

УДК 519.6, 51-7

АППРОКСИМАЦИЯ ИЗОБРАЖЕНИЙ В ОГРАНИЧЕННОМ КЛАССЕ ГРАФИЧЕСКИХ ПРИМИТИВОВ

Филиппов Семён Сергеевич¹, Ершов Николай Михайлович²

¹Студент;

Государственный университет «Дубна»;

Россия, 141980, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: fss.21@uni-dubna.ru.

²Кандидат физико-математических наук, доцент;

Московский государственный университет им. М. В. Ломоносова;

Россия, 119991, г. Москва, ул. Ленинские горы, 1;

e-mail: ershov@cs.msu.ru.

Настоящая статья посвящена разработке и исследованию унифицированного подхода к аппроксимации растровых изображений наборами графических примитивов одного класса с использованием методов многомерной непрерывной оптимизации. Задача аппроксимации изображений поставлена, как задача поиска оптимального набора параметров примитивов, максимизирующей целевую функцию на основе одной из метрик, используемых для определения схожести двух заданных изображений. Практическая реализация на основе предложенного подхода представляет собой модульную программную систему на языке программирования Python с графическим и командным интерфейсами, функциями отрисовки, сохранения/загрузки результатов и генерации анимированных изображений. В ходе экспериментов исследовалась эффективность используемых алгоритмов и влияние их параметров и настроек системы на качество аппроксимации.

Ключевые слова: аппроксимация изображений, непрерывная оптимизация, имитация отжига, метод роя частиц, стохастический поиск.

Для цитирования:

Филиппов С. С., Ершов Н. М. Аппроксимация изображений в ограниченном классе графических примитивов // Системный анализ в науке и образовании: сетевое научное издание. 2025. № 2. С. 14-21. EDN: UOOZDK . URL : <https://sanse.ru/index.php/sanse/article/view/660>.

IMAGE APPROXIMATION IN A LIMITED CLASS OF GRAPHIC PRIMITIVES

Filippov Semen S.¹, Ershov Nikolay M.²

¹Student;

Dubna State University;

19 Universitetskaya st., Dubna, 141980, Russia;

e-mail: fss.21@uni-dubna.ru.

²PhD in Physical and Mathematical Sciences, associate professor;

Lomonosov Moscow State University;

1 Leninskiye Gory, Moscow, 119991, Russia;

e-mail: ershov@cs.msu.ru.

The paper is devoted to the development and study of a unified approach to approximating raster images with sets of graphic primitives of one class using multidimensional continuous optimization methods. The problem of image approximation is posed as a problem of finding an optimal set of primitive parameters that maximizes the objective function based on one of the metrics used to determine the similarity of two given images. Practical implementation based on the proposed approach is a modular software system in the Python programming language with graphical and command interfaces, functions for drawing,



Статья находится в открытом доступе и распространяется в соответствии с лицензией Creative Commons «Attribution» («Атрибуция») 4.0 Всемирная (CC BY 4.0) <https://creativecommons.org/licenses/by/4.0/deed.ru>

saving/loading results, and generating animated images. During the experiments, the efficiency of the algorithms used and the influence of their parameters and system settings on the quality of approximation were studied.

Keywords: image approximation, continuous optimization, simulated annealing, particle swarm optimization, stochastic search.

For citation:

Filippov S. S., Ershov N. M. Image approximation in a limited class of graphic primitives. *System analysis in science and education*, 2025;(2):14-21 (in Russ). EDN: UOOZDK. Available from: <https://sanse.ru/index.php/sanse/article/view/660>.

Введение

Настоящее исследование посвящено вопросам разработки и программной реализации унифицированного подхода к решению задачи реконструкции заданного изображения набором графических объектов (примитивов) предопределённой структуры. Так как задача реконструкции изображений обычно сводится к оптимизационной задаче подбора параметров графических примитивов с целью нахождения наилучшего приближения к оригинальному изображению, то рассматриваемую задачу иногда называют задачей аппроксимации изображений.

