Алгоритм повышения качества смазанных снимков для GPU

А. А. Тихонов[1]

[1]МГУ, ВМК, каф. ММП

Всероссийская молодежная школа по параллельному программированию
«Суперкомпьютерные технологии и высокопроизводительные вычисления в образовании, науке и промышленности»
Нижний Новгород, октябрь 2010



Исследовательский проект в рамках летней сесии школы.
Задание предложено компанией NVIDIA.
Кураторы — А. В. Боресков и М. Смирнов
Научный руководитель — Д. П. Ветров





`http://school.hpc-russia.ru/`



Содержание

Проблема смазанных изображений

Современный подход

- Немного математики
  - Приближение скрытого изображения $\mathbf{L}$
  - Вычисление приближения градиента $\partial\mathbf{L} - \Psi$
  - Обновление $\mathbf{L}$
  - Нахождение ядра свертки $\mathbf{f}$
- Алгоритм для реализации
- Достоинства и недостатки алгоритма
- Почему GPU?

Обзор проекта

- Поставленные цели
- Полученный результат
- Продолжение




睇到女人定係
愛斯基摩人?



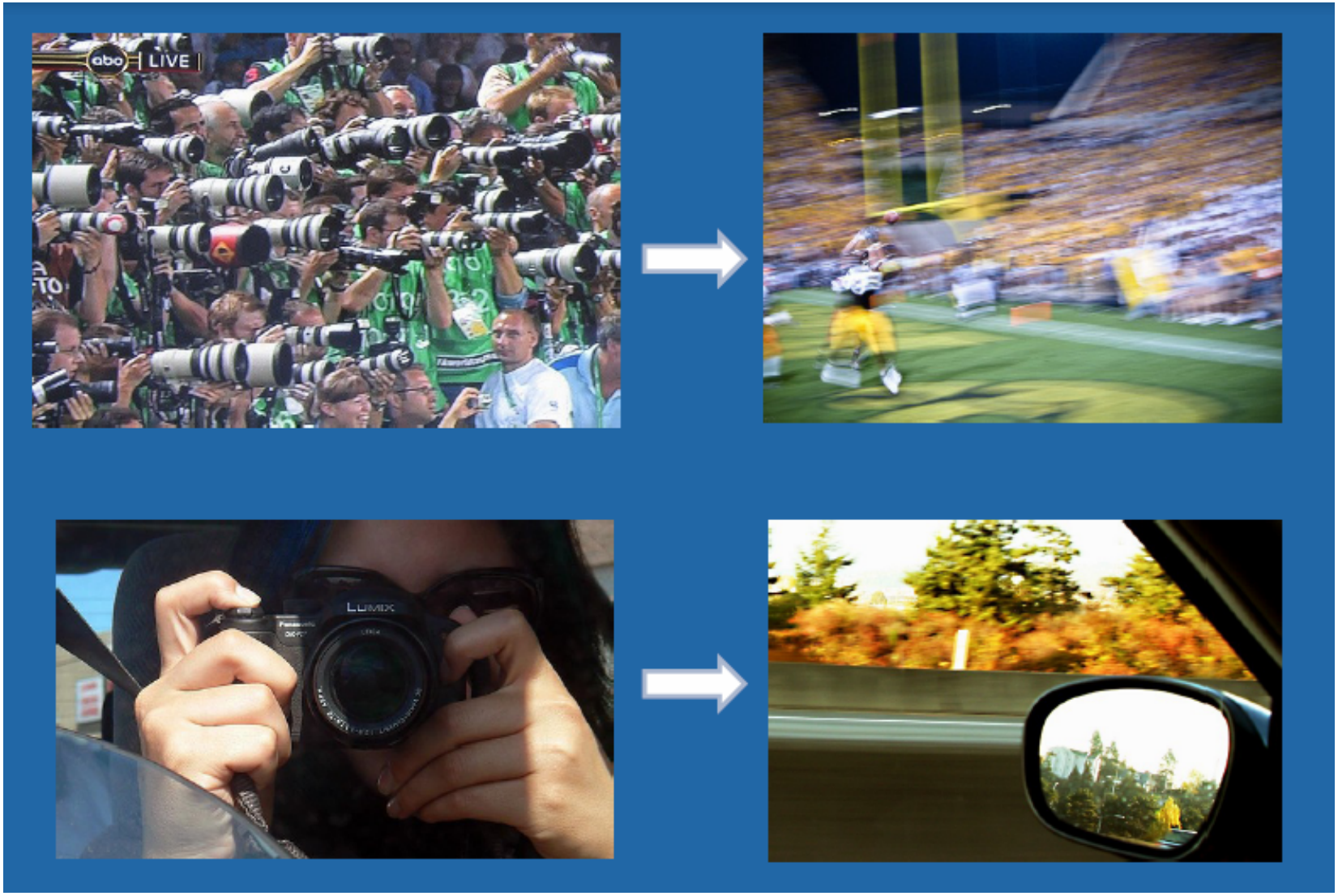


Рис.: Искажения движения (смазанность)

