Алгоритмы быстрого размытия изображений, основанные на фильтре Гаусса

П. Н. Красковский, email: kraskovskij@bsuir.by [1]

[1] Белорусский государственный университет информатики и радиоэлектроники

Аннотация. *Рассмотрены эффективные алгоритмы размытия изображений, позволяющие существенно уменьшить время обработки, по сравнению с алгоритмом размытия Гаусса. Произведено сравнение времени работы рассмотренных алгоритмов, а также выявлена зависимость между степенью размытия изображения и скоростью его обработки.*

Ключевые слова: *размытие изображений, аппроксимация, размытие Гаусса, скользящее среднее, ядро свёртки.*

Введение

Размытие изображений широко используется в машинном обучении, в игровых приложениях, в программах для редактирования растровой и векторной графики, цифровой обработки сигналов и т.д. Размытие изображений часто применяется для выделения общих черт объектов при распознавании образов, создания различных постэффектов в компьютерных играх, сокрытия конфиденциальной информации, уменьшения шума и т.д.

Существует несколько способов размытия изображений, одним из которых является размытие Гаусса. Данный способ основан на использовании функции Гаусса, которая совпадает с функцией плотности вероятности нормального распределения. Данная функция используется для построения ядра свёртки, которое применяется к каждому пикселю изображения для его размытия [1].

Основная суть алгоритма заключается в том, чтобы обработать каждый пиксель изображения путём сложения этого пикселя с соседними пикселями в некоторой окрестности, ограниченной размером ядра свёртки, предварительно умножив все суммируемые пиксели на соответствующие коэффициенты ядра свёртки, вычисленные с помощью функции Гаусса. Чем больше пикселей участвуют в вычислении результирующего пикселя (чем больше размер ядра свёртки), тем больше эффект размытия.

С увеличением разрешения изображения необходимо увеличивать размер ядра свёртки, чтобы достичь приемлемого уровня размытия. Так как ядро свёртки представляет собой квадратную матрицу, которую нужно применить к каждому пикселю изображения, с увеличением размера ядра количество выполняемых операций увеличивается квадратично.

В данной работе рассмотрены алгоритмы размытия изображений, использующие свойства функции Гаусса, которые позволяют существенно сократить количество выполняемых операций.

1. Стандартное размытие Гаусса с использованием двухмерной матрицы коэффициентов

Прежде чем применить размытие Гаусса к изображению, необходимо рассчитать весовые коэффициенты ядра свёртки, которые затем будут использоваться для обработки каждого пикселя изображения. Для этого можно использовать функцию Гаусса для двух измерений по формуле (1), которая получается путём перемножения двух одномерных функций Гаусса для каждого измерения.

$$G(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}, \quad (1)$$

- σ – стандартное отклонение нормального распределения;
- x – горизонтальное смещение от центра ядра свёртки;
- y – вертикальное смещение от центра ядра свёртки.

Так как в функции Гаусса используется стандартное отклонение нормального распределения, необходимо правильно определить соответствующий размер ядра свёртки. Это можно сделать, используя правило трёх сигм. Согласно данному правилу, практически все значения нормально распределённой случайной величины лежат в интервале 6σ. Из этого следует, что размер ядра свёртки должен быть равен $6\sigma + 1$.

На рис. **Ошибка! Источник ссылки не найден.** представлена матрица коэффициентов, вычисленная по формуле (1) для $\sigma = 0{,}6$. Применив операцию свёртки для каждого пикселя изображения с использованием сгенерированной матрицы, получается новое размытое изображение.

0,00001	0,00043	0,00171	0,00043	0,00001
0,00043	0,02749	0,11024	0,02749	0,00043
0,00171	0,11024	0,4421	0,11024	0,00171
0,00043	0,02749	0,11024	0,02749	0,00043
0,00001	0,00043	0,00171	0,00043	0,00001

Рис. 1. Матрица коэффициентов для $\sigma = 0,6$

На рис. 2 показан результат применения двухмерного размытия Гаусса с различными значениями сигмы, а также время обработки каждого изображения. Из рис. 2 видно, что при увеличении значения сигмы увеличивается степень размытия изображения. При этом время обработки изображения значительно увеличивается.

а б в

а – $\sigma = 5$, *б* – $\sigma = 10$, *в* – $\sigma = 20$

Рис. 2. Результат применения двухмерного размытия Гаусса

2. Двухпроходное размытие Гаусса с использованием одномерной матрицы коэффициентов

На рис. 1 можно увидеть, что матрица коэффициентов обладает свойством симметричности. Данное свойство можно использовать для сокращения времени обработки изображений. Для этого необходимо

построить одномерный массив коэффициентов по формуле (2), используя функцию Гаусса для одного измерения [1].

