

УДК 004.032.26

АНАЛИТИЧЕСКИЙ ОБЗОР АЛГОРИТМОВ МАТИРОВАНИЯ НА ИЗОБРАЖЕНИИ



А-К.Х. Исмаил
студент гр. 850503,
кафедра ЭВМ, БГУИР



Д.Ю. Перцев
кандидат технических наук,
кафедра ЭВМ, БГУИР

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь

E-mail: mrstormproxdd@gmail.com, pertsev@bsuir.by

Д.Ю. Перцев

Окончил аспирантуру Белорусского государственного университета информатики и радиоэлектроники (2016), защитил диссертацию в совете 05.13.01 (2020). Является доцентом кафедры ЭВМ.

А-К.Х. Исмаил

Является студентом Белорусского государственного университета информатики и радиоэлектроники по специальности 1-40 02 01 “Вычислительные машины, системы и сети”.

Аннотация. Проанализированы алгоритмы матирования на изображении, их преимущества и недостатки. Представлен авторский подход, применяемый на различных вычислительных архитектурах, проведена количественная и качественная сравнение авторской версии с альтернативными.

Ключевые слова: Матирование изображения, Решение в замкнутом виде, Robust matting, Глубокие нейронные сети, Самоприспосабливающийся алгоритм на основе патчей.

Введение.

Матирование изображения I является важной задачей в области компьютерного зрения, основанной на разделении заднего плана B (от англ. background) и переднего плана F (от англ. foreground):

$$I = \alpha F + (1 - \alpha)B, \quad (1)$$

где F – изображение переднего плана;

B – изображение заднего плана;

α – карта прозрачности переднего плана, элементы в которой принимают значение в диапазоне $[0,1]$ (значение 0 – пиксель с координатой (x, y) принадлежит заднему плану, значение 1 – переднему плану).

При работе с изображениями в формате RGB имеется 7 неизвестных (для изображения переднего плана – FR, FG, FB; для заднего плана – BR, BG, BB; карта прозрачности α) и три уравнения (по одному на каждый цветовой канал), что делает эту задачу в высокой степени неопределённой. Решением указанной проблемы является формирование вспомогательной информации – например, тримапов [1] и набросков [2], представляющих собой грубую сегментацию (как правило, выполненную пользователем вручную), или заранее

подготовленное изображение заднего плана. В случае с матированием видео в качестве дополнительных данных можно использовать информацию из предыдущих кадров.

Данная работа сфокусирована на рассмотрении существующих алгоритмов матирования видео в реальном времени, подборе подходящего для применения, в первую очередь, на мобильном телефоне с платформой Android.

Обзор алгоритмов

На данный момент разработано множество алгоритмов матирования видео [3]: решение в замкнутом виде, Robust matting, алгоритм, основанный на обучении, применение глубоких нейронных сетей, самоприспосабливающийся на основе патчей.

Алгоритм решения в замкнутом виде

В работе [4] представлено решение в замкнутом виде [5] для извлечения карты прозрачности α из изображений и доказательство того, что возможно аналитически оценить F и B , получить квадратичную функцию потерь для α , которая определяется с помощью поиска точки оптимума функции потерь. Основным преимуществом алгоритма заключается в том, что для расчета значений α требуется небольшое число входных данных (например, пользовательских набросков).



Рисунок 1. Пример работы алгоритма решения в замкнутом виде: *a* – изображение павлина с набросками: синий – задний план, красный – передний план; *в* – изображение огня с набросками; *б*, *г* – результаты работы алгоритма

Robust matting

В данном подходе упор сделан на оптимизацию выборки пикселей, на основе которых выполняется отнесения неопределённого пикселя к одному из планов. В работе [6], посвящённой данному подходу утверждается, что многие алгоритмы, разработанные до этого, чрезмерно полагаются на тримапы, которые задаются пользователем. В данном алгоритме выполняется оценка вероятности принадлежности выбранных пикселей к неопределённым областям; в выборку попадают лишь те, для которых эта вероятность достаточно высока, что позволяет существенно повысить качество маркировки.ц