Причинами смазанности могут выступать различные факторы:

1. Движение камеры в процессе съемки изображения;
2. Съемка на длинной выдержке, когда сцена сама претерпевает изменения;
3. Расфокусированность оптики;
4. Использование широкоугольных объективов;
5. Атмосферная турбулентность;
6. Съемка на короткой выдержка, что не позволяет захватить достаточно фотонов;
7. Рассеянние света в конфокальной микроскопии;



Общепринятая модель размытия — свертка

$$\mathbf{I} = \mathbf{L} \otimes \mathbf{f} + \mathbf{n};$$

$\mathbf{n}$ — шум. Для искажения движения (motion blur) ядро инвариантно относительно сдвига.



Blur Model





Blurred image I

$=$



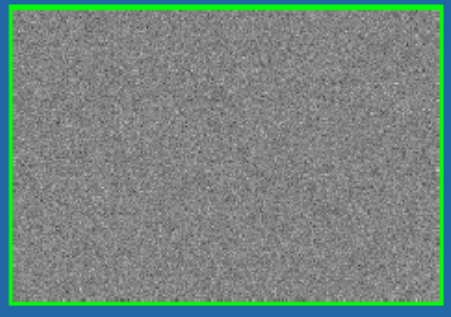
Sharp image L

$\otimes$



Blur kernel f

$+$



Camera Noise n

Shan Q., Jia J., Agarwala A. — High-quality Motion Deblurring from a Single Image.// ACM Transactions on Graphics (SIGGRAPH), 2008



Содержание

Проблема смазанных изображений

Современный подход

- Немного математики
  - Приближение скрытого изображения **L**
  - Вычисление приближения градиента ∂**L** — Ψ
  - Обновление **L**
  - Нахождение ядра свертки **f**
- Алгоритм для реализации
- Достоинства и недостатки алгоритма
- Почему GPU?

Обзор проекта

- Поставленные цели
- Полученный результат
- Продолжение



$$P(\mathbf{L}, \mathbf{f}, \mathbf{I}) = P(\mathbf{I}, \mathbf{f}|\mathbf{L})P(\mathbf{L});$$
$$P(\mathbf{L}, \mathbf{f}|\mathbf{I}) = P(\mathbf{I}|\mathbf{L}, \mathbf{f})P(\mathbf{L})P(\mathbf{f});$$



$$p(\mathbf{L}, \mathbf{f}|\mathbf{I}) \propto p(\mathbf{I}|\mathbf{L}, \mathbf{f})p(\mathbf{L})p(\mathbf{f});$$



$$p(\mathbf{I}|\mathbf{L},\mathbf{f}) = \prod_{\partial^* \in \Theta} \prod_i \mathcal{N}(\partial^* n_i | 0, \zeta_{\kappa(\partial^*)}) = \prod_{\partial^* \in \Theta} \prod_i \mathcal{N}(\partial^* I_i | \partial^* I_i^c, \zeta_{\kappa(\partial^*)});$$

Θ — множество производных ($\Theta = (\partial^0, \partial_x, \partial_y, \partial_{xx}, \partial_{xy}, \partial_{yy})$), I_i^c — i-й пиксель изображения $\mathbf{I^c} = \mathbf{L} \otimes \mathbf{f}$.



Ищем разреженное ядро:

$$p(\mathbf{f}) = \prod_j e^{-\tau f_j};$$

Здесь τ — параметр скорости [движения камеры].



Разложим правдоподобие в произведение локальной и глобальной компонент:

$$p(\mathbf{L}) = p_g(\mathbf{L})p_l(\mathbf{L});$$



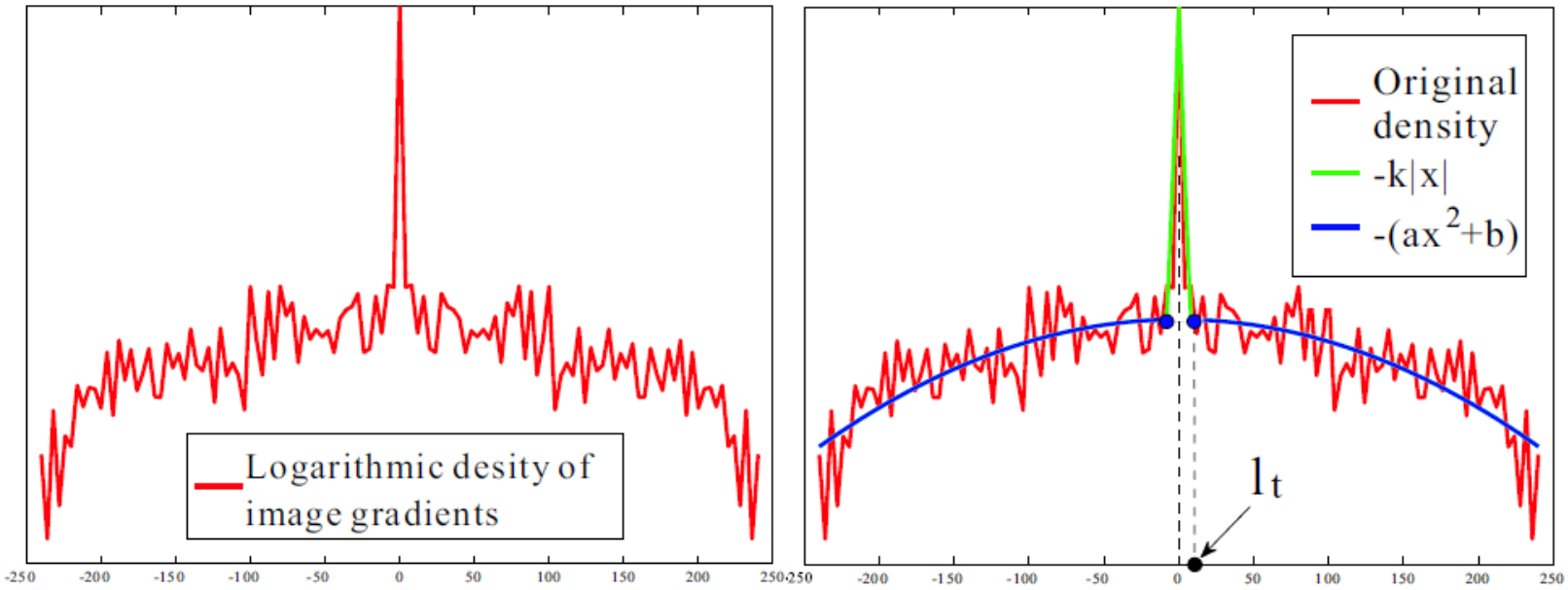


Рис.: График распределения градиентов в реальных изображениях и предлагаемая аппроксимация



$$p_I(\mathbf{L}) = \prod_{i \in \Omega} \mathcal{N}(\partial_x L_i - \partial_x I_i | 0, \sigma_1) \mathcal{N}(\partial_y L_i - \partial_y I_i | 0, \sigma_1);$$

Здесь за Ω обозначены точки изображения с локальной дисперсией менее некоторой константы.



$$E(\mathbf{L}, \mathbf{f}) = -\log p(\mathbf{L}, \mathbf{f}|\mathbf{I});$$

$$E(\mathbf{L}, \mathbf{f}) \propto \left(\sum_{\partial^* \in \Theta} w_{\kappa(\partial^*)} \|\partial^* \mathbf{L} \otimes \mathbf{f} - \partial^* \mathbf{I}\|_2^2 \right) + \lambda_1 \|\Phi(\partial_x \mathbf{L}) + \Phi(\partial_y \mathbf{L})\|_1 +$$
$$+ \lambda_2 \left(\|\partial_x \mathbf{L} - \partial_x \mathbf{I}\|_2^2 \circ \mathbf{M} + \|\partial_y \mathbf{L} - \partial_y \mathbf{I}\|_2^2 \circ \mathbf{M} \right) + \|\mathbf{f}\|_1; \quad (1)$$

Прямая оптимизация этого функционала сложна, сходимость медленная. Авторы [1] предлагают свой вариант, основанный на пошаговой оптимизации $\mathbf{L}$, а затем $\mathbf{f}$.



Зафиксируем некоторое приближение ядра $\mathbf{f}$:

$$E_{\mathbf{L}}(\mathbf{L},\mathbf{f}) = \left(\sum_{\partial^* \in \Theta} w_{\kappa(\partial^*)} \|\partial^* \mathbf{L} \otimes \mathbf{f} - \partial^* \mathbf{I}\|_2^2\right) + \lambda_1 \|\Phi(\partial_x \mathbf{L}) + \Phi(\partial_y \mathbf{L})\|_1 +$$
$$+\lambda_2 \left(\|\partial_x \mathbf{L} - \partial_x \mathbf{I}\|_2^2 \circ \mathbf{M} + \|\partial_y \mathbf{L} - \partial_y \mathbf{I}\|_2^2 \circ \mathbf{M}\right);$$

Это сильно невыпуклая функция от многих тысяч переменных,
Прямая оптимизаця очень трудоемка.



