

Сжатие трехмерных табулированных таблиц с помощью тензорной декомпозиции

Д.В. Макаров, М.К. Чобану

*Национальный исследовательский университет «МЭИ»,
г. Москва, Россия*

makarovdmv@gmail.com

Трехмерные табулированные таблицы (3DLUT) используются для ускорения сложных нелинейных операций. Большой размер трехмерных таблиц налагает ограничения на их применимость, особенно на уровне аппаратной реализации: ограничение количества различных 3DLUT, следовательно, количества различных цветовых преобразований, ограничение частоты вывода кадров из-за значительного времени пересылки таблицы из ОЗУ в аппаратный модуль. В работе рассмотрен метод сжатия, позволяющий эффективно уменьшить объем 3DLUT за счет контролируемого повышения уровня сложности вычислений. Предлагаемый метод основан на канонической тензорной декомпозиции, которая показала наибольшую эффективность по сравнению с другими декомпозициями. Проведена оценка эффективности метода на примере сжатия 3DLUT для операции преобразования цветового охвата, которая необходима для любого устройства вывода. В результате сравнения установлено, что при отсутствии видимых искажений может быть достигнуто значительное сжатие (шестикратное для таблицы размера $17 \times 17 \times 17$). Особенностью предлагаемого подхода является возможность вычисления отдельного элемента исходной таблицы на основе сжатых данных без необходимости полного восстановления 3DLUT, что существенно упрощает аппаратную реализацию.

Ключевые слова: тензорная декомпозиция; табулированная таблица; 3DLUT

Для цитирования: Макаров Д.В., Чобану М.К. Сжатие трехмерных табулированных таблиц с помощью тензорной декомпозиции // Изв. вузов. Электроника. – 2018. – Т. 23. – № 6. – С. 605–614. DOI: 10.24151/1561-5405-2018-23-6-605-614

3D Lookup Table Compression with Use of a Tensor Decomposition

D.V. Makarov, M.K. Tchobanou

National Research University «Moscow Power Engineering Institute»,
Moscow, Russia

makarovdmv@gmail.com

Abstract: The 3D lookup tables (3DLUT) are used for speeding up the sophisticated nonlinear operations. They are commonly used in displays and other output devices for color processing, in particular, the gamut mapping operations for images and video. However, the large size of 3DLUTs imposes the restrictions on their applicability, especially at the hardware level. It puts a limitation on a number of different 3DLUTs used (hence, a number of different color transformations) and on the frame rate due to a significant amount of time, needed for transferring LUT from RAM to the hardware module. In the paper a method for effective reducing the 3D lookup table size with the control of computational complexity has been considered. The proposed approach is based on the canonical tensor decomposition, which has shown the efficiency compared to other decompositions. The 3DLUTs compression ratio has been evaluated for the gamut mapping operation, which is an essential part of any display or printing device. As a result of the comparison it has been obtained that in absence of visible distortions a significant compression (six times compression for LUT $17 \times 17 \times 17$) can be achieved. The specific feature of the proposed approach is that this method permits to restore a single element of the original table without the need of complete LUT decompression, which greatly simplifies the hardware implementation.

Keywords: tensor decomposition; lookup table; 3DLUT

For citation: Makarov D.V., Tchobanou M.K. 3D lookup table compression with use of a tensor decomposition. *Proc. Univ. Electronics*, 2018, vol. 23, no. 6, pp. 605–614. DOI: 10.24151/1561-5405-2018-23-6-605-614

Введение. Табулированные таблицы (lookup tables, LUT) применяются для ускорения сложных вычислительных операций в устройствах вывода для выполнения цветовых преобразований, в частности для преобразования цветового охвата при выводе изображений и видео.

Пусть требуется ускорить вычисление функции $y_i = f(x_i)$, где x_i – дискретная величина с ограниченной областью определения. Тогда для всех значений x_i можно заранее вычислить значения функции f и сохранить в массиве (таблице). При необходимости вычисления значения функции $f(x_i)$ из таблицы выбирается результат y_i по соответствующему индексу i . Таким образом, за счет увеличения затрат памяти на хранение таблицы достигается выигрыш в скорости вычисления.

