Computational Thread Art

Twórcą komputerowej odmiany Thread Art jest Petros Vrellis [10], grecki magister inżynier elektryk i magister nauk o sztuce.

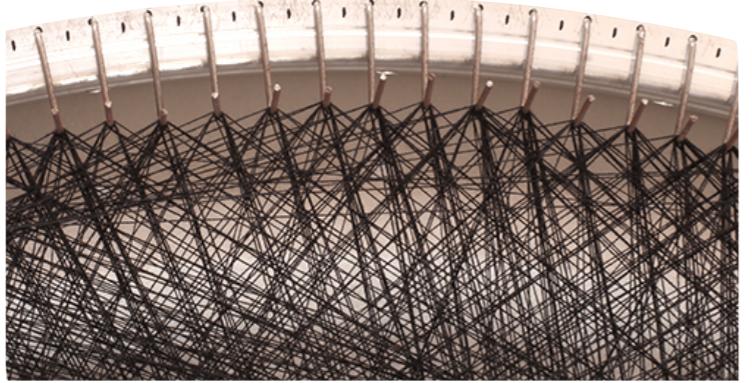


Petros Vrellis

Zaprezentował  
ją w 2016 roku.



# Fizyczne przykłady obrazów Thread Art



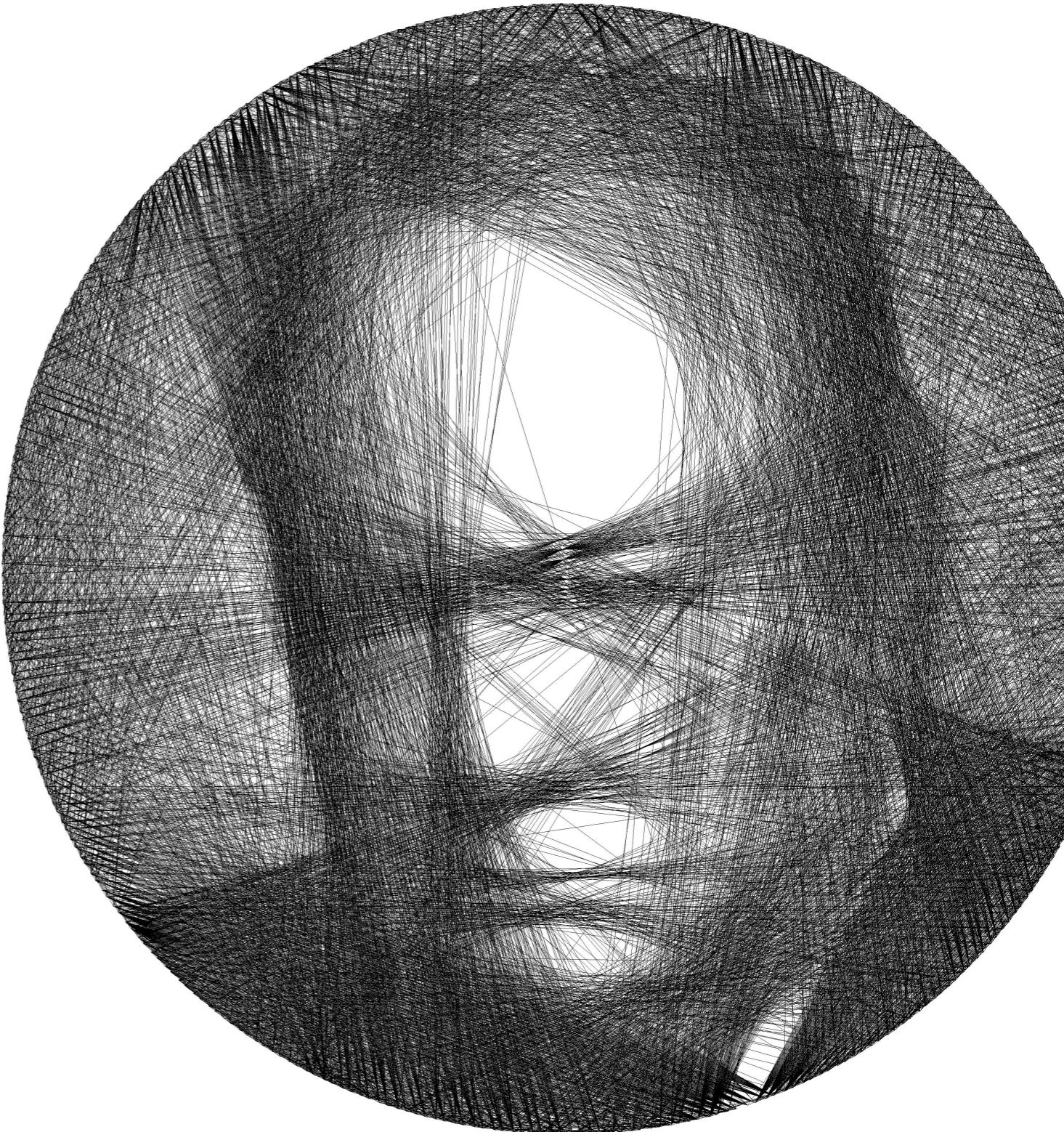
# Automatyczna generacja obrazów Thread Art

