

# Метод интерполяционного повышающего масштабирования растровых изображений

## Аннотация

В данной статье будут рассмотрены существующие методы повышающего масштабирования, их достоинства и недостатки, а также будет сконструирован комбинированный метод, ориентированный на приложения, рассчитанные на экраны с относительно небольшим разрешением на основе двух имеющихся.

## Введение

Масштабирование изображения с повышением разрешения широко применяется в компьютерной графике, обработке видео, в частности, реализуется на аппаратном уровне в телевизорах и видеопроигрывателях. Так современные телевизоры и мониторы имеют достаточно большое разрешение, наиболее популярные форматы[1] HD и FHD означают  $1280*720$  и  $1920*1080$  пикселей соответственно в то время как множество телепрограмм было сделано в формате 576i или 480i, то есть  $720*576$  или  $640*480$  пикселей. Также сейчас достаточно популярны старые игры, сделанные в 90х годах 20го века, они были предназначены для вывода изображения на мониторы с разрешением порядка  $320*200$ , что в разы меньше современных.

## Анализ существующих алгоритмов

Интерполяция – нахождение промежуточных значений по имеющемуся дискретному набору известных значений. Повышающее масштабирование может быть кратным, в 2, 3 или 4 раза больше исходного разрешения, а может быть универсальным, то есть подходящим под любое выходное разрешение.

## Анализ универсальный алгоритмов

Существует несколько достаточно простых универсальных алгоритмов масштабирования: Метод ближайшего соседа, Билинейный и Бикубический [2] (см рис.1). Они также пригодны для понижающего масштабирования.

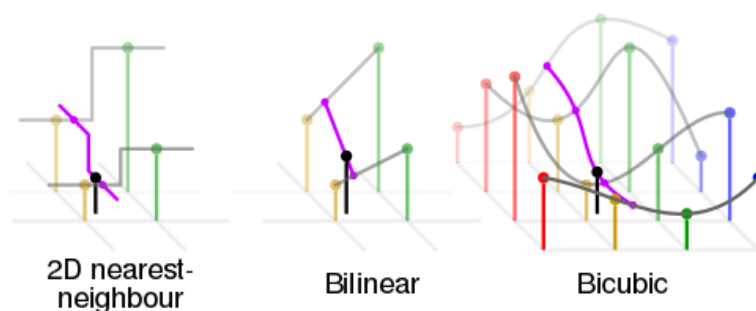


Рисунок 1

## Метод ближайшего соседа

Это самый примитивный и быстрый метод. Для каждого пикселя конечного изображения выбирается один пиксель исходного, наиболее близкий к его положению с учетом масштабирования. Такой метод дает изображение с значительно проявляющимся артефактом ступенчатости при увеличении и сильно зернистое изображение при уменьшении. Метод ближайшего соседа сохраняет резкие границы, но привносит в изображение ступенчатость, особенно, диагональные линии напоминают «лесенку» из квадратов.

## Билинейная интерполяция

Цветовые координаты текущего пикселя обычно вычисляются путём интерполяции четырёх соседних, в результате чего обработанное изображение получается размытым. Хотя это приемлемо для полноцветных изображений, применение интерполяции ведет к снижению контрастности (резкости на границах), и, поэтому, данный метод дает плохие результаты на изображениях с небольшим количеством цветов.

## Бикубическая интерполяция

Обобщение кубической интерполяции функций одной переменной для функций двух переменных. Кубическая интерполяционная функция также называется кубическим сплайном. Для построения кубического сплайна необходимо 4 точки, то есть для бикубической интерполяции всего используется 8 точек так как мы строим сплайн по высоте и по ширине. Кубический сплайн – это полином третьей степени, для которого необходимо

$$S_i(x) = a_i + b_i(x - x_i) + \frac{c_i}{2}(x - x_i)^2 + \frac{d_i}{6}(x - x_i)^3 \quad (1)$$

найти коэффициенты.

Пусть

$$h_i = x_i - x_{i-1}, f_i = f(x_i), \text{ тогда}$$

$$a_i = f(x_i),$$

$$c_0 = c_n = 0, \quad (2)$$

$$h_i c_{i-1} + 2(h_i + h_{i+1})c_i + h_{i+1} c_{i+1} = 6\left(\frac{f_{i+1}-f_i}{h_{i+1}} - \frac{f_i-f_{i-1}}{h_i}\right),$$

$$d_i = \frac{c_i - c_{i-1}}{h_i}, \quad b_i = \frac{f_i - f_{i-1}}{h_i} + \frac{h_i(2c_i + c_{i-1})}{6}$$

Метод	Ближайшего соседа	Билинейный	Бикубический
Ступенчатость	Значительная	Незначительная	Незначительная
Размытие	Отсутствует	Значительное	Незначительное

### Анализ кратных алгоритмов

Кратные алгоритмы основываются на анализе небольшой области вокруг исходного пикселя для получения новых.