Для обработки изображений часто используются сложные цветовые преобразования, требующие значительных вычислительных затрат. При таких преобразованиях табулируемая функция зависит от трех параметров (так как каждая точка изображения обычно представляется набором из красной, зеленой и синей компонент (RGB)) и имеет общий вид

$$y_i = f(r_i, g_i, b_i),$$

где r_i, g_i, b_i – исходные цветовые компоненты изображения.

Для ускорения вычисления в рассматриваемом случае используются трехмерные табулированные таблицы (3D lookup tables, 3 DLUT) [1]. Примерами таких преобразований могут служить получение изображения в оттенках серого из цветного, преобразование цветового пространства, например RGB в CMYK (Cyan, Magenta, Yellow, Black – голубой, пурпурный, желтый, черный), преобразование цветового охвата, т.е. набора различных цветов, которые способно отобразить устройство вывода, и др.

В случае преобразования цветного RGB-изображения, например при изменении цветового пространства, потребуется три 3DLUT (по одной таблице на каждую цветовую компоненту). При этом расход памяти становится значительным: для 24-битного изображения размер LUT составит $255 \times 255 \times 255 \times 3$ байт (≈ 47 Мбайт), что существенно ограничивает практическую полезность таких таблиц. Чтобы уменьшить объем 3DLUT, можно использовать равномерно прореженные таблицы и при необходимости интерполировать недостающие элементы. Метод интерполяции и размер прореженной 3DLUT выбираются в зависимости от ошибки, которую вносит интерполяция. Это, соответственно, налагает ограничение на преобразование f , для которого может быть применена таблица. Преобразование должно быть непрерывным, иначе интерполяция будет вносить значительную ошибку в те места, где функция имеет разрыв. Многие цветовые преобразования являются непрерывными, и использование таблиц размера $17 \times 17 \times 17$ с применением трилинейной или тетраэдральной интерполяции [2] дает приемлемый результат. Табулированные таблицы такого размера применимы в аппаратной реализации и обеспечивают существенное ускорение вычисления сложной операции без внесения значительных искажений.

Несмотря на эффективность подхода с прореживанием и интерполяцией, размер 3DLUT остается значительным: $17 \times 17 \times 17 \times 3$ байт (≈ 14 Кбайт). Кроме того, бывает необходимо иметь несколько таких таблиц, так как цветовые преобразования могут зависеть от дополнительных параметров, таких как текущая яркость дисплея, яркость и цвет окружающего освещения и др. В связи с этим возникает необходимость периодической пересылки таблицы из оперативной памяти в аппаратный модуль применения табулированных таблиц или выделения дополнительного модуля быстродействующего ПЗУ для хранения всех таблиц, чтобы избежать пересылки.

В настоящей работе предлагается метод аппроксимации 3DLUT с помощью тензорной декомпозиции. Такой подход позволяет дополнительно уменьшить объем 3DLUT без внесения значительных искажений, но с увеличением количества вычислений. Уменьшение объема LUT снижает задержку на пересылку LUT из оперативной памяти, что помогает избежать снижения числа кадров в секунду (frames per second, FPS) при отображении изображений/видео. В случае использования дополнительного ПЗУ предлагаемый метод сжатия удешевляет аппаратный модуль за счет уменьшения емкости ПЗУ.

В отличие от существующих подходов для сжатия LUT предлагаемый метод дает возможность получить любой элемент исходной таблицы без необходимости полного восстановления всей таблицы [3]. Кроме того, метод универсален и не привязан к конкретному аналитическому преобразованию [4] для построения 3DLUT и фактически может быть применен для сжатия любой трехмерной таблицы, элементы которой коррелированы. Также в отличие от алгоритмов, использующих неравномерные 3DLUT [5], процедура восстановления удобна для аппаратной реализации.

Сжатие табулированных таблиц с помощью тензорной декомпозиции. Рассмотрим и сравним методы тензорной декомпозиции, чтобы выбрать эффективный для сжатия 3DLUT. Под тензором будем понимать любой многомерный (d -мерный) массив чисел $T(i_1, i_2, i_3, \dots, i_d)$ размера $n_1 \times n_2 \times n_3 \times \dots \times n_d$, где i_k – соответствующие целочисленные индексы элементов массива, принимающие значения от 0 до n_k . Примерами тензоров могут служить: скалярная величина – тензор размерности 0; вектор – тензор размерности 1; матрица – тензор размерности 2.