В различных вариациях задачи реконструкции и аппроксимации изображений решаются с тем или иным успехом на протяжении иногда уже многих столетий. Прежде всего речь идёт об области технологии воспроизведения изображений с использованием разнообразных физических носителей: мозаика [1], вышивка [2], плетение (текстиль) [3], полиграфия [4]. Подобные технологии применяются даже в изобразительном искусстве, характерным примером этого является пуантилизм — живописная манера, основанная на технике письма мелкими точечными мазками почти чистых красок [5], изобретателем которой был французский живописец Жорж Сёра.

Востребованной данная тематика остаётся и в современном декоративно-прикладном искусстве, когда для воспроизведения изображений применяются средства и технологии, для этого изначально совсем не предназначенные. Примерами такого подхода являются *ASCII Art* – генерация изображений с помощью печатных символов [6; 7]; *String Art*, где изображение формируется набором нитей, натянутых на расположенные по периметру изображения шпильки [8; 9; 10]; *Shadow Art* – изображение (или даже серия изображений) формируется тенями от некоторого набора трёхмерных объектов [11; 12; 13] и т. д. Близкой к таким приложениям является и область дизайна, где подобные методы применяются в качестве средств стилизации заданных изображений под манеру того или иного художника, или под некоторую технику или стиль «рисования» [14; 15; 16].

В области компьютерной графики аналогичная задача называется задачей векторизации растровых изображений, т. е. задачей преобразования последних в векторную форму [17]. Одной из целей векторизации является сокращение объёма изображения. Кроме того, векторные изображения позволяют эффективно выполнять над ними различные модификации (масштабирование, повороты, нелинейные преобразования) без потери качества изображения, что делает их предпочтительным вариантом, например, в задачах компьютерного дизайна.

1. Задача аппроксимации изображений

Аппроксимацию (приближение) растрового изображения с использованием набора одного класса геометрических примитивов можно представить как задачу многомерной оптимизации. Целью является поиск набора параметров, описывающих положение, форму, размер, цвет набора примитивов, соответствующих изображению, наиболее близкому к заданному целевому изображению в соответствии с выбранной мерой схожести изображений. Целевое изображение T представлено в виде матрицы значений интенсивности пикселей.

Каждый примитив определяется вектором параметров фиксированной длины. Полный набор S параметров всех N примитивов образует вектор состояния, точку в пространстве возможных решений. Визуализация набора параметров S в приближаемое изображение описывается процедурой R отрисовки N графических примитивов на фоне холста белого цвета: $A = R(S)$.

Численная оценка сходства между сгенерированным изображением и целевым вычисляется с помощью метрики схожести изображений $M(T, A)$. В настоящем исследовании в качестве таких метрик использовались: среднеквадратичная ошибка (MSE), пиковое отношение сигнала к шуму ($PSNR$) и индекс структурного сходства ($SSIM$). Целевая функция $E(S)$ определяет качество приближения набора параметров S . Целевая функция вычисляется на основе выбранной метрики M . Для унификации все три метрики нормализуются в интервал $[0, 1]$ и преобразуются для максимизации, т. е. чем больше величина $E(S)$, тем лучше аппроксимация. Таким образом, задачу приближения изображения можно определить как поиск оптимального набора параметров S_0 , который максимизирует значение целевой функции $E(S)$ в пространстве допустимых решений: $S_0 = \text{argmax } E(S)$.

Пространство решений имеет чрезвычайно большую размерность и сложную, нелинейную и многоэкстремальную поверхность целевой функции $E(S)$, что делает аналитическое решение или полный перебор вариантов практически невозможным. Поэтому для поиска близкого к оптимальному решения применяются итерационные методы стохастической оптимизации. В настоящей работе для решения поставленной задачи оптимизации применялись следующие три метода, хорошо зарекомендовавшие себя при решении многомерных задач непрерывной оптимизации: метод поиска восхождением к вершине (*Hill Climbing, HC*) [18]; метод имитации отжига (*Simulated Annealing, SA*) [19]; метод роя частиц (*Particle Swarm Optimization, PSO*) [20].

2. Параметризация графических примитивов

Теоретически, в качестве графических примитивов могут выступать любые геометрические объекты произвольной сложности, вплоть до изображений символов заданного шрифта. В настоящей работе исследовались следующие базовые графические примитивы: круг, линия, прямоугольник и треугольник. Данные примитивы являются основными «строительными блоками» во многих задачах компьютерной графики и векторных представлений. Также они часто используются в существующих системах аппроксимации и генеративного искусства. Каждый из выбранных примитивов имеет простое математическое описание и небольшое (и фиксированное) количество параметров (см. рис. 1), что упрощает их параметризацию, рисование и оптимизацию. В то же время, такой набор классов позволяет аппроксимировать широкий спектр визуальных форм.