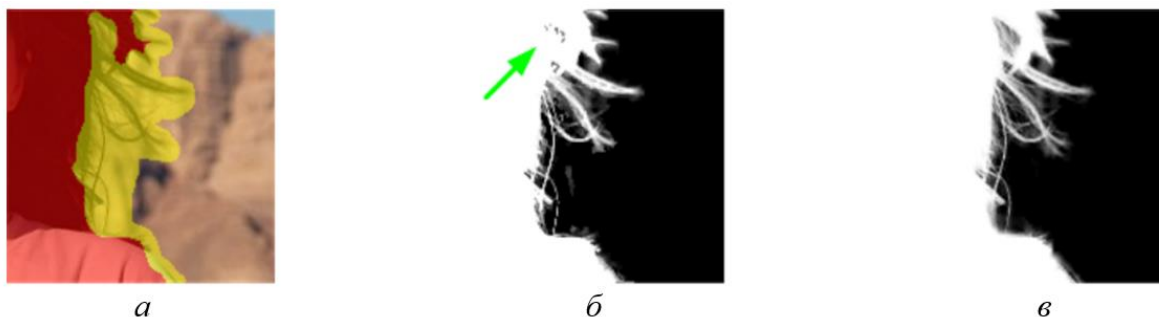


Рисунок 2. Пример работы алгоритма robust matting: *a* – исходное изображение с пользовательским вводом; *б* – результат работы алгоритма решения в замкнутом виде; *в* – результаты работы алгоритма

На рисунке 2 представлено сравнение результата работы алгоритмов robust matting и решения в замкнутом виде, описанного ранее. Визуально заметно, что robust matting выполняет более качественное матирование. На рисунке 2.б стрелкой отмечен пример артефакта, отсутствующий на рисунке 2.в.

Алгоритм, основанный на обучении

Алгоритм, основанный на обучении [2], основан на применении нейронных сетей с обучением с учителем. Авторами были предложены два подхода: глобальный и локальный. Первый подходит для использования совместно с алгоритмами, использующими тримапы в качестве дополнительной информации, в то время как последний предназначен для алгоритмов, основанных на использовании набросков. На рисунке 3 представлен результат работы локального подхода.



Рисунок 3. Пример работы алгоритма, основанного на обучении: *а* – исходное изображение с набросками: синий – передний план, красный – задний план; *б* – результат работы алгоритма

Как уже отмечалось, глобальный подход для определения принадлежности неопределённых пикселей к одному из планов использует тримапы. На рисунке 4.в зелёным цветом выделен пиксель в неопределённой области тримапа, который необходимо отнести к одному из планов, а синим и красным цветами выделены пиксели, относящиеся наверняка к переднему и заднему планам. На основе полученной информации глобальный подход выполняет маркировку пикселя.

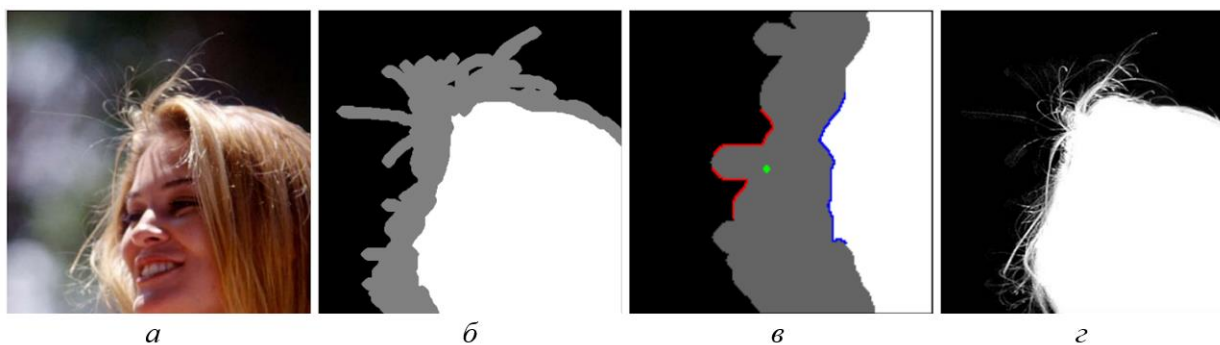


Рисунок 4. Пример работы алгоритма, основанного на обучении: *а* – исходное изображение; *б* – пользовательский тримап; *в* – пример работы алгоритма; *г* – результат работы алгоритма

