

Raport z projektu **Genetic Painting**

Maurycy Borkowski

15.12.2021

Przedstawienie problemu

Rozważamy problem *malowania* istniejących już obrazów prostymi figurami geometrycznymi. To znaczy tworzenie kompozycji i dobór kolorów by całość przypominała zadany obraz. Inspiracja (i prawdopodobnie) autor problemu.

Jako podstawowego elementu kompozycji my użyjemy trójkąta.

Dokładniej, genotyp każdego osobnika będzie się składał z k podstawowych elementów, trójkątów i ich kolorów. Fenotyp, trójkąt kodujemy za pomocą trzech par liczb, każda para odpowiada wierzchołkowi trójkąta, kolory kodujemy jako 4-krotki liczb z zakresu $[0, 256]$, (RGBA).

Funkcja celu oblicza jakość osobnika w dwóch etapach:

- Wygenerowanie obrazu (tablicy pixeli RGB) z danego osobnika
- Porównanie dwóch obrazów (MSE) i uśrednienie wyniku $\frac{1}{n \cdot m} \sum_i (p_i - o_i)^2$

Rozwiązanie

Okazuje się, że w rozwiązaniu tego problemu doskonale sprawdził prosty algorytm ewolucyjny:

```
input:  $N, T, N_d, obraz$   
 $P \leftarrow losowa\_populacja(N)$   
while  $i \leq T$  do  
   $P \leftarrow dodaj\_dzieci(N_d)$   
   $P \leftarrow wybierz\_najlepsze(N, obraz)$   
   $i \leftarrow i + 1$   
end while  
return  $P_{max}$ 
```

Dzieci dodajemy mutując każdego z osobników kilkakrotnie, z rozkładem normalnym zarówno współrzędne i kolory (o średniej w wartości rodzica). Z małym

prawdopodobieństwem zmieniamy osobnika na losowego.

Implementacja i pomysły

Naiwne uruchomienie algorytmu od razu na pełnej przestrzeni poszukiwań kończyło się niepowodzeniem, algorytm nie mógł znaleźć wzorca i ciągle błędził. Kluczem do sukcesu było dobieranie kilku trójkątów i *mrożenie* poprzednich z najlepszego osobnika. W naszej implementacji dodajemy 1 trójkąt na 250 iteracji i zmieniamy tylko dwa najnowsze.

Dodatkowo by zwiększyć efektywność w implementacji zadbaliliśmy by nie liczyć funkcji dopasowania (celu) dwa razy na tych samych osobnikach w każdej iteracji.

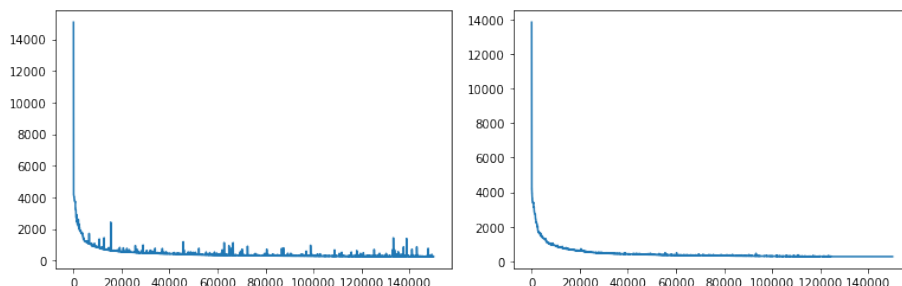
Początkowo algorytm i pomysły warto sprawdzać na małych rozmiarach obrazów, dla przyspieszenia obliczeń. Ostateczne obrazki liczą się $\approx 20h$.

Wyniki

Niestety z uwagi na bardzo długie obliczenia nie udało mi się zebrać wielu wyników i danych do analiz, ale o to część z nich.

Zgodnie z intuicją prostsze i mniejsze obrazy były o wiele łatwiejsze do przetworzenia.

Najlepsze obrazy powstały z populacji składającej się z jednego osobnika, co ciekawe jak widać na wykresie populacja pojedyncza często długo dopasowywała świeżo dodany, losowy trójkąt. Mimo to ostateczne obrazy dla tych samych ilości wywołań funkcji celu powstawały z pojedynczych populacji:



(a) 1 osobnik, 100 dzieci

(b) 5 osobników, 20 dzieci



(a) 1 osobnik, 100 dzieci

(b) 5 osobników, 20 dzieci

Widać różnicę, zwłaszcza w szczegółach twarzy, przy tej samej liczbie trójkątów (600).

Niestety podobna ilość iteracji i trójkątów nie daje podobnie dobrych efektów bardziej skomplikowanym obrazom (mimo zbliżonych rozmiarów):

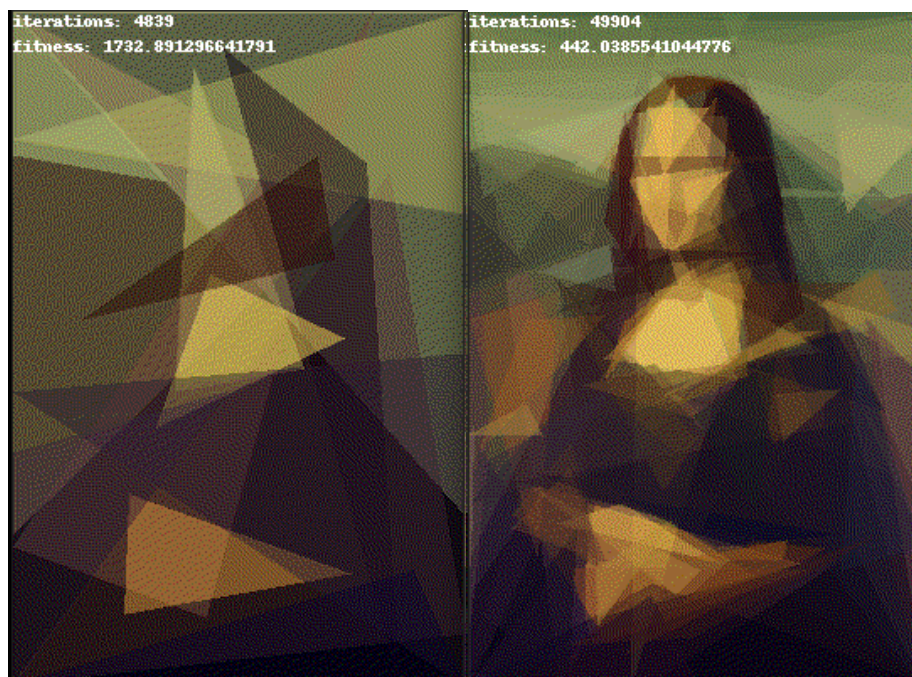


(a) wynik



(b) oryginał

Oto najlepsze dwa obrazy jakie udało mi się uzyskać przed kilkoma poprawkami:



(c) pierwszy dawałem trójkąty *po kolei*

(d) pierwsze dłuższe obliczenia z tym pomysłem

Podsumowanie

Zaskakująco trudne zadanie (do rozwiązania za pomocą standardowych metod) udało się rozwiązać dosyć prostym sposobem (i długimi obliczeniami). Wyniki są na prawdę satysfakcjonujące, ciekawe jest obserwowanie przebiegu algorytmu, jak dobiera kolejne trójkąty (nagrania przebiegów iteracji są w plikach z wynikami).

Ten projekt był na prawdę ciekawy i przyjemny w realizacji, najważniejsza rzecz jakiej się nauczyłem, to żeby planować obliczenia i brać pod uwagę ich długość.