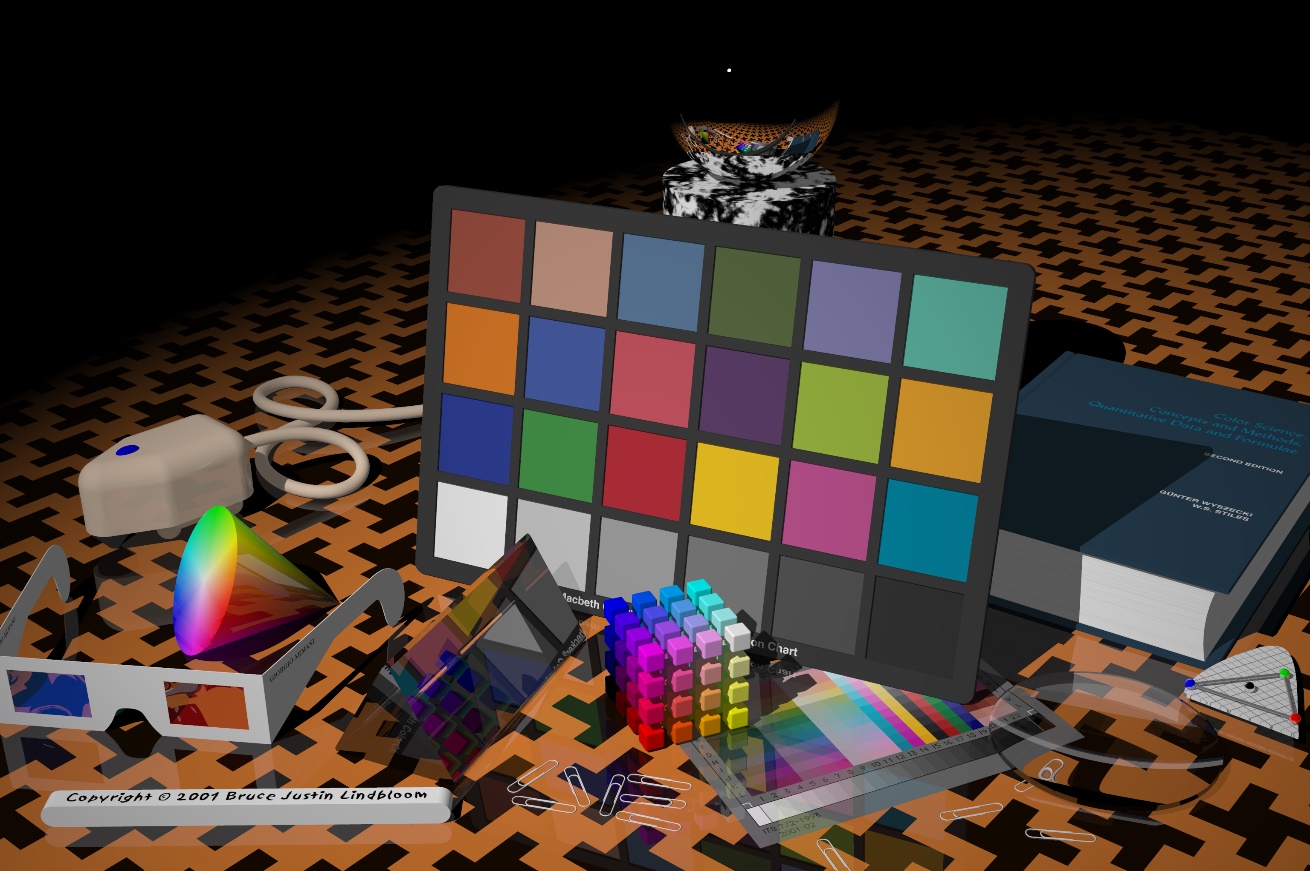
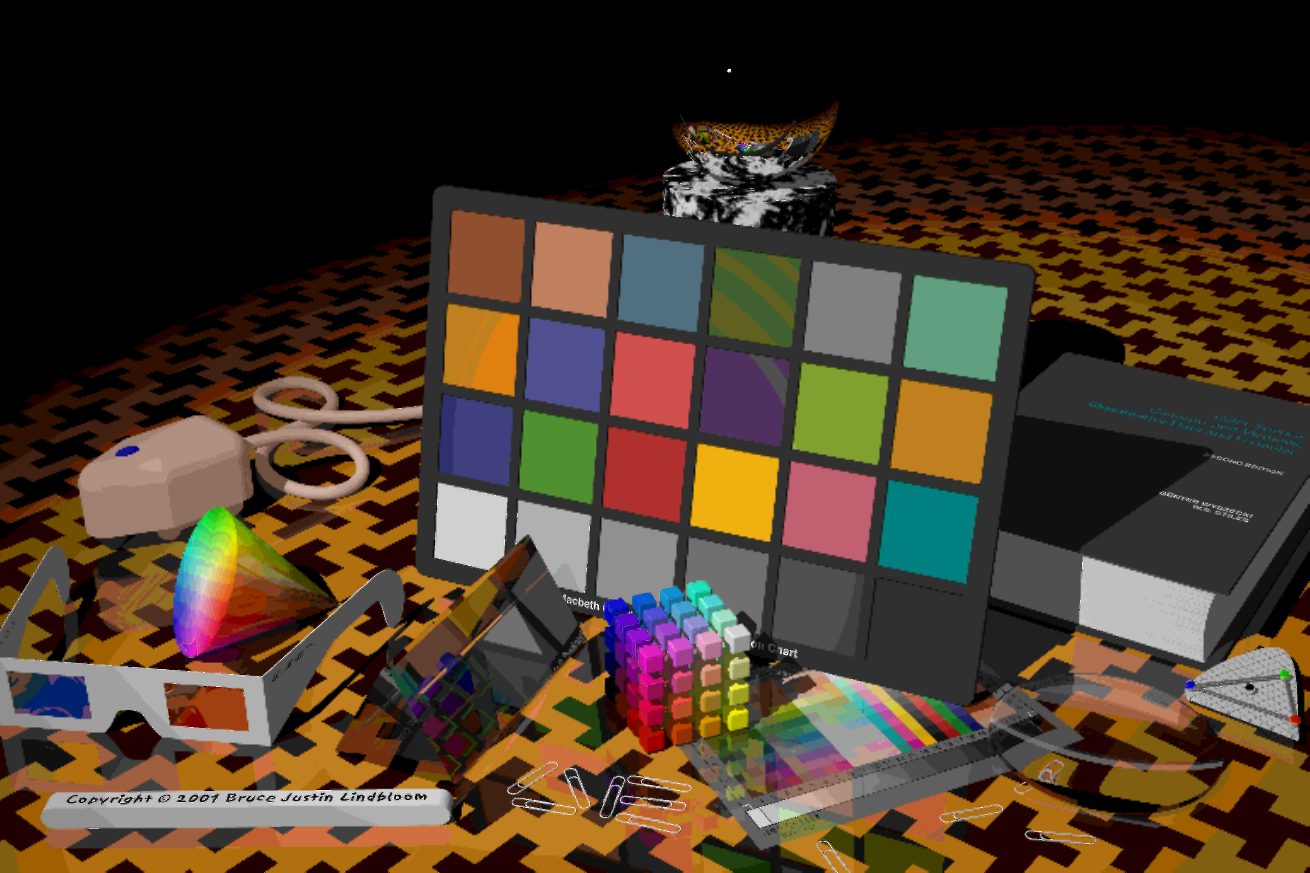
Hello, the task is to reduce image processing artifact after jpg for Compressed image.