Wybór **obrazu wejściowego**

**Dobór parametrów**  
**(Ilość gwoździ, szerokość linii,  
ilość iteracji etc.)**

Znalezienie **najlepszej drogi** dla  
nitki za pomocą algorytmu

**Wygenerowanie obrazu  
wyjściowego**



Thread Art portrait of Winston Churchill  
[<https://github.com/callummcdougall/computational-thread-art>]

# Ogólny algorytm tworzenia obrazu Thread Art

Wybrany zostaje losowy gwóźdź

Generowane są wszystkie możliwe linie zaczynające się od tego gwoździa

Znajdowana jest linia z najlepszym wynikiem funkcji dopasującej

Linia zostaje usunięta z obrazu

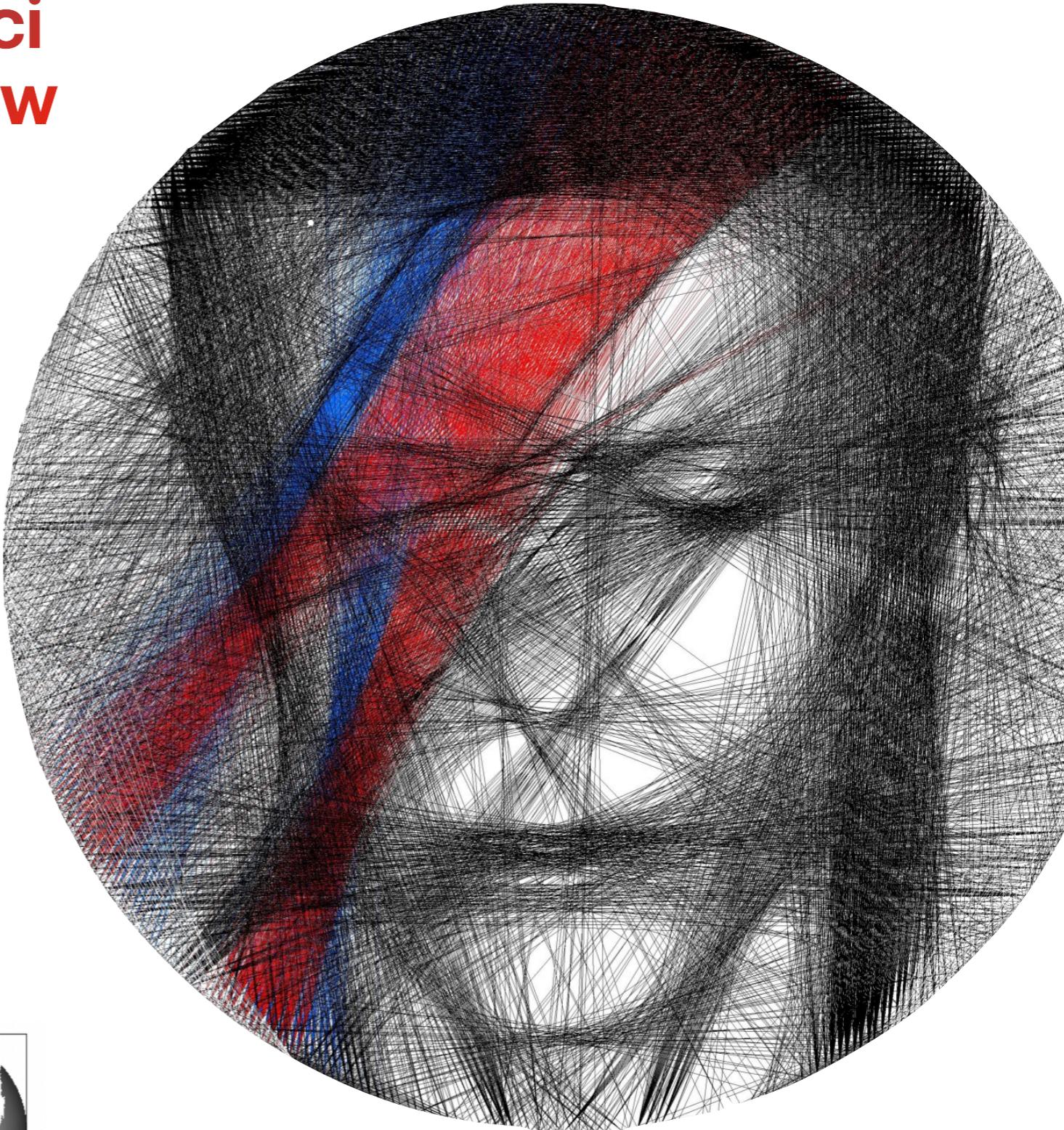
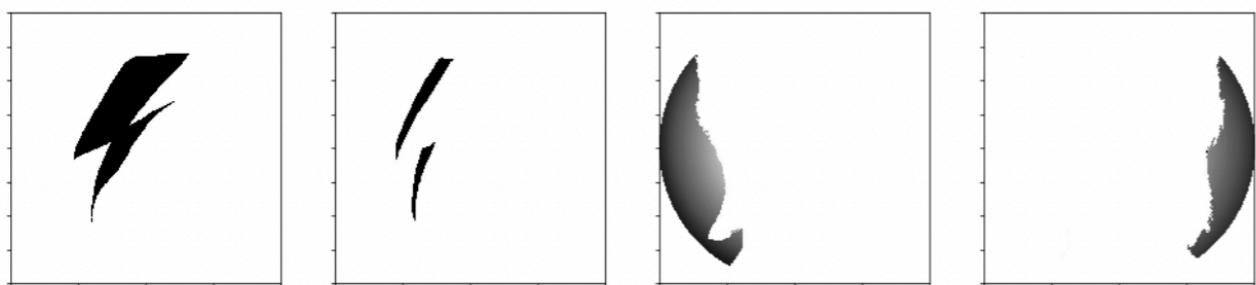
Na podstawie zmodyfikowanego obrazu i nowej pozycji startowej algorytm jest powtarzany