Тензорная декомпозиция используется для представления исходного тензора в виде структурированного набора тензоров. Это может потребоваться, например, для снижения размерности исходного тензора, т.е. тензорная декомпозиция работает аналогично методу главных компонент [6]. Снижение размерности полезно для визуализации многомерных данных в двумерном или трехмерном пространстве, при анализе данных и машинном обучении. Кроме того, в реальных задачах существуют тензоры очень большой размерности, например тензор с показаниями, снимаемыми с множества датчиков для нескольких объектов в разное время с разной частотой и т.д. Тензоры такого объема невозможно разместить в памяти для манипуляций с данными, так как число элементов равно n^d (пусть $n_1 = n_2 = n_3 = \dots = n_d = n$) и растет экспоненциально с увеличением размерности. Так, для $n = 2$ при $d = 50$ необходима память емкостью 9 Пбайт. Проблема больших тензоров получила название «проклятье размерности», и для решения этой проблемы применяют тензорные декомпозиции [7]. Рассмотрим базовые методы тензорных декомпозиций, которые находят наиболее широкое применение.

Каноническая декомпозиция (Canonical Polyadic Decomposition, CPD) [8], или CANDECOMP, или PARAFAC, позволяет представить тензор T любой размерности в виде набора двумерных тензоров (матриц):

$$T(i_1, i_2, \dots, i_d) = \sum_{r=1}^R U_1(i_1, r)U_2(i_2, r)\dots U_d(i_d, r), \quad (1)$$

где $T(i_1, i_2, \dots, i_d)$ – исходный тензор размера $n_1 \times n_2 \times n_3 \times \dots \times n_d$; U_k – двумерный тензор размера $n_k \times r$; R – ранг декомпозиции, также называемый каноническим рангом.

На рис.1 представлена визуализация канонической декомпозиции для трехмерного и двумерного случаев.

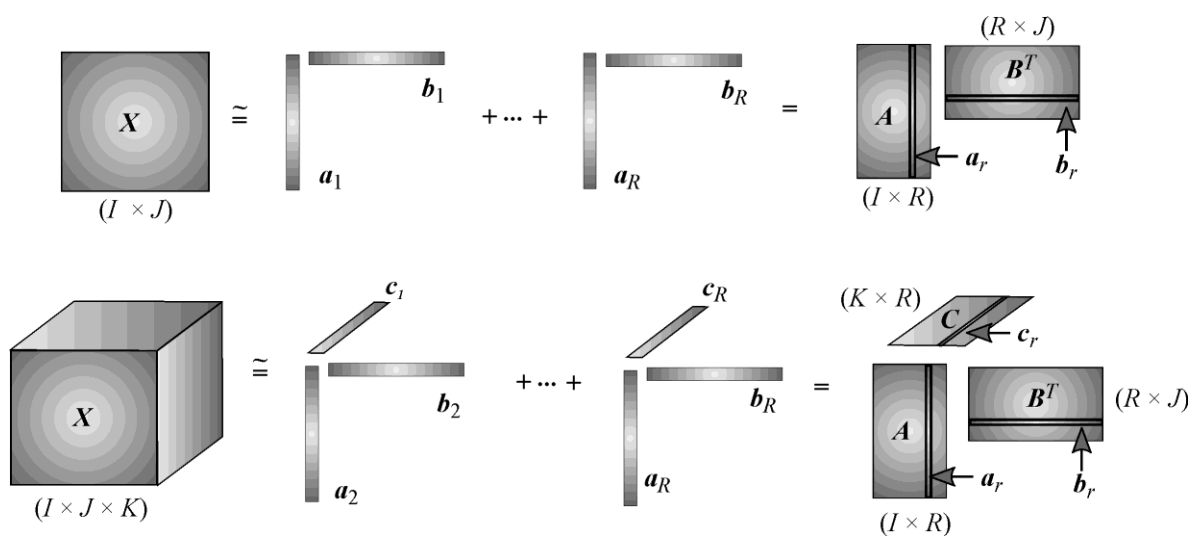


Рис.1. Визуализация канонической декомпозиции для трехмерного и двумерного случаев
 Fig.1. Visualization of canonical decomposition of 2D and 3D cases

Точность канонической декомпозиции зависит от выбранного ранга R . Если выбран небольшой ранг, то говорят о «низкоранговой аппроксимации» и элементы аппроксимированного тензора могут значительно отличаться от элементов исходного. От выбора ранга зависят размер аппроксимации, т.е. число матриц U_k . Также от ранга зависят ошибка аппроксимации и количество операций сложения и умножения при восстановлении тензора. Чем больше ранг, тем больше размер, точность аппроксимации и количество вычислений, необходимых для восстановления элементов исходного тензора.