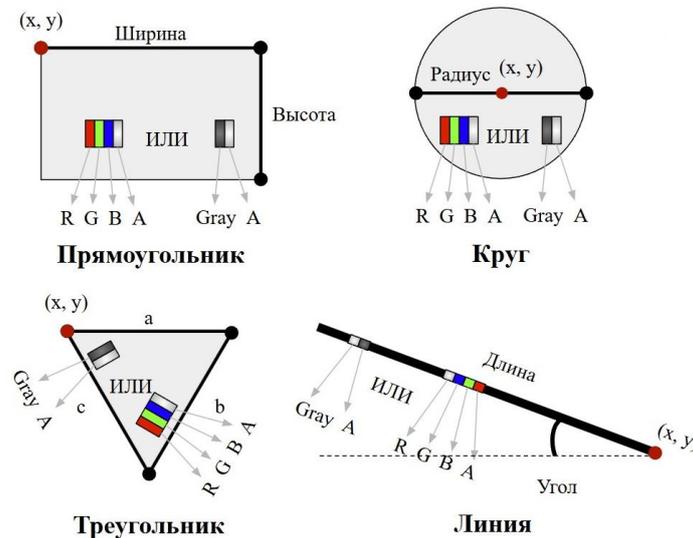


Рис. 1. Параметрическое описание базовых графических примитивов

Параметры каждого примитива в общем случае задают его местоположение на рисунке, размеры, форму и цвет в выбранном цветовом пространстве. Однако, в каждом конкретном случае часть этих параметров может быть зафиксирована пользователем и не подвергаться дальнейшей оптимизации. Например, мы можем заранее расположить примитивы на плоскости рисунка по какому-то предопределённому алгоритму, тогда параметры местоположения не будут меняться в процессе оптимизации. Аналогичным образом можно зафиксировать цвет, например, аппроксимировать чёрно-

белое изображение линиями чёрного цвета. Фиксация части параметров снижает размерность задачи и за счёт этого, теоретически, может приводить к ускорению работы выбранного метода оптимизации.

3. Программная реализация

Для программной реализации системы аппроксимации изображений брались во внимание скорость разработки, доступность библиотек для научных вычислений и обработки изображений, производительность важных участков кода и кроссплатформенность. В качестве основного языка программирования был выбран *Python*, так как он обладает большим количеством библиотек для научных вычислений и обработки изображений, а также позволяет быстро реализовывать и тестировать алгоритмы. Для работы с многомерными массивами, в которых можно представить изображения, и численной оптимизации, *NumPy* является основной библиотекой. Для ускорения вычислений в *Python* в критически важных по производительности участках кода используется пакет *Numba*. Это позволяет получать лучшую производительность в коде с массивами, функциями *NumPy* и циклами, близкую к компилируемому языку типа *C++*.

Для создания кроссплатформенного графического пользовательского интерфейса (*GUI*) была выбрана библиотека *PyQt6*, являющаяся набором привязок *Python* к кроссплатформенной библиотеке *Qt*, которая даёт возможность создать нативный интерфейс для различных операционных систем (*Windows, MacOS, Linux*).