# Techniki stosowane w celu poprawy jakości generowanych obrazów

Maska z wagami [6]

Algorytm samokorygujący się [1]

Powtarzalność drogi [5]



Thread Art portrait of David Bowie  
[<https://github.com/callummcdougall/computational-thread-art>]

# Problem badawczy

Dopasowanie optymalnych algorytmów Thread Art do różnych typów obrazów i ocena jakości wyników.

# Zakres i cele pracy

- Computer Art
- Algorytmika
- Sztuczna inteligencja
- Przetwarzanie obrazu
- Analiza danych
- Zbadanie technik tworzenia obrazów typu Thread Art
- Implementacja wybranych algorytmów
- Dobór algorytmu do rodzaju obrazu wejściowego
- Zaproponowanie autorskiej metody tworzenia takich obrazów
- Analiza porównawcza algorytmów

# Metoda badawcza

- Analiza aktualnego stanu wiedzy w dziedzinie tworzenia obrazów typu Thread Art
- Użycie technologii Python 3 [9] wraz z bibliotekami pomocniczymi takimi jak Pillow [2], NumPy [11], Matplotlib [12], w celu implementacji i analizy porównawczej istniejących metod z zaproponowaną autorską metodą tworzenia takich obrazów
- Dobór metod do rodzaju obrazu wejściowego
- Automatyczna i subiektywna ocena wyników oraz dyskusja rozwiązań