Декомпозиция Такера [9], или *HOSVD*, *MLSVD*, позволяет представить исходный тензор T размерности d в виде произведения «ядра» G и матриц U_1, U_2, \dots, U_d . Например, декомпозиция трехмерного тензора имеет вид

$$T(i, j, k) = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} G(r_1, r_2, r_3) U_1(i, r_1) U_2(j, r_2) U_3(k, r_3),$$

где R_1, R_2, \dots, R_d – ранги декомпозиции; $G(r_1, r_2, \dots, r_d)$ – элемент тензора G размера $R_1 \times R_2 \times \dots \times R_d$; U_k – матрицы размера $n_k \times R_k$; n_k – соответствующий размер исходного тензора.

На рис.2 представлена визуализация декомпозиции для трехмерного случая.

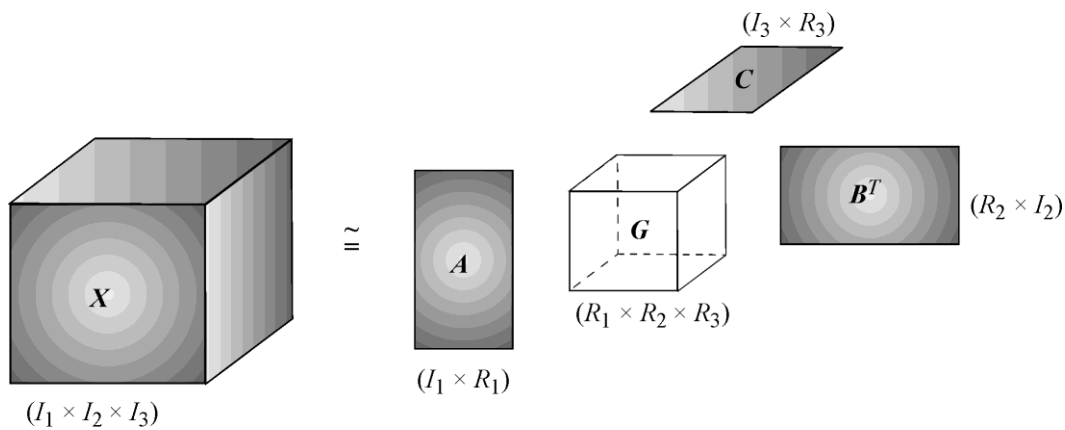


Рис.2. Визуализация декомпозиции Такера для трехмерного случая
Fig.2. Visualization of Tucker decomposition for 3D case

Преимуществом декомпозиции Такера по сравнению с канонической является наличие стабильного алгоритма вычисления. Для получения декомпозиции используется Singular Value Decomposition (SVD), следовательно возникает ограничение ортогональности на матрицы U_k . Однако такой метод декомпозиции практичен только для тензоров «небольшой» размерности ($d \leq 3$). Это ограничение связано с оценкой количества элементов в декомпозиции $O(dnR + R^d)$. Видно, что размер декомпозиции зависит от d экспоненциально.

Декомпозиция цепочкой тензоров (*Tensor-Train*, *TT*) [10] заключается в представлении тензора большой размерности набором трехмерных тензоров:

$$T(i_1, i_2, \dots, i_d) = \sum_{r_1, r_2, \dots, r_{d-1}}^{R_1, R_2, \dots, R_{d-1}} G^{(1)}(1, i_1, r_1) G^{(2)}(r_1, i_2, r_2) \dots G^{(d)}(r_{d-1}, i_d, 1),$$

где R_1, R_2, \dots, R_{d-1} – ранги декомпозиции; $G^{(k)}(r_{k-1}, i_k, r_k)$ – элемент трехмерного тензора размера $R_{k-1} \times n_k \times R_k$.

Декомпозиция цепочкой тензоров получается избыточной по сравнению с канонической из-за введения дополнительного индекса, но это позволяет получить стабильный алгоритм ее вычисления, основанный на разложении по собственным векторам (SVD [6]). В отличие от декомпозиции Такера тензоры большой размерности имеют более компактную декомпозицию.