Введем дополнительные переменные $\Psi = (\Psi_x, \Psi_y)$, которые будут аппроксимировать $\partial\mathbf{L}$, таким образом, каждой паре градиентов пикселя i — $(\partial_x L_i, \partial_y L_i) \in \partial\mathbf{L}$ — будет соответствовать пара $(\psi_{i,x}, \psi_{i,y}) \in \Psi$.
При фиксированном $\mathbf{L}$ можно найти такие Ψ, которые будут минимизировать энергию. То есть, вместо подбороа всего изображения, подберем лишь его градиент (влияет лишь на часть выражения), а затем по найденным градиентам обновим $\mathbf{L}$. То есть, перешли к минимизации:

$$
\begin{aligned}
E_{\mathbf{L}}(\mathbf{L}', \mathbf{f}) = &\left(\sum_{\partial^* \in \Omega} w_{\kappa(\partial^*)} \|\partial^* \mathbf{L} \otimes \mathbf{f} - \partial^* \mathbf{I}\|_2^2 \right) + \lambda_1 \|\Phi(\Psi_x) + \Phi(\Psi_y)\|_1 + \\
&+ \lambda_2 \left(\|\Psi_x - \partial_x \mathbf{I}\|_2^2 \circ \mathbf{M} + \|\Psi_y - \partial_y \mathbf{I}\|_2^2 \circ \mathbf{M} \right) + \\
&+ \gamma \left(\|\Psi_x - \partial_x \mathbf{L}\|_2^2 + \|\Psi_y - \partial_y \mathbf{L}\|_2^2 \right);
\end{aligned}
$$



С увеличением номера итерации параметр регуляризации по отклонению градиентов γ также будет расти, и в конечном итоге Ψ будет практически совпадать с $\partial\mathbf{L}$, что означает, что минимизация $E'_{\mathbf{L}}$ через несколько шагов процесса будет эквивалентна минимизации $E_{\mathbf{L}}$.
Распишем оба шага подробнее.



При фиксированном $\mathbf{L}$, а значит, и $\partial\mathbf{L}$, остается оптимизировать E_Ψ:

$$\begin{aligned} E_\Psi(\mathbf{L}, \mathbf{f}) =& \lambda_1 \|\Phi(\Psi_x) + \Phi(\Psi_y)\|_1 + \lambda_2 \|\Psi_x - \partial_x \mathbf{I}\|_2^2 \circ \mathbf{M} + \\ & \lambda_2 \|\Psi_y - \partial_y \mathbf{I}\|_2^2 \circ \mathbf{M} + \gamma \|\Psi_x - \partial_x \mathbf{L}\|_2^2 + \gamma \|\Psi_y - \partial_y \mathbf{L}\|_2^2; \end{aligned} \tag{2}$$

Можно произвести дальнейшее разделение по пикселям и координатам x, y, получим, что для $\nu \in \{x, y\}$ нужно минимизировать:

$$E_{\psi_{i,\nu}} = \lambda_1 |\Phi(\psi_{i,\nu})| + \lambda_2 m_i (\Psi_{i,\nu} - \partial_\nu I_i)^2 + \gamma (\Psi_{i,\nu} - \partial_\nu L_i)^2;$$



Каждая компонента $E_{\psi_{i,\nu}}$ содержит лишь одну переменную $\psi_{i,\nu}$, поэтому они легко могут быть оптимизированы, а результаты собраны вместе.
Ф состоит из 4-х выпуклых частей, каждая может быть оптимизирована отдельно, а затем лучший результат взят для определения глобально минимального Ψ.



При фиксированном Ψ, будем обновлять $\mathbf{L}$, минимизируя по нему следующее выражение:

$$E'_{\mathbf{L}} = \left(\sum_{\partial^* \in \Omega} w_{\kappa(\partial^*)} \|\partial^* \mathbf{L} \otimes \mathbf{f} - \partial^* \mathbf{I}\|_2^2 \right) + \\ + \gamma \|\Psi_x - \partial_x \mathbf{L}\|_2^2 + \gamma \|\Psi_y - \partial_y \mathbf{L}\|_2^2;$$

Так как здесь присутствует свертка, выгодно перейти к работе с образами Фурье. Если обозначить оператор преобразования Фурье за $\mathcal{F}$, а за $\mathcal{F}^{-1}$ — ему обратный, то получим:

$$E'_{\mathcal{F}(\mathbf{L})} = \left(\sum_{\partial^* \in \Omega} w_{\kappa(\partial^*)} \|\mathcal{F}(\mathbf{L}) \circ \mathcal{F}(\mathbf{f}) \circ \mathcal{F}(\partial^*) - \mathbf{F}(\mathbf{I}) \circ \mathcal{F}(\partial^*)\|_2^2 \right) + \\ + \gamma \|\mathcal{F}(\Psi_x) - \mathcal{F}(\mathbf{L}) \circ \mathcal{F}(\partial_x)\|_2^2 + \gamma \|\mathcal{F}(\Psi_y) - \mathcal{F}(\mathbf{L}) \circ \mathcal{F}(\partial_y)|_2^2;$$



Ссылаясь на формулу Планшераля, авторы [1] переходят от минимизации $E'_{\mathbf{L}}$ к эквивалентной задаче минимизации $E'_{\mathcal{F}(\mathbf{L})}$. Таким образом, искомое значение затем находится с помощью обратного преобразования Фурье:

$$\mathbf{L}^* = \mathcal{F}^{-1}(\arg\min_{\mathcal{F}(\mathbf{L})} E_{\mathcal{F}(\mathbf{L})})';$$