# Układ pracy

## Spis treści

<b>1 Wstęp</b>	<b>3</b>
1.1 Wprowadzenie do tematyki Sztuki Komputerowej . . . . .	3
1.2 Zakres i problematyka pracy . . . . .	3
1.3 Cele pracy . . . . .	3
1.4 Metoda badawcza . . . . .	3
1.5 Układ pracy . . . . .	3
<b>2 Problem tworzenia obrazów typu Thread Art</b>	<b>3</b>
2.1 Wstęp do zagadnienia generowania obrazów typu Thread Art . . . . .	3
2.2 Ogólna metoda generowania obrazu . . . . .	3
2.2.1 Dane wejściowe . . . . .	3
2.2.2 Ogólny opis algorytmu . . . . .	3
2.2.3 Dane wyjściowe . . . . .	3
2.3 Techniki stosowane w celu poprawy jakości wygenerowanych obrazów . . . . .	3
2.3.1 Maska z wagami . . . . .	3
2.3.2 Algorytm samokorygujący . . . . .	3
<b>3 Przegląd istniejących rozwiązań</b>	<b>3</b>
3.1 Rozwiązanie 1 . . . . .	3
3.1.1 Opis rozwiązania . . . . .	3
3.1.2 Wady i zalety . . . . .	3
3.2 Rozwiązanie 2 . . . . .	3
3.2.1 Opis rozwiązania . . . . .	3
3.2.2 Wady i zalety . . . . .	3
<b>4 Autorska metoda generowania obrazów</b>	<b>3</b>
4.1 Użyte technologie . . . . .	3
4.2 Opis algorytmu . . . . .	3
4.3 Użyte techniki optymalizacji i poprawy jakości generowanych obrazów . . . . .	3
<b>5 Implementacja wybranych metod generowania obrazów</b>	<b>3</b>
5.1 Opis działania programu . . . . .	3
<b>6 Analiza porównawcza wybranych metod generowania obrazów</b>	<b>3</b>
6.1 Wskaźniki oceny podobieństwa . . . . .	3
6.2 Opis zbioru obrazów wejściowych . . . . .	3
6.3 Porównanie z pomocą technik automatycznych . . . . .	3
6.4 Porównanie subiektywne . . . . .	3
<b>7 Podsumowanie</b>	<b>3</b>
7.1 Omówienie wyników . . . . .	3
7.2 Wnioski . . . . .	3
7.3 Możliwości rozwinięcia tematu pracy . . . . .	3
<b>8 Bibliografia</b>	<b>3</b>
<b>9 Spis rysunków</b>	<b>3</b>
<b>10 Spis tabel</b>	<b>3</b>

# Przegląd literatury

## **String art: Towards computational fabrication of string images. [1]**

Opis jednej z metod tworzenia obrazów typu thread Art (cyfrowych jak i fizycznych)

## **Wprowadzenie do Algorytmów. [3]**

Opis i implementacja algorytmów optymalizacyjnych

## **Automatic thread painting generation. [4]**

Opis jednej z metod tworzenia obrazów typu thread Art

## **Computational Thread Art [6]**

Opis jednej z metod tworzenia obrazów typu thread Art

## **ThreadTone [8]**

Opis jednej z metod tworzenia obrazów typu thread Art

## **A new way to knit (2016); Experimental / computational string art. [10]**

Ogólny opis techniki Thread Art

# Spodziewane rezultaty

- Opracowanie autorskiej metody tworzenia obrazów Thread Art z użyciem algorytmów sztucznej inteligencji zapewniającej co najmniej porównywalne rezultaty od metod istniejących
- Implementacja algorytmów
- Opracowanie wskaźnika podobieństwa obrazów Thread Art

# Możliwość kontynuacji

- Użycie technik nauki maszynowej do rozwiązania problemu
- Zbadanie wpływu generowania portretów Thread Art na automatyczne rozpoznawanie twarzy
- Stworzenie maszyny produkującej fizyczne egzemplarze uwzględniającej parametry wejściowe

# Bibliografia

- [1] Birsak, M., Rist, F., Wonka, P., & Musalski, P. (2018). String art: Towards computational fabrication of string images. *Computer Graphics Forum*, 37(2), 263–274. <http://doi.org/10.1111/cgf.13359>
- [2] Clark , A., & Contributors. (2010). Pillow documentation. Retrieved from <https://pillow.readthedocs.io/en/stable/>
- [3] Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., Diks, K., Malinowski, A., ... Rytter, W. (2021). Wprowadzenie do Algorytmów. Warszawa: Wydawnictwo Naukowe PWN SA.
- [4] Fang, X.-N., Liu, B., & Shamir, A. (2016). Automatic thread painting generation. *Communications in Information and Systems*, 16(4), 255–274. <http://doi.org/10.4310/cis.2016.v16.n4.a3>
- [5] I Wrote an Algorithm to Draw Portraits from Thread | Thread Art. (2021). YouTube. YouTube. Retrieved from <https://www.youtube.com/watch?v=UsbBSttajOs>
- [6] McDougall, C. (2020, September 11). Computational Thread Art. Retrieved from <https://medium.com/@cal.s.mcdougall/computational-thread-art-a9e46ecf34de>
- [7] Microsoft. (2015, April 29). Visual Studio Code - code editing software. Retrieved from <https://code.visualstudio.com/>
- [8] Scheepers, T. (2016, December 25). ThreadTone [web log]. Retrieved from <http://www.thevelop.nl/blog/2016-12-25/ThreadTone/>
- [9] Van Rossum, G., Drake, & L., F. (2009). Python 3. Retrieved from <https://www.python.org/>
- [10] Vrellis, P. (2016). A new way to knit (2016); Experimental / computational string art. Retrieved from <http://artof01.com/vrellis/works/knit.html>
- [11] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. *Nature* 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.
- [12] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. <http://doi.org/10.1109/mcse.2007.55>

# Dziękuję za uwagę