Оценим эффективность описанных методов на примере декомпозиции 3DLUT для сложной нелинейной операции преобразования цветового охвата [11], для выполнения которой в реальном времени используют 3DLUT. Данное преобразование является необходимым для любого устройства вывода информации. Для проведения экспериментов построена трехмерная таблица размера $17 \times 17 \times 17$ для преобразования цветового охвата DCI-P3 в sRGB по методу SCLIP [11, с. 187]. Такая трехмерная таблица необходима для каждой цветовой компоненты RGB-изображения. Для оценки эффективности построим зависимость среднеквадратичной ошибки (Mean Squared Error, MSE) от степени аппроксимации K_a . MSE рассчитывается по формуле

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (t_i - \hat{t}_i)^2,$$

где n – общее число элементов исходной таблицы; t_i – элемент исходной таблицы; \hat{t}_i – соответствующий элемент таблицы, восстановленной после декомпозиции.

Степень аппроксимации определяется как

$$K_a = \frac{N_s}{N_d},$$

где N_s – число элементов в исходной таблице; N_d – число элементов после декомпозиции.

На рис.3 приведено сравнение эффективности методов декомпозиции для аппроксимации 3DLUT размера $17 \times 17 \times 17$ (аппроксимировалась таблица, соответствующая преобразованию зеленой цветовой компоненты).

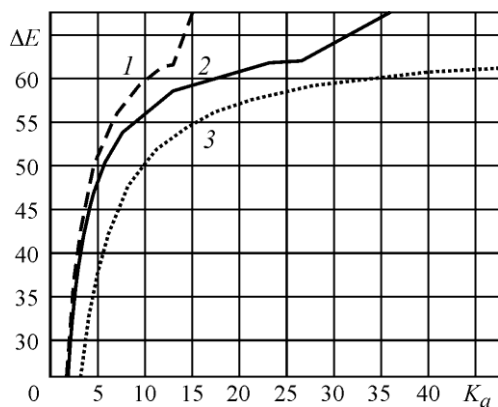


Рис.3. Зависимость искажений MSE от степени аппроксимации K_a 3DLUT размера $17 \times 17 \times 17$ с помощью декомпозиции Такера (кривая 1), декомпозиции цепочкой тензоров (кривая 2), канонической декомпозиции (кривая 3)

Fig.3. Graphs of distortion MSE versus approximation ratio K_a for decomposition of $17 \times 17 \times 17$ 3DLUT with Tucker decomposition (curve 1), TT decomposition (curve 2), canonical decomposition (curve 3)

В результате проведенного тестирования самым эффективным оказался метод канонической декомпозиции. Поэтому для сжатия 3DLUT предлагается использовать именно эту декомпозицию. Каноническая декомпозиция редко применяется на практике, так как нет стабильного алгоритма для получения декомпозиции. Существующие алгоритмы основаны на инициализации декомпозиции случайными числами и последующем оптимизационном процессе. Поэтому может потребоваться множество запусков алгоритма, прежде чем будет достигнут приемлемый результат. Но в рассматриваемом случае вычисление аппроксимации и подготовка 3DLUT происходят заранее и лимит времени на это можно считать неограниченным. Кроме того, каноническая декомпозиция не имеет ограничения ортогональности для

матриц U_k , что позволяет достичь наиболее компактной аппроксимации по числу элементов $O(dnR)$, и характеризуется наименьшей вычислительной сложностью восстановления исходного элемента тензора.

Для вычисления канонической декомпозиции можно использовать MatLab-библиотеку методов с открытым исходным кодом TensorLab [12]. Для получения результатов в работе применяется метод `cpd` из TensorLab версии 3.0. После вычисления аппроксимации необходимо преобразовать полученные значения в целые числа, чтобы уменьшить занимаемый ими объем и найти ускорение за счет целочисленных операций. В работе используется равномерное квантование в соответствии с заданной разрядностью, один разряд при этом отводится под знак числа. Квантованное значение рассчитывается по формуле

$$\hat{v}_i = v_i \left(\frac{\max_j(v_j) - \min_j(v_j)}{2^k} \right)^{-1}.$$

Оценка искажений при цветовых преобразованиях посредством 3DLUT обычно проводится с помощью ΔE [13], поэтому будем использовать эту метрику, которая отражает заметность искажений (при $\Delta E < 3$ искажения считают незаметными). Для ее расчета необходимо выполнить преобразование эталонных и искаженных RGB-компонент в цветовое пространство Lab [13] и вычислить евклидово расстояние:

$$\Delta E = \sqrt{(L_s - L_d)^2 + (a_s - a_d)^2 + (b_s - b_d)^2},$$

где (L_s, a_s, b_s) и (L_d, a_d, b_d) – значения в пространстве Lab, соответствующие эталонному и искаженному RGB.

