Algorytm genetyczny generowania obrazów

Wojciech Ciężobka, Paweł Świder

# Wstęp

Celem algorytmu jest wygenerowanie obrazu za pomocą określonej liczby różnokolorowych trójkątów jak najbardziej podobnego do obrazu źródłowego.

W efekcie uzyskane obrazy przypominają swoje oryginały. Np wygenerowana Mona Lisa wygląda następująco:



Rysunek 1 Wygenerowana Mona Lisa

# Algorytm

Zastosowany algorytm generyczny składa się z następujących faz:

1. Wygenerowanie nowej populacji na podstawie starej (pierwsza populacja jest generowana losowo) za pomocą odpowiednich mutacji.
2. Ewaluacja danej populacji poprzez wygenerowanie obrazów na podstawie DNA osobników.
3. Wybranie najlepszych osobników które posłużą do tworzenia nowej populacji.

Proces ten zostawał powtarzany wielokrotnie aż do osiągnięcia zadowalających wyników.

# Zawartość DNA

DNA służące do generowania obrazów zawiera następujące informacje dotyczące osobnika:

1. Ile trójkątów zawiera jego DNA
2. Wierzchołki każdego z trójkątów przechowywane jako liczby zmiennoprzecinkowe od 0 do 1
3. 3 liczby zmiennoprzecinkowe przyjmujące wartości od 0 do 1, które reprezentują kolor (w naszym przypadku używaliśmy przestrzeni barw RGB)

# Mutacje:

Przetestowaliśmy różne rodzaje mutacji dające różne rezultaty.

## Delikatna zmiana aktualnego DNA:

Mutacja polegała na losowej nieznacznej zmianie każdej wartości każdej liczby. Powodowało to że każdy trójkąt zmutowanego obrazu był nieco gdzie indziej oraz miał nieco inny kolor. Mutacja ta była nieefektywna ponieważ bardzo często więcej zmian było negatywnych niż pozytywnych i w efekcie proces ewolucji następował bardzo wolno.

## Losowa zmiana części trójkątów na inne:

Polegała na wybraniu losowo określonej liczby trójkątów z DNA a następnie zastąpienie ich nowymi, losowymi trójkątami. Metoda dała o wiele lepszy wynik niż metoda pierwsza ponieważ umożliwiała regularną poprawę generowanego obrazu, nawet jeśli zmiana była niewielka to mnogość takich zmian przyczyniała się do stopniowego dopasowania się generowanego obrazka do źródła.

# Generowanie obrazków i podobieństwo:

Generowanie obrazów polegało na naniesieniu na czarny obraz trójkątów w odpowiednich kolorach. W celu uniknięcia przykrywania kolorów w przypadku, kiedy w danym punkcie znajdowało się więcej niż 1 trójkąt to kolor w punkcie był średnią arytmetyczną kolorów trójkątów.

Podobieństwo obrazów powstało poprzez zsumowanie kwadratów różnic kolorystycznych pomiędzy pikselami.

Gdzie indeks p to indeks odpowiadający pikselowi, , to odpowiednio obrazek aktualnie generowany, oraz wzorcowy

# Wybrane hiper parametry modelu:

## Wielkość generowanego obrazku

Określał jak duże obrazki były generowane do porównania z obrazem źródłowym (który został dopasowany do obrazów generowanych). Dwukrotne zwiększenie obrazu powodowało czterokrotny wzrost czasu obliczeń – dlatego staraliśmy się trzymać obrazki dość małe (100x100). Tym bardziej że nasz zapis obrazków był wektorowy co pozwalało na późniejsze wygenerowanie większych obrazów.

Ilość trójkątów

Ilość trójkątów używanych do generowania obrazu – generalnie zwiększenie tego parametru powodowało polepszenie rezultatów jednak spowalniało algorytm oraz proces ewolucji. W efekcie dążyliśmy do użycia jak najmniejszej liczby trójkątów która pozwoli otrzymać satysfakcjonujący wynik.

## Populacja początkowa/po przeprowadzeniu selekcji

Określał ile najlepszych osobników przeżywało i brało udział w kolejnej epoce. Praktyka pokazała że branie tylko jednego – najlepszego osobnika daje zadowalające efekty – tym bardziej że bardzo często kolejnymi osobnikami z najlepszym dopasowaniem do zdjęcia były jego dzieci lub rodzic.