Наиболее известными алгоритмами этого класса являются: [3]

- EPX\Scale2x\AdvMAME2x [4]
- Eagle
- SaI
- hqx [5]
- Edge-Directed-Interpolation

Метод	EPX	Eagle	SaI	hqx	NEDI
Ступенчатость	Незначительная	Незначительная	Незначительная	Отсутствует	Незначительная
Размытие	Отсутствует	Отсутствует	Отсутствует	Отсутствует	Незначительное
Потеря детализации	Отсутствует	Присутствует	Отсутствует	Отсутствует	Отсутствует

На рис.2 видно сравнение ряда рассмотренных алгоритмов.

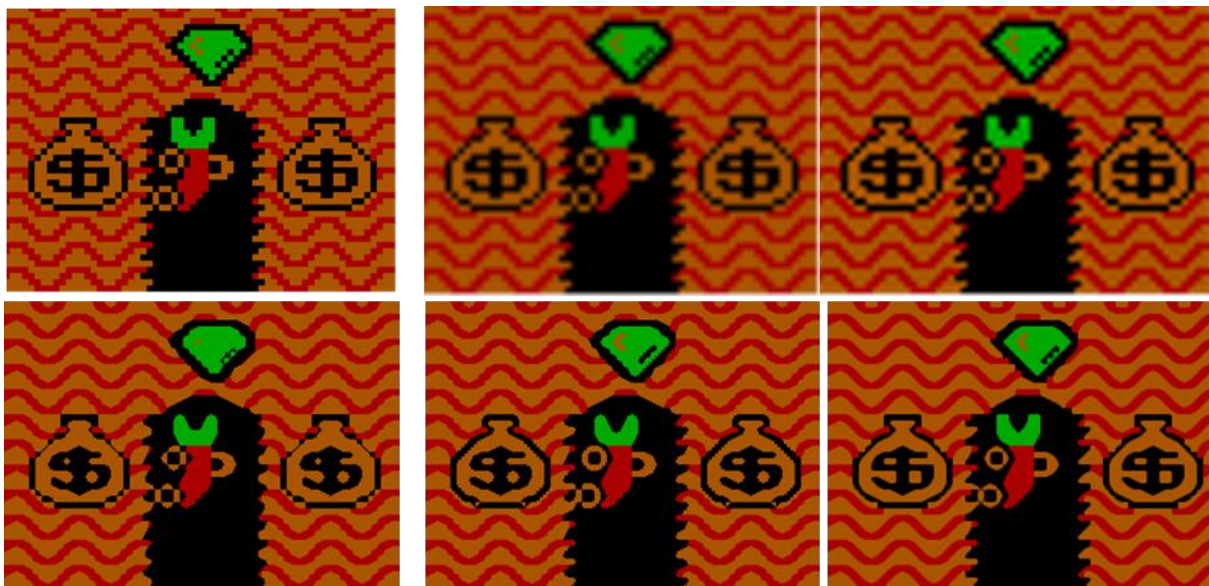


Рисунок 2 Сравнение методов: ближайшего соседа, билинейного, бикубического, Eagle, EPX, HQX

Из данного сравнения можно сделать вывод что наиболее перспективным методом является метод hqx, рассмотрим его подробнее.

### Детальное описание метода hqx

Метод генерирует на основе сравнения пикселя исходного изображения со всеми его окружающими четыре новых. Определение схожести цвета происходит путём сравнения цветов с определённым порогом, рекомендуемыми порогом являются  $T_Y = \frac{47}{255}$ ,  $T_U = \frac{7}{255}$ ,  $T_V = \frac{6}{255}$ . Эти пороги означают максимальную допустимую разницу компонент цветового пространства YUV. На основе схожести с 8ю соседними пикселями возможно построить соответственно  $2^8 = 256$  паттернов, для каждого из которых определены коэффициенты интерполяции новых четырёх пикселей. Таким образом генерируется изображение с в два раза большей шириной и высотой.

### Анализ алгоритмов векторизации

Большинство алгоритмов векторизации проприетарные, кроме того их общим недостатком являются высокие трудозатраты, однако существует современный алгоритм ориентированный на игры с низким разрешением.