Рассмотрим предлагаемый алгоритм сжатия 3DLUT с помощью канонической декомпозиции. Входные данные для алгоритма аппроксимации следующие: исходные 3DLUT для каждой цветовой компоненты; пороговое значение искажений по ΔE ; разрядность для квантования значений аппроксимации b_d .

Алгоритм вычисления аппроксимации имеет вид:

Шаг 1. Задать ранг декомпозиции $R = 1$.

Шаг 2. Увеличить R на 1.

Шаг 3. Задать количество попыток декомпозиции $T = 1000$.

Шаг 4. Вычислить каноническую декомпозицию с рангом R для всех входных 3DLUT.

Шаг 5. Выполнить равномерное квантование результата декомпозиции для заданной разрядности.

Шаг 6. Восстановить 3DLUT из аппроксимированных и квантованных значений.

Шаг 7. Оценить искажения ΔE по сравнению с исходными 3DLUT.

Шаг 8. Запомнить наилучший результат.

Шаг 9. Уменьшить значение T и перейти к шагу 4, если $T > 0$.

Шаг 10. Если значение искажений по ΔE меньше порогового, то завершить алгоритм, иначе – перейти к шагу 2.

Как показала практика, достаточно 5-10 запусков алгоритма канонической декомпозиции, чтобы получить более компактную аппроксимацию при одинаковых искажениях по сравнению с методами декомпозиции цепочкой тензоров и декомпозицией Такера. Тем не менее в алгоритм заложено большое число попыток (см. шаг 3), чтобы повысить вероятность нахождения наилучшего результата. Восстановление элемента

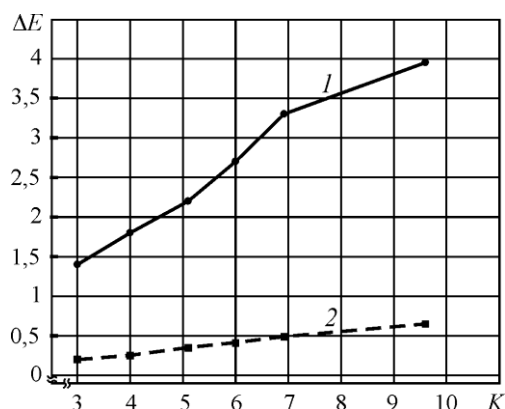


Рис.4. Зависимость степени искажения ΔE от степени сжатия K при аппроксимации 3DLUT размера $17 \times 17 \times 17$ для преобразования цветового охвата: 1 – максимальное значение ΔE ; 2 – среднее значение ΔE

Fig.4. Graphs of distortion ΔE versus compression ratio K for gamut mapping $17 \times 17 \times 17$ 3DLUT: 1 – maximum ΔE ; 2 – average ΔE

аппроксимированной 3DLUT происходит с использованием формулы (1). Следует отметить, что с уменьшением ранга R не только повышается степень сжатия, но и снижается количество операций для восстановления исходного элемента, поэтому нужно стремиться получить декомпозицию с наименьшим рангом. Для удобства аппаратной реализации можно ввести дополнительное ограничение на разрядность регистра аккумулятора, используемого при восстановлении элемента 3DLUT по формуле (1). При суммировании регистр не должен переполняться. Соответственно, в описанный алгоритм следует добавить дополнительный шаг для проверки переполнения. Время получения декомпозиции для 3DLUT, удовлетворяющей входным требованиям ΔE и разрядности, составило приблизительно 10 мин на PC с процессором Intel Core i5-4670S 3ГГц и емкостью оперативной памяти 8 Гбайт.

На рис.4 представлена зависимость степени искажений от степени сжатия 3DLUT размера $17 \times 17 \times 17$ для преобразования цветового охвата. Степень сжатия оценивалась как

$$K = K_a \frac{b_s}{b_d},$$

где b_s – разрядность исходных значений; b_d – разрядность квантования сжатых значений 3DLUT.