## Ilość dzieci generowana z jednego dziecka

Parametr odpowiedzialny za stan liczbowy populacji po wygenerowaniu nowych osobników. W toku symulacji miał w miarę mały wpływ na efekty. Z reguły przyjmował wartości pomiędzy 5 a 15. Jeżeli zachowywaliśmy większą ilość najlepszych osobników to zmniejszaliśmy my ten parametr. W przypadku użycia wielu procesorów (co w pythonie nie dawało dobrych efektów) parametr mógłby zostać dostosowany do ilości rdzeni.

# Zapis i wczytywanie DNA

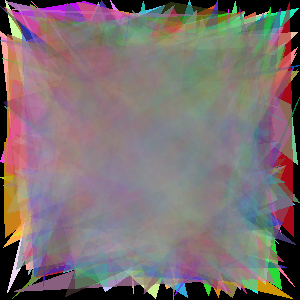
Dodatkowo program umożliwia zapisanie i wczytanie aktualnego DNA co pozwala na przechowanie wygenerowanych dna. Dodatkowo podczas procesu nauki możemy zapisywać co określoną epokę dna najlepszego osobnika co pozwala na prześledzenie procesu ewolucji populacji.

# Wyniki:

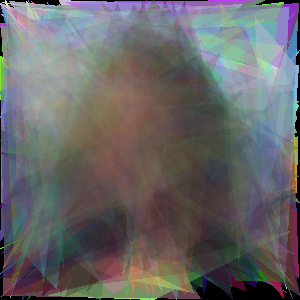
Poniżej prezentujemy wyniki poszczególnych symulacji.

## Mona Lisa

Symulacja przeprowadzona na 300 trójkątów, trwająca 30000 epok:



Rysunek 2 Mona Lisa iteracja nr 0



Rysunek 3 Mona Lisa iteracja 1000



Rysunek 4 Mona Lisa iteracja 5000



Rysunek 5 Mona Lisa iteracja 10000



Rysunek 6 Mona Lisa iteracja 20000

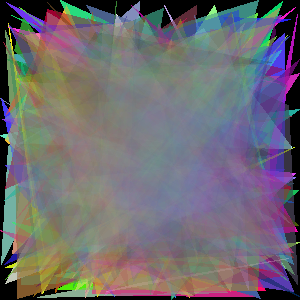


Rysunek 7 Mona Lisa iteracja 29900

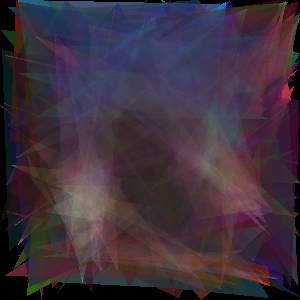
Wspólną cechą wielu symulacji był początkowy szybki wzrost podobieństwa a potem jego spowolnienie.

## Kopernik

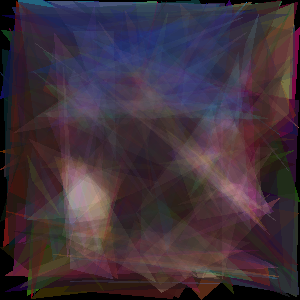
Symulacja na 250 trójkątach, trwająca 20000 epok



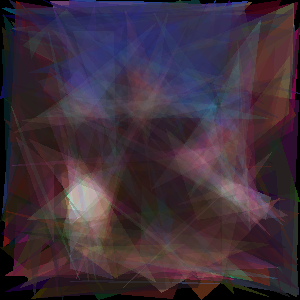
Rysunek 8 Kopernik iteracja 0



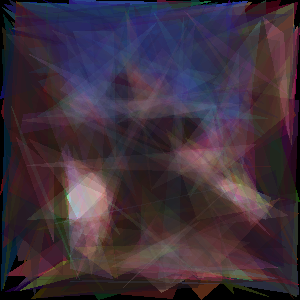
Rysunek 9 Kopernik iteracja 1000



Rysunek 10 Kopernik iteracja 5000



Rysunek 11 Kopernik iteracja 10000

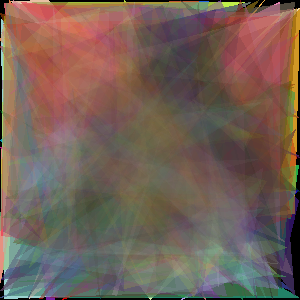


Rysunek 12 Kopernik iteracja 19900

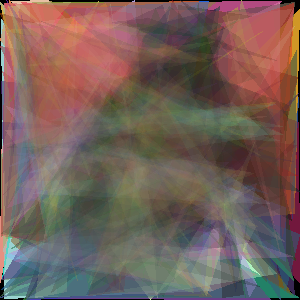
Jak widać algorytm jest w stanie przybliżać dowolne obrazy.