Глубокие нейронные сети для матирования изображений

Впервые использование глубоких нейронных сетей в задаче матирования изображений было приведено в работе Deep Image Matting [6]. Глубокая нейронная сеть, представленная в данной работе, состоит из двух частей: первая часть – это глубокий свёрточный кодер-декодер, на вход которого поступает исходное изображение и соответствующий ему тримап, а результатом работы является оценка коэффициентов для

исходного изображения. Второй частью является небольшая свёрточная нейронная сеть, назначение которой – повышение точности и резкости результата работы первой части. Полная архитектура данной сети представлена на рисунке 5.

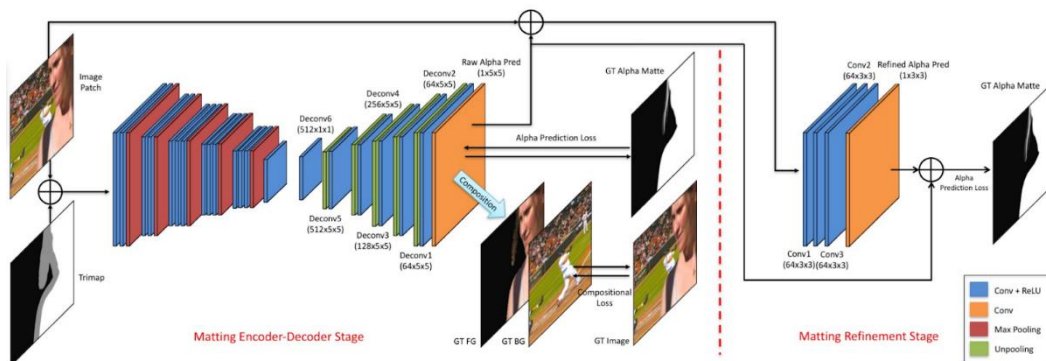


Рисунок 5. Архитектура сети

На рисунках 6 и 7 представлено сравнение данного алгоритма и его предшественников. Как видно из сравнения, данный подход обеспечивает качество, которое сильно превосходит остальные подходы к матированию изображений.

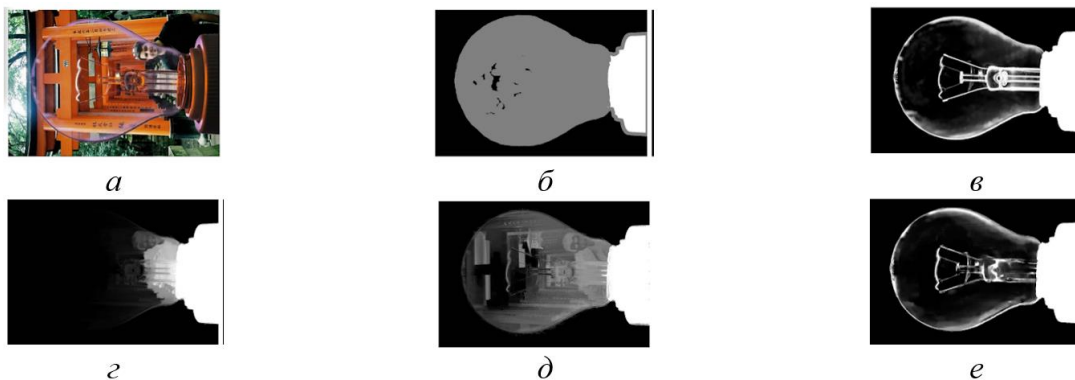


Рисунок 6. Сравнение эффективности применения алгоритмов: *a* – исходное изображение; *б* – пользовательский тримап; *в* – истинные значения; *г* – алгоритм решения в замкнутом виде; *д* – алгоритм, основанный на обучении; *е* – алгоритм с применением глубокой нейронной сети