Теперь минимальное значение такой энергии может быть получено, если приравнять к нулю его производную $\partial E'_{\mathcal{F}(\mathbf{L})}/\partial \mathcal{F}_{\mathbf{L}}$, это можно сделать в явном виде, откуда:

$$\mathbf{L}^* = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(\mathbf{f})} \circ \mathcal{F}(\mathbf{I}) \circ \Delta + \gamma\overline{\mathcal{F}(\partial_x)} \circ \mathcal{F}(\Psi_x) + \gamma\overline{\mathcal{F}(\partial_y)} \circ \mathcal{F}(\Psi_y)}{\overline{\mathcal{F}(\mathbf{f})} \circ \mathcal{F}(\mathbf{f}) \circ \Delta + \gamma\overline{\mathcal{F}(\partial_x)} \circ \mathcal{F}(\partial_x) + \gamma\overline{\mathcal{F}(\partial_y)} \circ \mathcal{F}(\partial_y)}\right); \tag{3}$$

Тут $\Delta = \sum_{\partial^* \in \Theta} w_{\kappa(\partial^*)}\overline{\mathcal{F}(\partial^*)} \circ \mathcal{F}(\partial^*)$, $\overline{(\cdot)}$ обозначает оператор комплексного сопряжения, и деление производится поэлементно.



Будем оптимизировать энергию при фиксированном $\mathbf{L}$:

$$E(\mathbf{f}) = \left(\sum_{\partial^* \in \Omega} w_{\kappa(\partial^*)} \|\partial^* \mathbf{L} \otimes \mathbf{f} - \partial^* \mathbf{I}\|_2^2 \right) + \|\mathbf{f}\|_1; \qquad (4)$$

Авторы алгоритма ссылаются на работу [?], в которой функционал такого вида успешно оптимизируется с помощью специального interior point method.





Содержание

Проблема смазанных изображений

Современный подход

Немного математики

Приближение скрытого изображения **L**

Вычисление приближения градиента ∂**L** — Ψ

Обновление **L**

Нахождение ядра свертки **f**

Алгоритм для реализации

Достоинства и недостатки алгоритма

Почему GPU?

Обзор проекта

Поставленные цели

Полученный результат

Продолжение



Схема полученного алгоритма:

Вход:

$\mathbf{I}$ — размытое изображение, $\mathbf{f}$ — начальное приближение ядра;

Выход:

$\mathbf{L}$ — искомое четкое изображение, $\mathbf{f}$ — исходное ядро размытия;

$\mathbf{L} \Leftarrow \mathbf{I}$; // *инициализация скрытого изображения наблюдаемым*;

оптимизация $\mathbf{L}$ и $\mathbf{f}$:

повторять

оптимизация $\mathbf{L}$:

повторять

Обновить Ψ, минимизируя (2);

Вычислить $\mathbf{L}$ согласно (3);

пока $\|\Delta\mathbf{L}\|_2 < 1 \times 10^{-5}$ и $\|\Delta\Psi\|_2 < 1 \times 10^{-5}$;

Обновить $\mathbf{f}$, минимизируя (4);

пока $\|\Delta\mathbf{f}\|_2 < 1 \times 10^{-5}$ или максимальное число итераций завершено;



Итого два итерационных процесса — внутренний (вложенный), чередование вычисления Ψ и $\mathbf{L}$, и внешний, вычисление очередного приближения скрытой картинки $\mathbf{L}$ и на его основе уточнение ядра $\mathbf{f}$.



Содержание

Проблема смазанных изображений

Современный подход

Немного математики

Приближение скрытого изображения **L**

Вычисление приближения градиента ∂**L** — Ψ

Обновление **L**

Нахождение ядра свертки **f**

Алгоритм для реализации

Достоинства и недостатки алгоритма

Почему GPU?

Обзор проекта

Поставленные цели

Полученный результат

Продолжение



Недостатки метода

1. Вычислительная сложность.
2. Требовательность к памяти.
3. Параметры (около 7-8) — эвристика
4. Много лишних вычислений.
5. Нет четкого критерия останова (?).



Достоинства

1. Лучшее качество (на сегодня).
2. Получает ядро в явном виде.
3. Обоснованность (с точностью до вероятностной модели).
4. Есть идеи для улучшения алгоритма.
5. Можно распараллелить.
6. Можно выдавать результат постепенно, останов по требованию.



Содержание

Проблема смазанных изображений

Современный подход

- Немного математики
  - Приближение скрытого изображения **L**
  - Вычисление приближения градиента ∂**L** — Ψ
  - Обновление **L**
  - Нахождение ядра свертки **f**
- Алгоритм для реализации
- Достоинства и недостатки алгоритма
- Почему GPU?

Обзор проекта

- Поставленные цели
- Полученный результат
- Продолжение



Почему GPU?

1. Необходимо ускорить.
2. Отлично подходит под архитектуру GPU.
3. Под силу рабочим станциям/серверам/компьютерам специалистов, задача повседневная, не для суперкомпьютера.
4. Много данных, которые трудно разделить, на GPU — в общей памяти.
5. Уже есть эффективная реализация на BSGP (Bulk-Synchronous GPU Programming).
6. Улучшение аппаратуры — рост качества/скорости/разрешения картинки без смены алгоритма (почти).