## Mieszko I

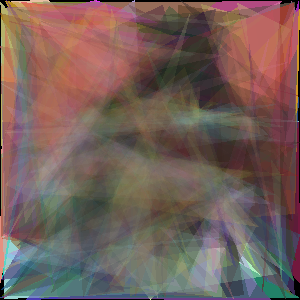
Symulacja na tych samych parametrach co Kopernik



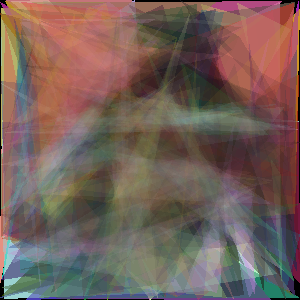
Rysunek 13 Mieszko I iteracja 1000



Rysunek 14 Mieszko I iteracja 5000



Rysunek 15 Mieszko I iteracja 10000



Rysunek 16 Mieszko I iteracja 19900

W tym obrazem algorytm poradził sobie dość słabo – jedną z hipotez jest to że obraz sam charakter generowanego obrazu:

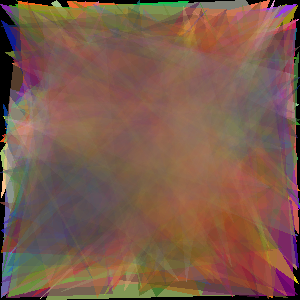


Rysunek 17 Mieszko I Jan Matejko

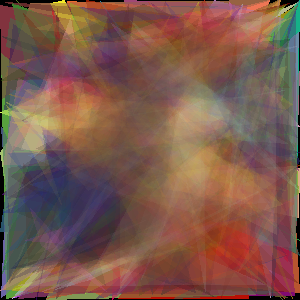
Który zawiera bardzo dużo ciemnych i jasnych miejsc znajdujących się bardzo blisko siebie – co powoduje że trójkątom jest ciężko znaleźć odpowiedni kolor.

### Marysia

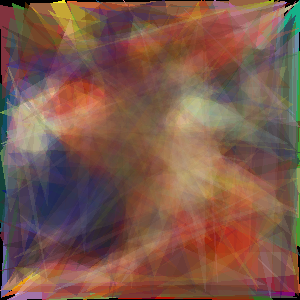
Obraz generowany na 300 trójkątach i w wymiarach (200, 200)



Rysunek Marysia iteracja 1000



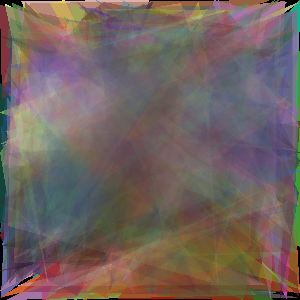
Rysunek Marysia iteracja 10000



Rysunek Marysia iteracja 29900

Niestety ten obraz Witkacego był duży i skomplikowany w efekcie do jego generowania użyliśmy za małej liczby trójkątów – a mimo to samo generowanie obrazu trwało około 5 godzin.

Użyliśmy natomiast obrazu wynikowego jako wejście do generowanie kolejny raz Mona Lisy.



Rysunek Mona Lisa wygenerowana z Marysi iteracja 100



Rysunek Mona Lisa wygenerowana z Marysi iteracja 500

Obraz bardzo szybko zatracił wygląd obrazu z którego został generowany.



Rysunek Mona Lisa wygenerowana z Marysi iteracja 39500

# Wnioski:

* zaprezentowany algorytm jest w stanie generować obrazy podobne do oryginału.
* generowanie obrazu trwa długo dlatego algorytmu nie można użyć jako metody kompresji obrazów.
* wygenerowane obrazy są w specyficzny sposób zdeformowane względem oryginałów co nadaje im pewną „wartość artystyczną”
* algorytm jest w stanie przyspieszyć jeśli zostałby napisany w innym, szybszym języku programowania niż Python, oraz jeśli algorytm zostanie zrównoleglony – generowanie obrazu z DNA zajmuje najwięcej czasu, a może być całkowicie zrównoleglone.