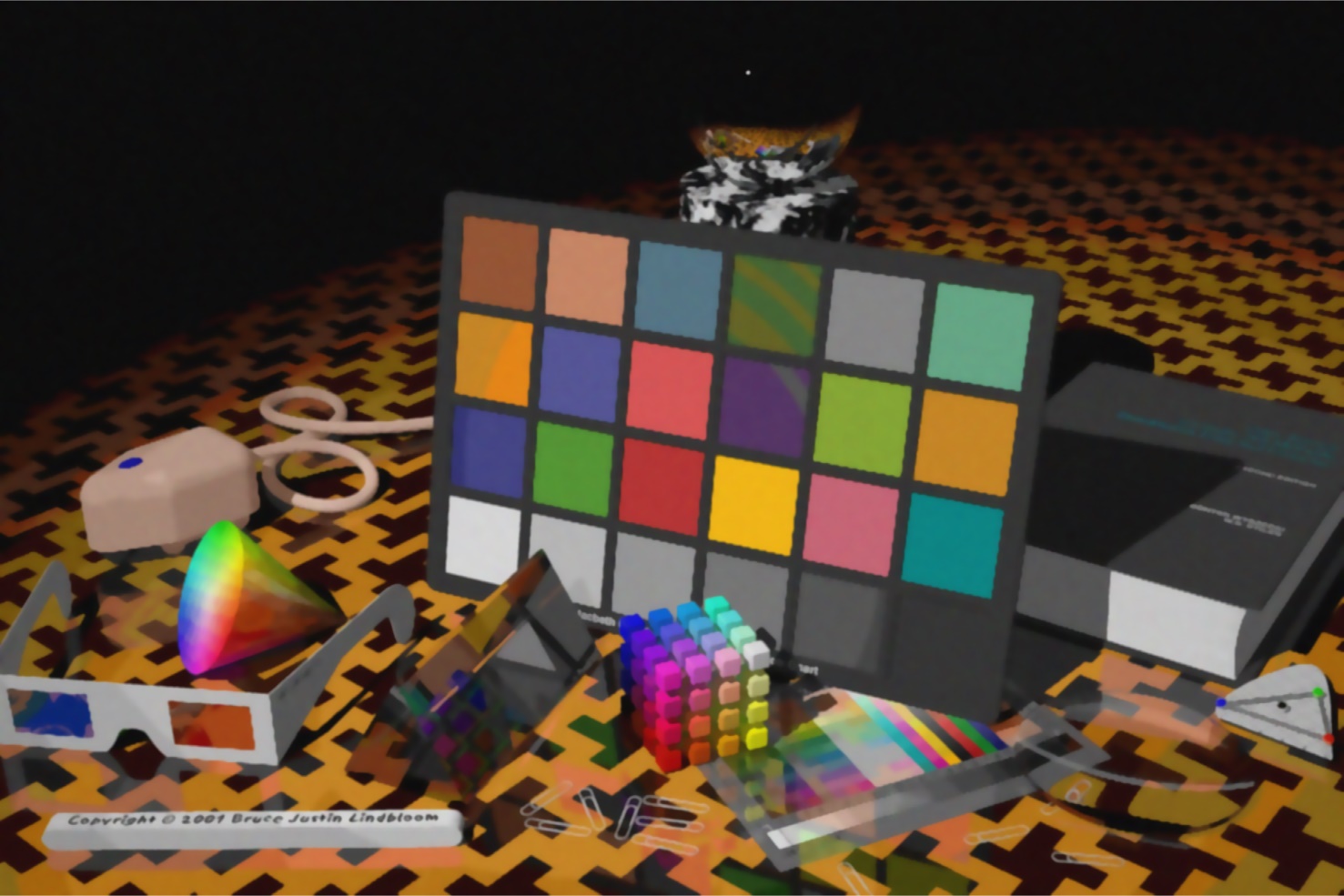
Original Image is:



Compressed image is:



My solution is:



Изображение выглядит как снимок экрана, коллаж, искусство, мультфильм

Автоматически созданное описание

My solution looks for this image.

The result of my algorithm partially resolved aliasing, a continuous transition of colors on the prism, but the image lost sharpness and the colors have a different filter, not the same as in the original image, these are the consequences of jpeg algorithm compression.

Artifact that not solved: (Left my solution, Right – Original Image)

Изображение выглядит как Красочность, Прямоугольник, апельсин, прямоугольный

Автоматически созданное описаниеИзображение выглядит как снимок экрана, прямоугольный, Прямоугольник

Автоматически созданное описание

Artifact that not solved: (Down my solution, Up – Original Image)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, знак

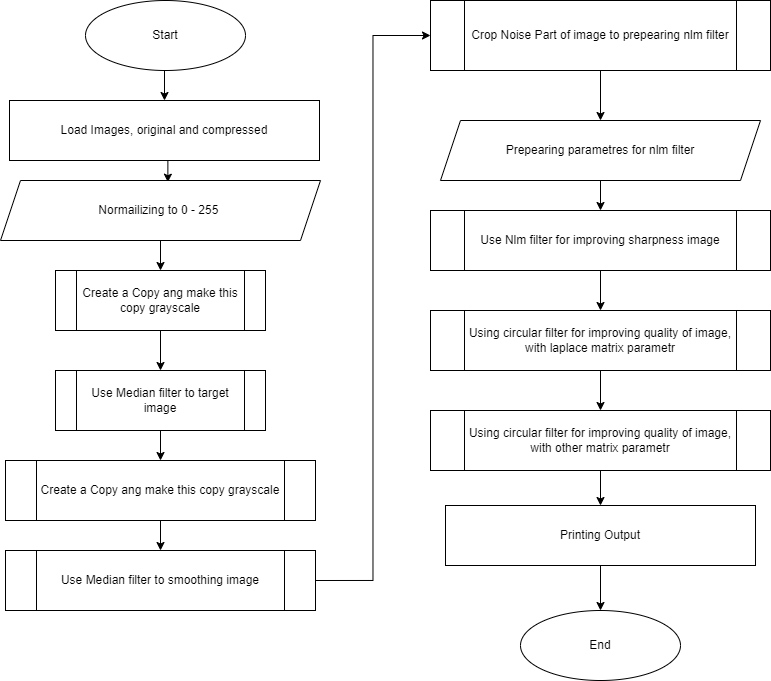
Автоматически созданное описание

Изображение выглядит как Красочность, Игрушечный блок

Автоматически созданное описаниеИзображение выглядит как Игрушечный блок, Красочность, Механическая головоломка

Автоматически созданное описание Artifact that not solved: (Left my solution, Right – Original Image)

Main Algorithm of Image Processing



Problem of current C++ code is performance. Need Improve all filters.

Questions:

1. How can I improve performance using STL containers ?
2. Need I make loop unroll using for image filtering?
3. Open CV filter is good, can I make my own filter that will be fast as opencv?

Files that need to be optimized.

1. convolution\_tools.cpp
2. gauss\_filter.cpp
3. image\_mat\_tools.cpp
4. image\_noiser.cpp
5. median\_filter.cpp
6. nlm\_filter.cpp