Содержание

Проблема смазанных изображений

Современный подход

- Немного математики
  - Приближение скрытого изображения **L**
  - Вычисление приближения градиента ∂**L** — Ψ
  - Обновление **L**
  - Нахождение ядра свертки **f**
- Алгоритм для реализации
- Достоинства и недостатки алгоритма
- Почему GPU?

Обзор проекта

- Поставленные цели
- Полученный результат
- Продолжение



Поставленные цели

- Разработка тестовой реализации алгоритма на основе доступных материалов, проверка качества результатов
- Исследование возможности адаптивного подбора параметров
- Портирование алгоритма на GPU с использованием технологии CUDA
- Портирование алгоритма на несколько GPU с использованием наиболее подходящего API



Содержание

Проблема смазанных изображений

Современный подход

- Немного математики
  - Приближение скрытого изображения **L**
  - Вычисление приближения градиента ∂**L** — Ψ
  - Обновление **L**
  - Нахождение ядра свертки **f**
- Алгоритм для реализации
- Достоинства и недостатки алгоритма
- Почему GPU?

Обзор проекта

- Поставленные цели
- Полученный результат
- Продолжение



Результат на сегодня

- Реализован основной алгоритм в среде MATLAB
- Ведутся эксперименты по его модификации
- Практически решен вопрос выбора библиотек (CUFFT, OpenCV)
- Есть анализ производительности, выявлены узкие места



где $r = \sqrt{a^2 + b^2}$ и $\varphi = \operatorname{arctg} \frac{b}{a}$. Применяя теорему о вычетах, получаем:

$$\int_{I} + \int_{C_R} + \int_{II} + \int_{c_r} = \frac{\pi}{b} 4\varphi\,(\pi - i \ln r).$$

В соответствии со сказанным выше имеем:

$$\int_{I} + \int_{II} = -\int_{r}^{R} \frac{4\pi i \ln x - 4\pi^2}{(x+a)^2 + b^2}\, dx.$$

Так же как в предыдущем примере, докажем, что $\lim\limits_{r \to 0} \int\limits_{c_r} = \lim\limits_{R \to \infty} \int\limits_{C_R} = 0$,

и тогда в пределе при $r \to 0$, $R \to \infty$ будем иметь:

$$4\pi^2 \int_0^{\infty} \frac{dx}{(x+a)^2 + b^2} - 4\pi i \int_0^{\infty} \frac{\ln x\, dx}{(x+a)^2 + b^2} = \frac{4\pi\varphi}{b}\,(\pi - i \ln r).$$

Отсюда, сравнивая мнимые части, получим искомый интеграл

$$\int_0^{\infty} \frac{\ln x\, dx}{(x+a)^2 + b^2} = \frac{\varphi \ln r}{b} = \frac{\ln \sqrt{a^2 + b^2}}{b} \operatorname{arctg} \frac{b}{a}. \qquad (2)$$

Пример 3. С помощью того же контура вычисляется интеграл (Эйлер)

$$\int_0^{\infty} \frac{x^{a-1}}{1+x}\, dx \qquad (0 < a < 1),$$

если в качестве вспомогательной функции взять

$$f(z) = \frac{z^{a-1}}{1+z} = \frac{e^{(a-1)\ln z}}{1+z}$$

где $z = \sqrt{R^2 + 1}\, e^{i\varphi}$ и $\varphi = \operatorname{arctg} \frac{1}{R}$. Переходя теперь к пределу, получаем:

$$\int_{-R}^{R} + \int_{C_R} + \int_{\gamma_\varepsilon} + \int_{\gamma_1} = \frac{\pi i}{2} \operatorname{tg}(\pi z - i \ln r).$$

В соответствии со сказанным выше имеем:

$$\int_{-R}^{R} + \int_{C_R} = -\int_{\gamma_1} \frac{\operatorname{tg}(\pi z - i \ln r)}{(z+R)(z+R^2+R)}\, dz.$$

Так же как в предыдущем примере, можно показать, что $\lim\limits_{R \to \infty} \int_{C_R} = \lim\limits_{\varepsilon \to 0} \int_{\gamma_\varepsilon} = 0$, и после перехода при $\varepsilon \to 0$, $R \to \infty$ будет иметь:

$$\operatorname{tg} \int_{-\infty}^{\infty} \frac{dx}{(x+R)(x+R^2+R)} - \operatorname{tg} \int_{-\infty}^{\infty} \frac{\ln x\, dx}{(x+R)(x+R^2+R)} = \frac{\pi \operatorname{tg} r}{2} (\pi z - i \ln r).$$