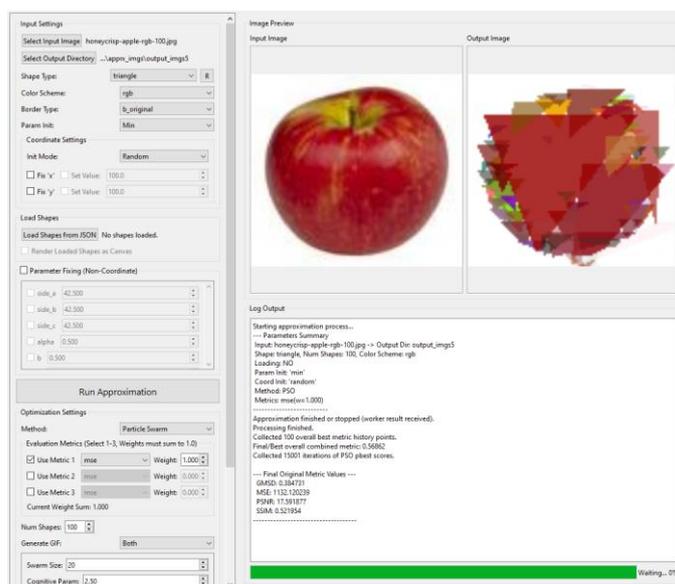


Рис. 2. Графический интерфейс программной системы

Разработанная программная система предоставляет два интерфейса для взаимодействия с пользователем: графический интерфейс (*GUI*) для интерактивной настройки и визуализации процесса аппроксимации (рис. 2), и интерфейс командной строки (*CLI*) для автоматизации и пакетной обработки.

Модульность программной системы обеспечивает упрощение изменения кода, лёгкую расширяемость на новые классы примитивов, метрики или алгоритмы. Также это позволяет упростить тестирование, так как все отдельные модули системы могут быть протестированы независимо.

4. Численное исследование

С помощью разработанной программной системы были проведены масштабные исследования различных характеристик процесса решения задачи аппроксимации изображений. В качестве

тестовых изображений были выбраны классические произведения (или их фрагменты). Все изображения перед использованием были масштабированы к ширине 100 пикселей.

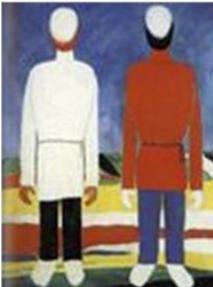
Оригинал	Круг	Линия	Прямоугольник	Треугольник
				
Итоговая E(S)	0,767	0,685	0,818	0,396
Время (сек.)	1682,98	3311,69	1489,33	317,72
MSE	529,310	758,126	402,640	1850,813
PSNR (дБ)	20,894	19,333	22,082	15,457
SSIM	0,674	0,623	0,719	0,436
Размер файла (КБ)	16,76	18,7	11,23	13,93

Рис. 3. Сравнение результатов работы метода имитации отжига с использованием различных графических примитивов

На рис. 3 приведены результаты применения метода имитации отжига при аппроксимации фрагмента картины Казимира Малевича «Две мужские фигуры». Использовалась метрика MSE , число графических примитивов – 200, число итераций алгоритма – 50 000. Лучшие результаты продемонстрировал набор прямоугольных примитивов, что было также подтверждено и другими экспериментами.

Оригинал	MSE	PSNR	SSIM	SSIM(0,8), MSE(0,2)
				
MSE	378,813	366,29	1031,071	387,360
PSNR (дБ)	22,347	22,493	17,998	22,250
SSIM	0,530	0,543	0,632	0,632
Время (сек.)	1615,40	1711,23	2401,25	2732,45

Рис. 4. Влияние различных метрик, применяемых в качестве целевых функций оптимизации, на качество восстановленного изображения

На рис. 4 показаны примеры работы алгоритма имитации отжига, в которых в качестве целевой функции оптимизации применялись различные метрики или их комбинация (последний столбец). В качестве целевого изображения использовалась картина «Крик» Эдварда Мунка. Каждая метрика акцентирует внимание на разных аспектах качества изображения. Результаты этого и подобных ему

экспериментов показали, что метрики MSE и $PSNR$ выдают практически схожие результаты, что обусловлено схожестью внутреннего устройства этих двух метрик. Метрика $SSIM$ хорошо обрабатывает границы объектов на изображениях, но во многих случаях полностью игнорирует большие однотонные участки изображений. Лучшим вариантом с точки зрения качества генерируемых изображений является применение взвешенной комбинации двух метрик MSE и $SSIM$, что обеспечивает корректную обработку и однотонных областей, и их границ. Однако, этот приём существенно увеличивает время работы алгоритма оптимизации.