Для оценки цветовых искажений, вносимых сжатием, применялись две табулированные таблицы – исходная и восстановленная после сжатия – для всех возможных RGB-значений. Затем оценивалось ΔE . Из рис.4 видно, что с помощью предложенного подхода можно значительно уменьшить объем 3DLUT. Трехмерная таблица размера $17 \times 17 \times 17$ может быть сжата без видимых искажений в шесть раз. Также отметим, что среднее значение ΔE значительно меньше максимального ΔE , следовательно большинство значений искажено незначительно.

Заключение. Предложенный метод сжатия 3DLUT, основанный на использовании канонической тензорной декомпозиции, показал свою эффективность. Метод позволяет существенно уменьшить размер 3DLUT за счет увеличения количества вычислений, необходимых для восстановления элемента. Преимуществом метода является возможность восстановления любого элемента таблицы независимо от других, в отличие от классических алгоритмов компрессии, что особенно важно для аппаратной реализации алгоритма. Метод универсален и может быть применен для сжатия любых трехмерных таблиц, в которых существует корреляция между элементами. Кроме того, размещение в памяти и процедура восстановления элемента фиксированы для заданной степени сжатия, что позволяет зафиксировать и аппаратную реализацию. Также процедура восстановления исходного элемента может быть ускорена за счет использования параллельных вычислений.

Литература

1. **Kang H.R.** Color technology for electronic imaging devices // SPIE Press. – 1997. – Vol. PM28 of Press Monographs (Book 28). – 384 p.
2. **Sun B., Zhou S.** Study on the 3d interpolation models used in color conversion // International J. of Engineering and Technology. – 2012. – Vol. 4. – P. 57–60.
3. **Aravindh Balaji, Gaurav Sharma, Mark Q. Shaw, Randall Guay.** Hierarchical compression of color look up tables // 15th Color Imaging Conference Final Program and Proceedings. – 2007. – P. 261–266.
4. **Yun-Tae Kim, Yang Ho Cho, Cheol-Hee Lee, Yeong-Ho Ha.** Color look-up table design for gamut mapping and color space conversion // IS&Ts International Conference on Digital Production Printing and Industrial Applications. – 2003. – P. 28–29.
5. **Monga V., Bala R.** Algorithms for color look-up-table (lut) design via joint optimization of node locations and output values // 2010 IEEE International Conference on Acoustics, Speech and Signal Processing. – 2010. – P. 998–1001.
6. **Strang G.** Introduction to linear algebra. – Wellesley: Wellesley-Cambridge Press, 2016. – 584 p.
7. Tensor decomposition for signal processing and machine learning / **N.D. Sidiropoulos, Li. De Lathauwer, Xiao Fu et al.** // IEEE Transactions on Signal Processing. – 2016. – Vol. 65. – P. 3551–3582.
8. **Kolda T., Bader B.** Tensor decompositions and applications // SIAM Rev. – 2009. – Vol. 51. – P. 455–500.
9. **Tucker L.R.** Some mathematical notes on three-mode factor analysis // Psychometrika. – 1966. – Vol. 31. – No. 5. – P. 279–311.
10. **Oseledets I.V.** Tensor-train decomposition // SIAM J. SCI. COMPUT. – 2011. – Vol. 33. – No. 5. – P. 2295–2317.
11. **Golub G., Van Loan C.** Matrix computations. – Johns Hopkins University Press, 1996. – 728 p.
12. **Morovic J.** Color gamut mapping. Wiley-IS & T series in imaging science and technology. – Hoboken: John Wiley & Sons Ltd, 2008. – 302 p.
13. Tensorlab 3.0. – 2016. / **N. Vervliet, O. Debals, L. Sorber et al.** – URL: <https://www.tensorlab.net> (дата обращения: 10.07.2018).
14. **Fairchild M.D.** Color appearance models. Wiley-IS & T series in imaging science and Technology. – New York: John Wiley & Sons Inc, 2013. – 472 p.

Поступила 01.08.2018 г.; принята к публикации 25.09.2018 г.