Отсюда, сравнивая мнимые части, получим искомый интеграл

$$\int_{-\infty}^{\infty} \frac{\ln x\, dx}{(x+R)(x+R^2+R)} = \frac{\pi \ln r}{2} = \frac{\ln \sqrt{R^2+R}}{2} \operatorname{arctg} \frac{1}{R}. \qquad (9)$$

Пример 2. С помощью того же метода вычисляется интеграл (6.6.39)

$$\int_0^{\infty} \frac{x^{\alpha-1}}{1+x}\, dx \quad (0 < \alpha < 1),$$

если в качестве вспомогательной функции взять

$$w(z) = \frac{z^{\alpha-1}}{1+z} = \frac{e^{(\alpha-1)\ln z}}{1+z}$$

где $r=\sqrt{a^2+b^2}$ и $\varphi=\operatorname{arctg}\frac{b}{a}$. Применяя теорему о вычетах, получаем:

$$\int_{I}+\int_{C_R}+\int_{II}+\int_{c_r}=\frac{\pi}{b}\,4\varphi\,(\pi-i\ln r).$$

В соответствии со сказанным выше имеем:

$$\int_{I}+\int_{II}=-\int_{r}^{R}\frac{4\pi i\ln x-4\pi^2}{(x+a)^2+b^2}\,dx.$$

Так же как в предыдущем примере, докажем, что $\lim\limits_{r\to 0}\int\limits_{c_r}=\lim\limits_{R\to\infty}\int\limits_{C_R}=0$, и тогда в пределе при $r\to 0$, $R\to\infty$ будем иметь:

$$4\pi^2\int_0^{\infty}\frac{dx}{(x+a)^2+b^2}-4\pi i\int_0^{\infty}\frac{\ln x\,dx}{(x+a)^2+b^2}=\frac{4\pi\varphi}{b}\,(\pi-i\ln r).$$

Отсюда, сравнивая мнимые части, получим искомый интеграл

$$\int_0^{\infty}\frac{\ln x\,dx}{(x+a)^2+b^2}=\frac{\varphi\ln r}{b}=\frac{\ln\sqrt{a^2+b^2}}{b}\operatorname{arctg}\frac{b}{a}. \qquad (2)$$

Пример 3. С помощью того же контура вычисляется интеграл (Эйлер)

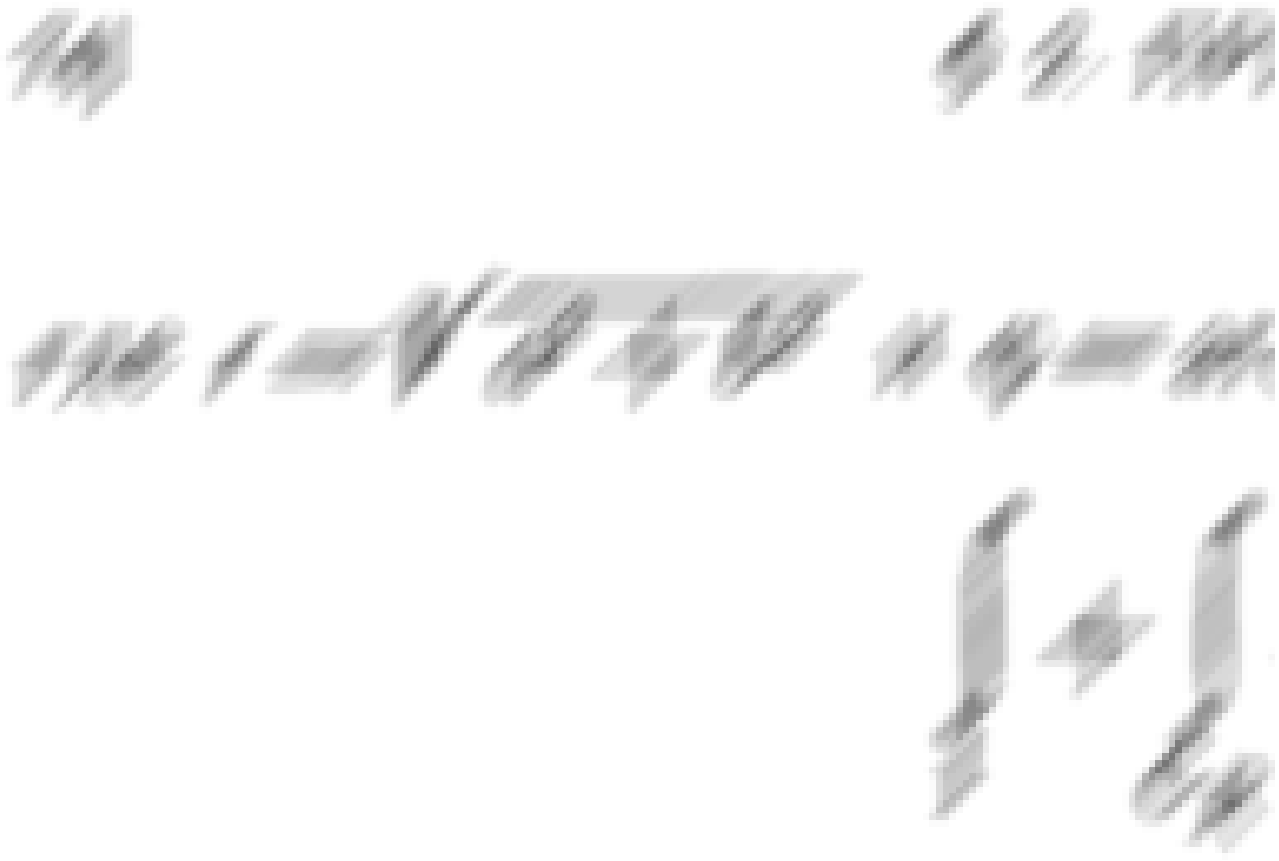
$$\int_0^{\infty}\frac{x^{a-1}}{1+x}\,dx \qquad (0<a<1),$$