Метод	HC	SA	PSO	PSO
Метрика(-и) и Вес	SSIM (0,8), MSE (0,2)	SSIM (0,8), MSE (0,2)	SSIM (0,8), MSE (0,2)	SSIM (0,2), MSE (0,8)
MSE	220,511	194,337	9590,622	367,132
PSNR (дБ)	24,697	25,245	8,312	22,483
SSIM	0,752	0,748	0,446	0,588
Время (сек.)	3885,27	4560,78	19403,38	39,331

Рис. 5. Сравнение методов оптимизации

Была проведена серия вычислительных экспериментов по сравнению трёх реализованных в системе методов оптимизации. Пример их работы для изображения «Мона Лиза» Леонардо да Винчи показан на рис. 5. Лучшие в среднем результаты показывает метод имитации отжига. Метод поиска восхождением к вершине выдаёт примерно такого же качества изображения, но работает существенно быстрее. Метод роя частиц работает медленнее всех, прежде всего, за счёт того, что он является популяционным. По этой причине за одно и то же количество обращений к целевой функции (включая рисование примитивов) этот метод показывает и худшие результаты по качеству генерируемых изображений.

Отдельно исследовалось влияние на время работы системы, оказываемое использованием пакета *Numba*. Численные эксперименты показали, что ускорение работы собственно метода оптимизации при использовании *Numba* составляет величину примерно равную 40. Особенно эффективен этот приём оказывается при пакетной обработке изображений, так как в этом случае минимизируются накладные расходы, связанные с компиляцией кода.

Заключение

Разработанный подход и программная система предоставляют широкие возможности для дальнейших исследований и усовершенствований. Одним из направлений является расширение набора используемых графических примитивов. Включение более сложных элементов, таких как эллипсы или кривых Безье и сплайнов, потенциально позволит аппроксимировать более комплексные визуальные формы с использованием меньшего общего количества элементов, что может положительно сказаться на качестве изображений.

Другим возможным направлением исследований является разработка адаптивных и гибридных методов оптимизации. Это включает исследование алгоритмов с динамической адаптацией их гиперпараметров в ходе работы, например, адаптивное изменение коэффициента охлаждения в

методе имитации отжига или корректировка когнитивных и социальных коэффициентов в методе роя частиц в зависимости от условий задачи.

Важной задачей для задачи аппроксимации изображения графическими примитивами является улучшение целевой функции и используемых метрик качества. Целесообразно исследовать применение других современных метрик оценки качества изображений, которые могут быть более устойчивы к множеству типов искажений или лучше коррелировать с восприятием определённых деталей. Разработка методов автоматического подбора весовых коэффициентов для комбинированных метрик, возможно, на основе анализа характеристик самого целевого изображения, могла бы повысить универсальность и эффективность целевой функции.

Ещё одним направлением для развития является оптимизация вычислительно интенсивных участков кода с использованием библиотеки *Numba*, более строгую типизацию данных и профилирование для выявления узких мест. Также следует рассмотреть возможности большего распараллеливания вычислений, что весьма актуально для популяционных методов, таких как метод роя частиц, где оценка каждой частицы производится независимо, или для организации параллельного запуска нескольких независимых прогонов алгоритма имитации отжига с различными начальными условиями или параметрами.

Проведение более расширенного экспериментального исследования позволит глубже понять свойства разработанного подхода. Необходимо тестирование на более разнообразном наборе целевых изображений, включая изображения большего разрешения, чтобы оценить масштабируемость методов. Требуется более систематическое исследование влияния наборов гиперпараметров для каждого из реализованных методов оптимизации.

Практическое применение разработанной системы для решения конкретных прикладных задач, таких как художественная стилизация изображений, генерация векторных представлений для нужд полиграфии или веб-дизайна, или создание уникальных художественных композиций, может выявить новые требования и направления для её совершенствования.