### Алгоритм Копфа-Лицински

Этот алгоритм отличается от остальных используемых для пиксельной графики тем что он переводит пиксельное изображение с небольшим количеством цветов в векторную форму. Стоит заметить, что если к изображению до этого было применено сглаживание, результаты получаются хуже. Однако данный алгоритм достаточно ресурсозатратный, по

измерениям, приведённым в оригинальности статье[6] обработка одного кадра 256\*256 занимает в среднем 0.79с на процессоре частотой 2.4ггц, он состоит из нескольких этапов:

- Создание полносвязного графа существующих пикселей
- Преобразование графа в планарный на основе определённых эвристик
- Построение сплайнов, соединяющих грани графа
- Оптимизация сплайнов
- Растеризация

### Создание комбинированного метода

На основе проведённого ранее анализа были выбраны два метода – метод билинейного масштабирования и метод hqx. Метод hqx был выбран как имеющий меньшее количество артефактов кратный метод, а билинейный был выбран для понижающего масштабирования так как он даёт схожее с бикубическим качество при масштабировании в 2-3 раза [2].

Комбинированный метод состоит из нескольких этапов:

- Перевод из цветового пространства RGB в YUV
- Кратное масштабирование методом HQx превышающее целевое разрешение или равное ему
- Перевод из цветового пространства YUV в RGB
- Понижающее масштабирование билинейным методом если целевое разрешение не кратно исходному

### Результаты сравнения

Комбинированный метод сочетает достоинства метода hqx с возможностью универсального масштабирования, что, однако делает его более ресурсозатратным. Проведём сравнение с несколькими другими рассмотренными методами на задачи масштабирования разрешения 320\*200 до разрешения 1280\*800. Тестовый стенд использует процессор Intel i5-6200u.

Метод	Количество кадров в секунду, сцена 1	Количество кадров в секунду, сцена 2
-------	--------------------------------------	--------------------------------------

Ближайшего соседа	14.1	14.2
Билинейный	14	14.1
Бикубический	10.5	10.3
EPX	14.3	14.1
Eagle	14	13.8
HQx	10.1	10
Комбинированный	8.3	8.5

Также визуально сравним результаты работы комбинированного метода с hqx (см рис.3).

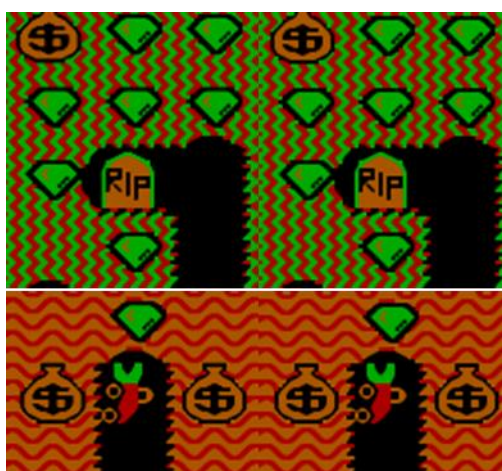


Рисунок 3 Сравнение HQx2 и комбинированного метода

## Выводы

По результатам экспериментов можно сказать что все рассмотренные методы подходят для работы в реальном времени, в том числе разработанный комбинированный метод, при чём он сочетает достоинства как метода билинейного масштабирования, так и метода hqx2 ценой несколько больших ресурсозатрат.

## Заключение

В рамках данной работы нами был проведён анализ существующих алгоритмов повышающего масштабирования растровых изображений, два из них были успешно скомбинированы, шесть реализованы, после чего был проведено исследование скорости их работы и наличия артефактов. Также в рамках исследования были подтверждены теоретические преимущества и недостатки алгоритмов. По итогам данного исследования можно утверждать, что комбинированный алгоритм повышающего может использоваться в реальном времени и имеет преимущества над более примитивными алгоритмами.

## Список использованных источников

1. Steam Hardware Survey – Режим доступа:  
<https://store.steampowered.com/hwsurvey>
2. Трубаков А.О. Сравнение интерполяционных методов масштабирования растровых изображений / Трубаков А.О., Селейкович М.О.// Научно-технический вестник Брянского государственного университета – 2017 - №1 – С. 92-98.
3. Pixel-art scaling algorithms – Режим доступа:  
[https://en.wikipedia.org/wiki/Pixel-art\\_scaling\\_algorithms](https://en.wikipedia.org/wiki/Pixel-art_scaling_algorithms)
4. MAZZOLENI, A., 2001. Scale2x. – Режим доступа: <http://scale2x.sourceforge.net>.
5. STEPIN, M., 2003. Hqx - Режим доступа: <http://web.archive.org/web/20070717064839/www.hiend3d.com/hq4x.html>
6. Johannes Kopf and Dani Lischinski - Depixelizing pixel art // ACM Transactions on Graphics (Proceedings of SIGGRAPH 2011). 2011 – Режим доступа: <http://johanneskopf.de/publications/pixelart/>