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (2)$$

Полученный массив применяется к каждому пикселю изображения сначала в виде вектора-строки для размытия изображения по горизонтали, а затем в виде вектора-столбца для размытия результирующего изображения по вертикали. В итоге получается то же самое изображение, что и после применения двухмерного размытия Гаусса, однако временная сложность алгоритма становится линейной $O(2n)$.

На рис. 3 показано сравнение двухмерного размытия Гаусса и двухпроходного размытия Гаусса для $\sigma = 20$. Из рис. 3 видно, что время обработки двухпроходным размытием Гаусса значительно сократилось, по сравнению с двухмерным алгоритмом, а стандартное отклонение между изображениями пренебрежительно мало (0,00507%), т.е. разница между изображениями практически отсутствует.

а б

а – двухмерное размытие Гаусса, б – двухпроходное одномерное размытие Гаусса

Рис. 3. Сравнение двух алгоритмов размытия изображений

3. Аппроксимация размытия Гаусса трёхпроходным скользящим средним

Размытие Гаусса можно аппроксимировать скользящим средним, если последовательно применить данный фильтр несколько раз. Чем больше количество применений фильтра, тем точнее аппроксимация. Трёх применений достаточно в большинстве случаев.

Скользящее среднее представляет собой фильтр, который производит усреднение всех пикселей, попавших в скользящее окно, заданного размера.

Так же, как и для двухпроходного размытия Гаусса, двухмерное скользящее окно можно разбить на горизонтальное и вертикальное одномерные окна. Однако нет необходимости на каждом шаге алгоритма вычислять среднее арифметическое всех пикселей, попавших в скользящее окно, так как можно воспользоваться суммой, полученной на предыдущем шаге. Для этого необходимо из предыдущей суммы вычесть крайний левый пиксель окна на предыдущем шаге и прибавить крайний правый пиксель окна на текущем шаге [2].

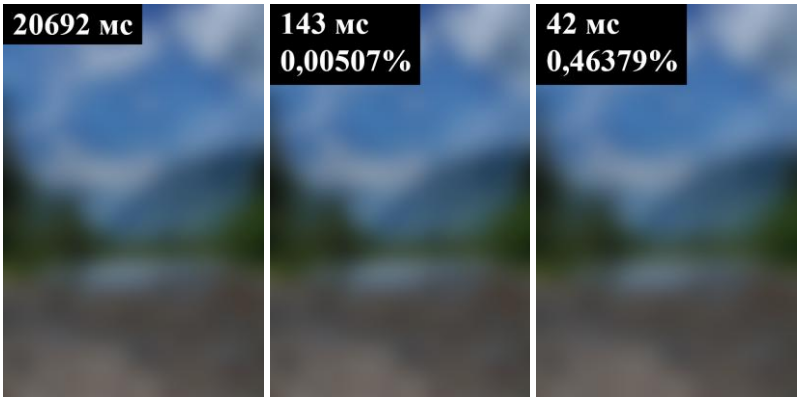
После завершения всех проходов скользящим средним необходимо разделить накопленные суммы для каждого пикселя на количество пикселей, участвовавших в формировании суммы.

Также для точной аппроксимации размытия Гаусса необходимо вычислить размер скользящего окна для заданной сигмы по формуле (3) [3].

$$size = \sqrt{\frac{12\,\sigma^2}{n} + 1}\,, \tag{3}$$

– n – количество проходов скользящим средним.

На рис. 4 показано сравнение трёх рассмотренных алгоритмов для $\sigma = 20$. Аппроксимация размытия Гаусса скользящим средним показывает наилучшее время обработки изображения при незначительном стандартном отклонении от эталонного изображения (0,46379%). Так как в данном алгоритме происходит накопление суммы пикселей, скорость обработки изображения практически не зависит от размера скользящего окна.



а б в

а – двухмерное размытие Гаусса, б – двухпроходное одномерное размытие Гаусса, в – трёхпроходное скользящее среднее

Рис. 4. Сравнение трёх алгоритмов размытия изображений

Заключение

В данной работе были рассмотрены три алгоритма размытия изображений. Также были проанализированы временные затраты каждого алгоритма, и выявлена зависимость между степенью размытия изображения и скоростью обработки. В результате можно сделать вывод, что аппроксимация размытия Гаусса скользящим средним даёт схожие результаты при существенном сокращении времени обработки, которое не зависит от размера окна (значения сигмы).

Список литературы

1. LearnOpenGL – Bloom [Электронный ресурс]. – Режим доступа : https://learnopengl.com/Advanced-Lighting/Bloom

2. Fastest Gaussian Blur (in linear time) [Электронный ресурс]. – Режим доступа : http://blog.ivank.net/fastest-gaussian-blur.html

3. Kovesi P. Fast Almost-Gaussian Filtering / P. Kovesi // The Australian Pattern Recognition Society Conference: DICTA 2010. – Sydney, Australia, December 2010. – P. 1-5.