Список источников

1. Di Blasi, G., Gallo, G. Artificial mosaics // *The Visual Computer*. – 2005. – Т. 21. – С. 373-383. – DOI: 10.1007/s00371-005-0292-4.
2. Atkinson K., Roberts J. Graphics and Visualization within Cross-Stitch // *Eurographics UK, Conference Proceedings: 1999*. – Eurographics, 1999. – С. 129-141.
3. Posselt E. A. The Jacquard machine analyzed and explained: with an appendix on the preparation of Jacquard cards. – Philadelphia: Pennsylvania museum and school of industrial art, 1887.
4. Mirseyedi S. Side by Side: The Halftone's Visual Culture of Pragmatism // *History of Photography*. – 2017. – Т. 41, № 3. – С. 286-310. – DOI: 10.1080/03087298.2017.1353208.
5. Власов В. Г. Пуантилизм // *Новый энциклопедический словарь изобразительного искусства* : [в 10 т.]. — СПб.: Азбука-Классика, 2007. – Т. VII. – С. 876.
6. Chung M., Kwon T. Fast Text Placement Scheme for ASCII Art Synthesis // *IEEE Access*. – 2022. – Т. 10. – С. 40677-40686. – DOI: 10.1109/ACCESS.2022.3167567.
7. ASCII art synthesis from natural photographs / Xu X., Zhong L., Xie M. [и др.] // *IEEE Transactions on Visualization and Computer Graphics*. – 2016. – Т. 23, №. 8, – С. 1910–1923. – DOI: 10.1109/TVCG.2016.2569084.
8. String Art: Towards Computational Fabrication of String Images / M. Birsak, F. Rist, P. Wonka, P. Musialski // *Computer Graphics Forum*. – 2018. – Т. 37, №. 2. – С. 263-274. – DOI: 10.1111/cgf.13359.
9. Demoussel B., Larboulette C., Dattatreya R. A Greedy Algorithm for Generative String Art // *Bridges 2022 (Mathematics • Art • Music • Architecture • Culture)*. – 2022. – URL: <https://hal.science/hal-03901755/> (дата обращения: 03.02.2025).

10. Ostanin I. "String art" approach to the design and manufacturing of optimal composite materials and structures // *Composite Structures*. – 2020. – Т. 246. – С. 112396. – DOI: 10.1016/j.compstruct.2020.112396.
11. Mitra N. J., Pauly M. Shadow art // *ACM Transactions on Graphics*. – 2009. – Т. 28, № 5. – С. 156: 1-156: 7DOI: 10.1145/1618452.1618502.
12. Alexa M., Matusik W. Images from self-occlusion // *Proceedings of the International Symposium on Computational Aesthetics in Graphics, Visualization and Imaging, CAe'11*. – 2011. – С. 17-24. – DOI: 10.1145/2030441.2030445.
13. Printed Perforated Lampshades for Continuous Projective Images / H. Zhao, L. Lu, Y. Wei [et al.] // *ACM Trans. Graph.* – 2016. – Т. 35, № 5. – С. 1-11. – DOI: 10.1145/2907049.
14. Zamuda A., Mlakar U. Tiled EvoLisa image evolution with blending triangle brushstrokes and gene compression DE // *2016 IEEE Congress on Evolutionary Computation (CEC): (Vancouver, BC, Canada, 2016)*. – IEEE, 2016. – С. 2618-2625. – DOI: 10.1109/CEC.2016.7744116.
15. Animated drawings rendered by genetic programming / P. Barile, V. Ciesielski, M. Berry, K. Trist // *Proceedings of the 11th Annual conference on Genetic and evolutionary computation: ACM*. – 2009. – С. 939–946. – DOI: 10.1145/1569901.1570030.
16. den Heijer E., Eiben A. E. Evolving pop art using scalable vector graphics // *Evolutionary and Biologically Inspired Music, Sound, Art and Design*. – Berlin, Heidelberg : Springer Berlin Heidelberg, 2012. – С. 48–59. – DOI: 10.1007/978-3-642-29142-5_5.
17. Image Vectorization: a Review / M. Dziuba, I. Jarsky, V. Efimov, A. Filchenkov // *Journal of Mathematical Sciences*. – 2024. – Т. 285. – С. 155–168. – DOI: 10.1007/s10958-024-07422-4.
18. Hernando L., Mendiburu A., Lozano J. A. Hill-Climbing Algorithm: Let's Go for a Walk Before Finding the Optimum // *2018 IEEE Congress on Evolutionary Computation (CEC)*. – IEEE, 2018. – С. 1-7. – DOI: 10.1109/CEC.2018.8477836.
19. Kirkpatrick S., Gelatt C., Vecchi M. Optimization by simulated annealing // *Science*. – 1983. – Т. 220, №. 4598. – С. 671-680. DOI: 10.1126/science.220.4598.6.
20. Kennedy J., Eberhart R. C. Particle swarm optimization // *Proceedings of ICNN'95-International Conference on Neural Networks*. – IEEE, 1995. – Т. 4. – С. 1942-1948. – DOI: 10.1109/ICNN.1995.488968.