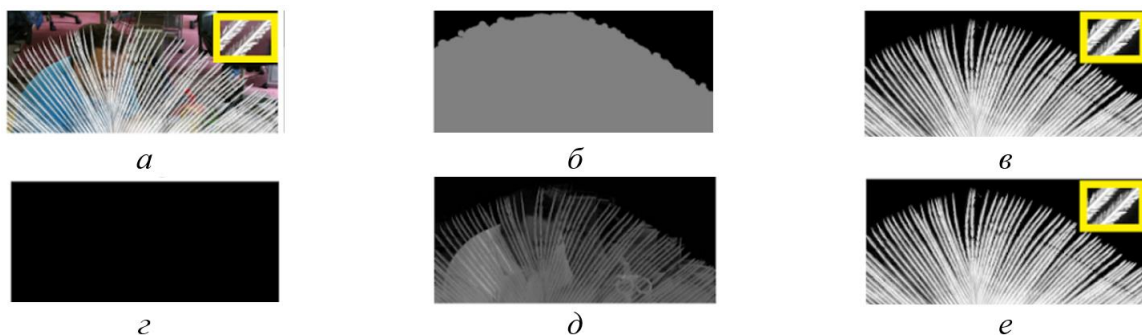


Рисунок 7. Сравнение эффективности применения алгоритмов: *a* – исходное изображение; *б* – пользовательский тримап; *в* – истинные значения; *г* – алгоритм решения в замкнутом виде; *д* – алгоритм, основанный на обучении; *е* – алгоритм с применением глубокой нейронной сети

Самоприспосабливающийся алгоритм на основе патчей

До этого момента все рассмотренные алгоритмы выполняли матирование изображения целиком, что приводит к повышению затрат памяти и снижению качества. Основываясь на подходе «Разделяй и Властвуй», самоприспосабливающийся алгоритм на основе патчей [7] решает указанные проблемы путём адаптивного деления матируемого изображения на участки небольшого размера (патчи) на основе распределения неопределённых пикселей тримапа. Затем, после параллельного анализа каждого патча, результат объединяется посредством алгоритма взвешенной оптимизации. Таким образом, требуемая память зависит не от размера исходного изображения, а только от размера патча. Чтобы избежать артефактов (рисунок 8) на границах патчей, при матировании учитываются отношения не только пикселей внутри патча, но и с некоторыми пикселями соседних патчей. Кроме того, т.к. внутри патча может быть недостаточно информации для качественного матирования, предусмотрено нахождение подобных участков, объединение их с некоторыми соседними патчами и повторное разделение (рисунок 9).

