Стилизация фотографии с помощью k-means

Кластеризация

- Разбиение списка объектов на группы, более или менее схожих элементов в соответствии с их признаками.
- Примеры: определение размера футболок для страны, определение типа покупателя в магазине (транжира, экономный и т.п.) -> специальные предложения, обработка социологических данных

K-means

- https://www.youtube.com/watch?v=_aWzGGNrcic

```
const int K = 15;
const int IterationCount = 30;

using Centroids = std::vector<sf::Color>;
using ClosestCentroidIndices = std::vector<int>;
```

Применяем для кластеризации цветов на фотографии

```
·auto·image·=·texture.copyToImage();

·Centroids·centroids·=·initializeCentroids(image,·K);

·for·(int·it·=·0;·it·<·IterationCount;·++it)
·{
···ClosestCentroidIndices·ids·=·findClosestCentroids(image,·centroids);
···centroids·=·computeMeans(image,·ids,·K);
·}

·changeColors(image,·centroids);

·image.saveToFile("result.png");
·texture.update(image);

·sf::Sprite·photo;
·photo.setTexture(texture);
```

В итоге получаем … (сжатие изображения/стилизация)

K=10

Из цветной фото

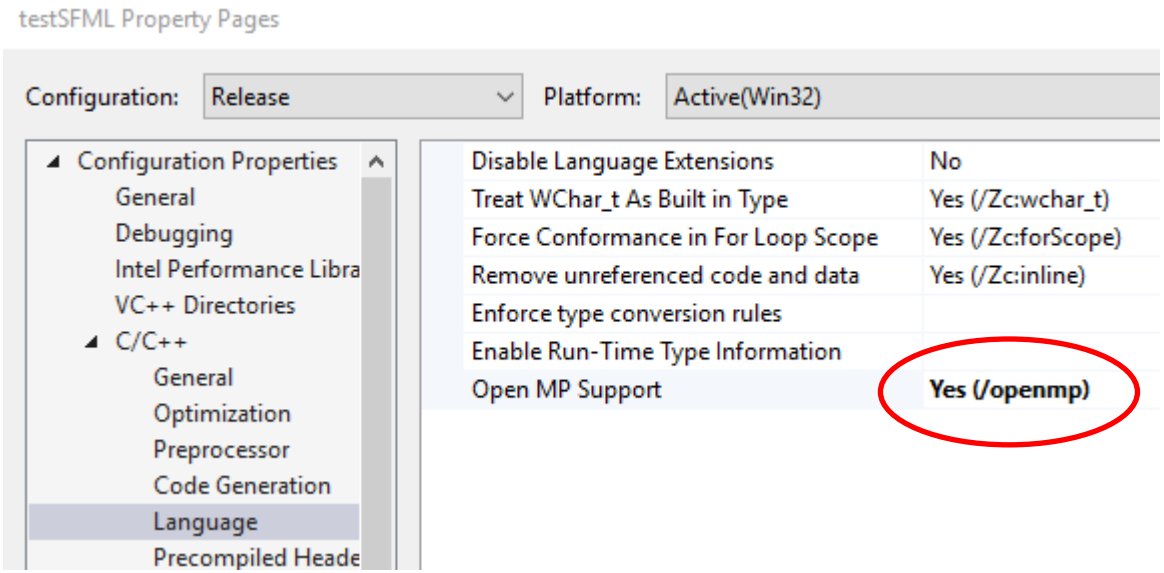
Из ЧБ фото

Даже при К=2 человека на фото можно узнать

Идею можно развить…

- … до проекта, позволяющего получить по фотографии заготовку для картины (разбиение на регионы + цвета их заливки)
- нужно убрать отдельные небольшие участки пикселей, выделить границы, выбрать более подходящее представление цвета (rgb – не лучший вариант)
- результат можно распечатать на бумаге/холсте и раскрасить

Как просто распараллелить свой код (библиотека openmp)




```
#pragma omp parallel for
for (int i = 0; i < image.getSize().y; ++i)
```