УДК 004.92

О НЕКОТОРЫХ АЛГОРИТМАХ РАЗМЫТИЯ ИЗОБРАЖЕНИЙ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ ИХ ПРОГРАМНЫХ РЕАЛИЗАЦИЙ

Бенидзе Нана, Габисония Инга

Сухумский государственный университет, Тбилиси, Грузия

Аннотация

В данной работе проанализированы некоторые виды алгоритмов размытия, широко применяемых в компьютерной графике, методы их програмной реализации на языке программирования C++, которые позволяют в ряде случаев повысить скорость работы (быстродействие) алгоритмов. А также даны эмпирические оценки при распараллеливании вычислительных процессов.

Abstract

In this paper, we analyzed some types of Blur Algorithms that are widely used in computer graphics; also methods of their software implementation in the C ++ programming language, which allow in some cases to increase the speed of the algorithms. And also empirical estimates are given for Parallelizing Computational processes.

Введение

В компьютерной графике широко применяются различные виды размытия изображений. Существует большой класс разработанных алгоритмов размытия, которые применяются в графических программах. Данные алгоритмы также полезны при сжатии графических изображений, так как размытые изображения достаточно удобно сжимаемы и хорошо обрабатываемы. Поэтому данные в работе эмпирические оценки для некоторых классов алгоритмов могут быть весьма полезны в практических реализациях.

Существует достаточно большое количество алгоритмов размытия изображений, применяемых в различных графических редакторах, причем они практически постоянно усовершенствуются. Несмотря на известность математической составляющей данного класса алгоритмов, при програмной реализации встречается множество "подводных камней", связанных с реализацией алгоритмов на конкретном языке программирования. Интересен также вопрос о распараллеливании вычислений, так как даже для сложных алгоритмов заметные результаты с целью ускорения работы достигаются уже в случае четырехьядерных процессоров (восьмипоточные алгоритмы).

Наиболее распространенным, но не всегда наилучшим, является Алгоритм Гаусса, основанный на классическом N -мерном преобразовании Гаусса, которое имеет вид:

$$G(r) = \frac{1}{\left(2\pi\sigma^2\right)^{N/2}} e^{\frac{-r^2}{2\sigma^2}},$$

В двумерном случае можно записать в виде:

$$G(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}},$$

где r-радиус размытия $r^2 = u^2 + v^2$, σ- стандартное отклонение распределения Гаусса. Так, в случае двух измерений этой формулой задается семейство концентрических окружностей с распределением Гаусса от центральной точки. Значение каждого из пикселей будет средне взвешенным для соседних пикселей. Исходное значение пикселя принимает наибольшее гауссово значение, а соседние пиксели

принимают меньшие значения (веса), которые зависят от расстояния до них. В практических реализациях при работе с дискретным преобразованием Гаусса в алгоритмах не учитываются те пиксели, расстояние до которых больше, чем 3σ из-за малости значений. Последнее означает достаточность расчета матрицы размером $[6\sigma]X[6\sigma]$ для приемлемой точности приближения. Удобность применения Гауссовых преобразований в силу свойства линейной сепарабельности (т.е. свойства факторизации) вместо применения двухмерной матрицы сводится к применению двух одномерных матриц по горизонтальному и вертикальному направлениям. Данный факт позволяет снизить обьем вычислений до $O(n\times M\times N)+O(m\times M\times N)$ вместо $O(n\times m\times M\times N)$ для несепарабельной матрицы свертки (ядра), где M,N -размеры фильтруемого изображения, m,n -размеры ядра фильтра.

Обычно при программной реализации преобразование осуществляется над окном, состоящим из пикселя, находящегося в центре и восьми соседних обрамляющих пикселей, причем каждый из этих девяти пикселей разбивается на составляющие в цветовой модели (наиболее часто применима модель - RGB). При применении преобразования нужно нормализовать коэффициенты в диапазоне (0...255). В работе также анализируются результаты работы алгоритмов, в которых применяются рекурентные соотношения, что заметно ускоряет процесс вычислений.

В данной работе проанализированы некоторые виды алгоритмов размытия, методы их програмной реализации на C++, которые позволяют в ряде случаев повысить скорость работы (быстродействие) алгоритмов. А также даны эмпирические оценки при распараллеливании вычислительных процессов. Приведены програмные коды, иллюстрирующие эти эмпирические оценки.

Список использованных источников:

1.Frederick M. Waltza and John W. V. Millerb, "An efficient algorithm for Gaussian blur using finite-state machines", a2095 Delaware Avenue, Mendota Heights, MN 55118-4801 USA bECE Department, Univ. of Michigan-Dearborn, Dearborn, MI 48128-1491 USA, SPIE Conf. on Machine Vision Systems for Inspection and Metrology VII Originally published Boston, Nov. 1998 Copyright July 1998 by F. M. Waltz.

2.Christopher Mei and Ian Reid, Department of Engineering Science University of Oxford, Modeling and Generating Complex Motion Blur for Real-time Tracking, {cmei,ian}@robots.ox.ac.uk, 2008.

3.L.HÁJEK, Gaussian blur in Matlab, http://kfe.fjfi.cvut.cz/~hajeklu2/files/PIN3/Bonus/GaussianBlur/GBlur.m

4.G.J.Brostow and I.Essa. Image-based motion blur for stop motion animation. In Proc. Of ACM SIGGRAPH'01, 2001.4

5.S.Cho, Y.Matsushita and S.Lee. Removing non-uniform motion blur from images. In IEEE International Conference on Computer Vision, 2007.