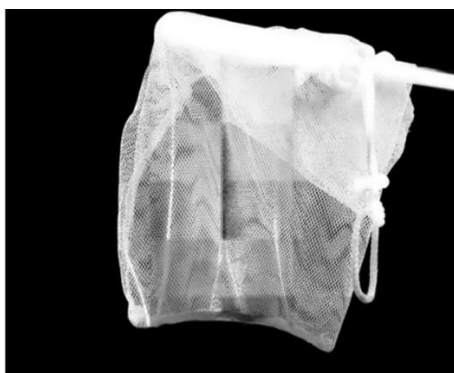


Рисунок 8. Пример артефактов

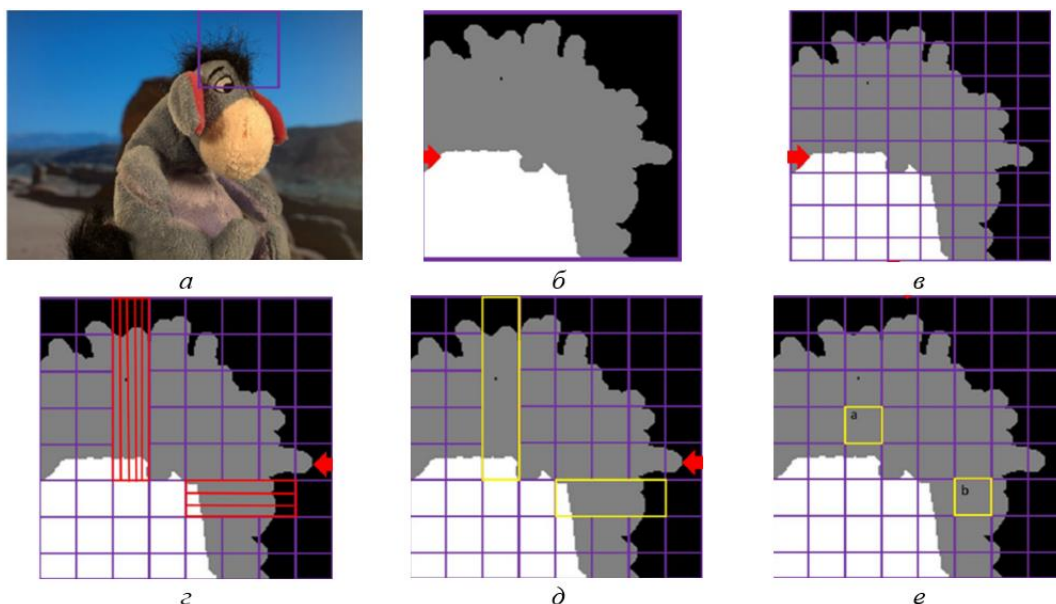


Рисунок 9. Пример работы алгоритма: *а* – исходное изображение; *б* – тримап выделенного участка; *в* – исходное разбиение на патчи; *г* – проблемные патчи; *д* – результат объединения; *е* – результат повторного разбиения

Представленная реализация алгоритма матирования

Алгоритм, представленный в данной работе, имеет несколько существенных достоинств в сравнение с описанными ранее алгоритмами. В то время, как классические алгоритмы требуют наличия заранее подготовленного фона [8], тримапа или набросков, данный алгоритм лишён этого недостатка. Это достигается за счёт рекуррентной архитектуры, и позволяет использовать его в более широком классе задач, так как отсутствует необходимость прямого пользовательского вмешательства. Таким образом, данный алгоритм наилучшим образом подходит для использования в мобильных приложениях.