Макаров Дмитрий Викторович – аспирант кафедры вычислительных машин, систем и сетей Института автоматизации и вычислительной техники Национального исследовательского университета «МЭИ» (Россия, 111250, г. Москва, Красноказарменная ул., д. 14), makarovdmv@gmail.com

Чобану Михаил Константинович – доктор технических наук, профессор кафедры вычислительных машин, систем и сетей Института автоматизации и вычислительной техники Национального исследовательского университета «МЭИ» (Россия, 111250, г. Москва, Красноказарменная ул., д. 14), tchobanou@gmail.com

References

1. Kang H.R. *Color technology for electronic imaging devices*. SPIE Press, 1997. Vol. PM28 of Press Monographs (Book 28). 384 p.
2. Sun B., Zhou S. Study on the 3d interpolation models used in color conversion. *International Journal of Engineering and Technology*. 2012, vol. 4, pp. 57–60.
3. Aravindh Balaji, Gaurav Sharma, Mark Q. Shaw, Randall Guay. Hierarchical compression of color look up tables. *15th Color Imaging Conference Final Program and Proceedings*, 2007, pp. 261–266.
4. Yun-Tae Kim, Yang Ho Cho, Cheol-Hee Lee, Yeong-Ho Ha. Color look-up table design for gamut mapping and color space conversion. *IS&Ts International Conference on Digital Production Printing and Industrial Applications*, 2003, pp. 28–29.
5. Monga V., Bala R. Algorithms for color look-up-table (lut) design via joint optimization of node locations and output values. *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 998–1001.

6. Strang G. *Introduction to Linear Algebra*. Wellesley, Wellesley-CambridgePress, 2016. 584 p.
7. Nicholas D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu et al. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 2016, vol. 65, pp. 3551–3582.
8. Kolda T., Bader B. Tensor decompositions and applications. *SIAMRev*, 2009, vol. 51, pp. 455–500.
9. Tucker L.R. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 1966, vol. 31, no. 5, pp. 279–311.
10. Oseledets I.V. Tensor-train decomposition. *SIAM J. SCI. COMPUT.*, 2011, vol. 33, no. 5, pp. 2295–2317.
11. Golub G., Van Loan C. *Matrix computations*. Johns Hopkins University Press, 1996. 728 p.
12. Morovic J. *Color Gamut Mapping*. Wiley-IS & T Series in Imaging Science and Technology, Hoboken, John Wiley & Sons Ltd, 2008. 302 p.
13. Vervliet N., Debals O., Sorber L. et al. *Tensorlab 3.0*, 2016. Available at: <https://www.tensorlab.net> (accessed: 10.07.2018).
14. Fairchild M.D. *Color appearance models*. Wiley-IS & T Series in Imaging Science and Technology. New York, John Wiley & Sons Inc, 2013. 472 p.

Submitted 01.08.2018; Accepted 25.09.2018.

Information about the authors:

Dmitry V. Makarov – PhD student of the Computing Machines, Systems and Networks Department of the Institute of Automatics and Computer Engineering, National Research University «Moscow Power Engineering Institute» (Russia, 111250, Moscow, Krasnokazarmennaya, 14), makarovdmv@gmail.com

Mikhail K. Tchobanou – Dr. Sci. (Eng.), Prof. of the Computing Machines, Systems and Networks Department of the Institute of Automatics and Computer Engineering, National Research University «Moscow Power Engineering Institute» (Russia, 111250, Moscow, Krasnokazarmennaya, 14), tchobanou@gmail.com

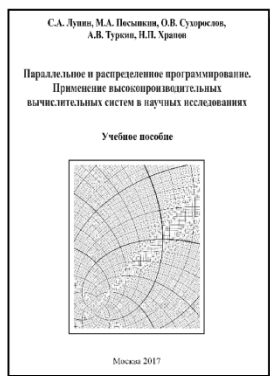


Издательско-полиграфический комплекс
Национального исследовательского университета «МИЭТ»
информирует



Вышло в свет учебное пособие

Лупин С.А., Посыпкин М.А., Сухорослов О.В., Туркин А.В., Храпов Н.П.
Параллельное и распределенное программирование. Применение высокопроизводительных вычислительных систем в научных исследованиях: учеб. пособие. - М.: МИЭТ, 2017. - 160 с.: ил.



ISBN 978-5-7256-0865-6

Рассмотрены технологии, ориентированные на системы с распределенной памятью. К этому классу относятся кластеры и грид-системы, объединяющие отдельные вычислительные модули, каждый из которых обладает собственной памятью. Приведены примеры разработки и развертывания распределенных грид-приложений.

Для студентов и специалистов в области информационных технологий.