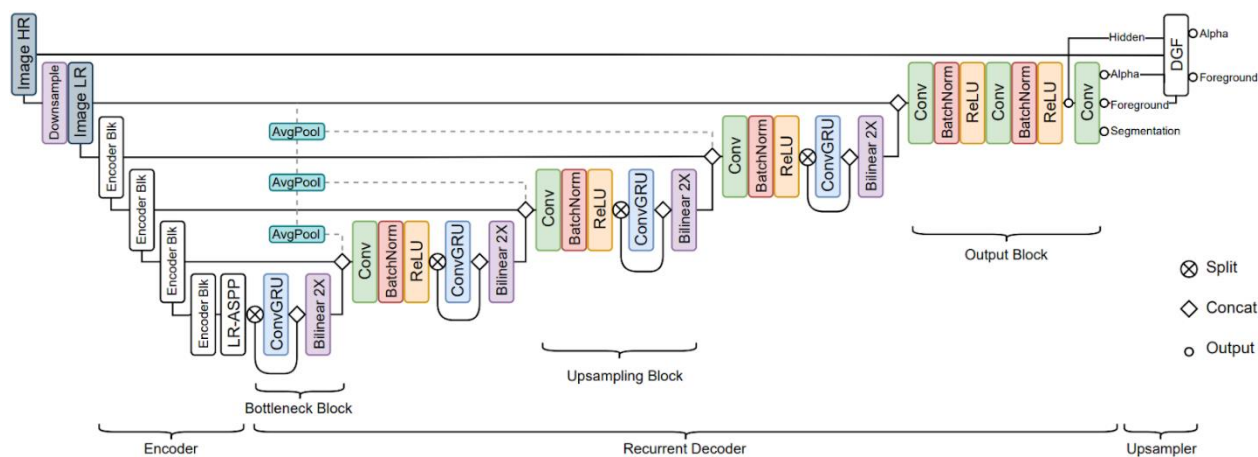


Рисунок 10. Архитектура сети

На рисунке 10 представлена архитектура нейронной сети, осуществляющая матирование изображения. Она состоит из трёх основных частей: кодер, рекуррентный декодер, выходной блок, а также блок повышения разрешения. Перед попаданием в первый блок, изображение высокого разрешения уменьшается. Данный шаг позволяет ускорить процесс обработки кадра, при этом, за счёт фильтра повышения разрешения на выходе сети, существенной потери качества на выходе не происходит. Блок кодирования осуществляет извлечение полезных признаков из изображения и уменьшает полученный кадр в 2 раза и передаёт на следующий кодер. Этот этап повторяется 4 раза. Таким образом, кодеры извлекают признаки в масштабе 12, 14, 18 и 116. Затем, рекуррентный декодер, также поэтапно, обновляет рекуррентные состояния сети и декодирование признаков, понижая количество каналов исходного изображения до одного – альфа. Выходной блок выполняет финальное извлечение признаков и генерирует коэффициенты альфа и передний план, который затем, вместе с исходным изображением в высоком разрешении, передаются фильтру для повышения разрешения результирующих изображений.

Однако, одной из серьёзных проблем реализации представленного алгоритма является то, что на данный момент не существует мобильного устройства способного выполнить такой объём вычислений за малый промежуток времени. Кроме того, существуют ограничения, затрудняющие реализацию подобных комплексных алгоритмов напрямую на устройствах на платформе Android.

Таким образом, было принято решение, что наиболее оптимальным и высокопроизводительным методом реализации такого приложения является клиент-серверная архитектура, в которой мобильное приложение является клиентом, отправляющим изображения на удалённый сервер, который, в свою очередь, обрабатывает принятый кадр и возвращает результат. Далее, задачей клиента остаётся только вывести полученное от сервера изображение. При использовании такой архитектуры приложения,

сложные вычисления выполняются на полноценных, более мощных, графических процессорах. В дополнение к этому, такой подход даёт возможность создания кроссплатформенного приложения без необходимости изменения серверной части, что сильно ускоряет процесс его разработки.

Данный подход также имеет и свои недостатки. Для работы приложению необходимо постоянное подключение к Интернету. Более того, скорость работы приложения напрямую зависит от скорости обмена информацией между клиентом и сервером.

Тестовая платформа

Параметры тестовых систем представлены в таблице 1.

Таблица 1. Параметры тестовых систем

	Смартфон	Laptop-клиент	Desktop-клиент	Сервер
CPU	HiSilicon Kirin 970	Intel Core i7-8550U	Intel Core i5-6550	4x Intel Xeon Platinum 8180M
GPU	Mali-G72 MP12	Nvidia MX250	Nvidia GTX 1060 6GB	Nvidia Tesla K80
NIC	WI-FI	Intel Dual Band Wireless-AC 8265	Realtek PCIe GbE Family Controller	?

В таблице 2 представлены результаты тестирования производительности приложения для различных тестовых систем.

Таблица 2. Временные метрики клиента

	Разрешение камеры	Кодирование изображения, мс			Отправка запроса на сервер (и ожидание ответа), мс			Декодирование ответа от сервера, мс			Средняя частота кадров, к/с
		Мин.	Ср.	Макс.	Мин.	Ср.	Макс.	Мин.	Ср.	Макс.	
Смартфон	1280x720	38	40	42	470	500	1500	63	75	92	1,08
	1920x1080	77	80	86	650	900	1700	75	105	124	0,7
	2816x1584	140	166	168	843	1100	2100	165	180	193	0,55
Laptop-клиент	1920x1080	13	17	27	490	550	820	13	30	49	1,15
Desktop-клиент	1920x1080	11	15	21	550	640	660	13	18	30	1,07

Метрики производительности серверной части приложения приведены в таблице 3.

Таблица 3. Временные метрики сервера

Разрешение изображения	Декодирование изображения от клиента, мс	Предсказание нейросети, мс	Постобработка изображения, мс
1280x720	40	5	166
1920x1080	40	6	170
2816x1584	80	6,5	250

Исходя из результатов тестирования можно судить о том, что «бутылочным горлышком» данной реализации является код, ответственный за кодирование-декодирование изображений.

Результаты

В ходе проделанной работы были рассмотрены наиболее популярные и значимые алгоритмы в области матирования изображений и, в частности, видео высокого разрешения в режиме реального времени. Посредством анализа достоинств и недостатков каждого из них, выбран алгоритм, реализованный в мобильном приложении.

При дальнейшей разработке планируется существенно снизить зависимость между скоростью работы приложения и скоростью сети посредством улучшений и оптимизации алгоритмов сжатия кадров, пересылаемых между клиентом и сервером. Кроме того, с ростом вычислительных мощностей мобильных устройств, будет реализована возможность исключения сервера из архитектуры и обработки изображений локально на устройстве, что позволит приложению работать без подключения к Интернету. Данные улучшения позволят значительно повысить производительность приложения.

Доступ к исходному коду приложения можно получить по ссылке [9].

Список использованных источников

- [1] Chuang Y. A Bayesian Approach to Digital Matting / Y.-Y. Chuang, [et al.] // Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. – 2001. – Kauai, HI, USA. – DOI: 10.1109/CVPR.2001.990970.
- [2] Zheng Y. Learning Based Digital Matting / Y. Zheng, C. Kambhamettu // 2009 IEEE 12th International Conference on Computer Vision. – 2009. – Kyoto, Japan. – DOI: 10.1109/ICCV.2009.5459326.
- [3] VideoMatting [Electronic Resource] / Lomonosov Moscow State University. – Mode of access: <https://videomattng.com>.
- [4] Levin A. A Closed Form Solution to Natural Image Matting / A. Levin, [et al.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2008. – Volume 30. – Issue 2. – P. 228-242. – DOI: 10.1109/TPAMI.2007.1177.
- [5] Closed-form expression [Electronic Resource] / Wikipedia. – Mode of access: https://en.wikipedia.org/wiki/Closed-form_expression.
- [6] Xu N. Deep Image Matting [Electronic Resource] / arXiv. – Mode of access: <https://arxiv.org/abs/1703.03872>.
- [7] Hao G. Patch-based self-adaptive matting for high-resolution image and video / G. Hao, [et al.] // The Visual Computer. – Volume 35. – 2019. – P.133-147. – DOI: 10.1007/s00371-017-1424-3.
- [8] Real-Time High-Resolution Background Matting [Electronic Resource] / University of Washington. – Mode of access: <https://grail.cs.washington.edu/projects/background-matting-v2>.
- [9] Background Matting Source Code [Electronic Resource] / GitHub. – Mode of access: <https://github.com/Sp00nyMan/BackgroundMatting>.

ANALYTICAL REVIEW OF IMAGE MATTING ALGORITHMS

A-K. ISMAIL

student of group 850503,

*Electronic Computing Machines Department,
BSUIR*

D. PERTSAU

PhD,

*Electronic Computing Machines Department,
BSUIR*

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus

E-mail: mrstormproxdd@gmail.com, pertsev@bsuir.by

Abstract. The matting algorithms on the image, their advantages and disadvantages are analyzed. The author's approach used on various computing architectures is presented, a quantitative and qualitative comparison of the author's version with alternative ones is carried out.

Keywords: Image matting, Closed-form solution, Robust matting, Deep neural networks, Self-adapting patch-based algorithm.