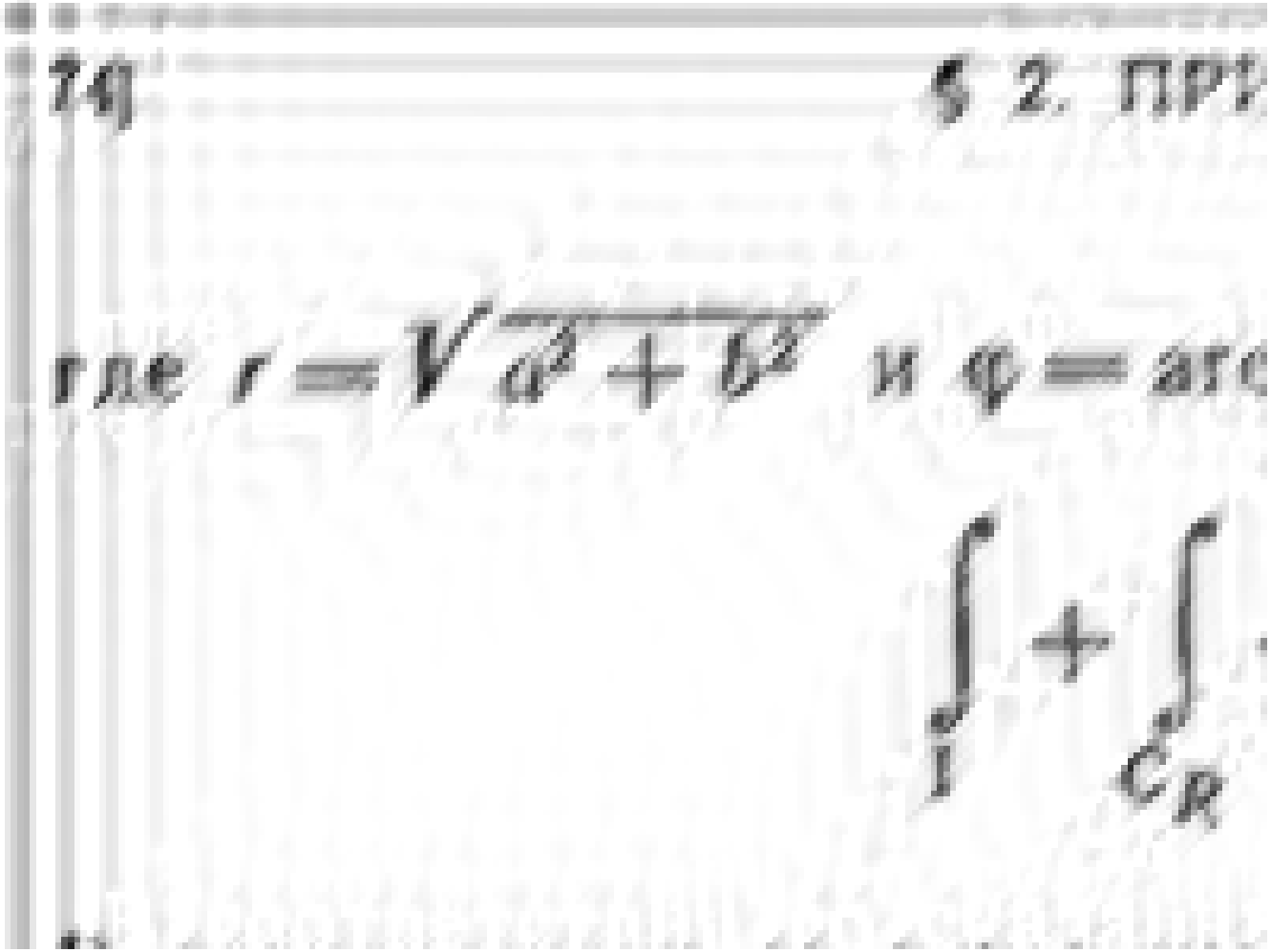
если в качестве вспомогательной функции взять

$$f(z)=\frac{z^{a-1}}{1+z}=\frac{e^{(a-1)\ln z}}{1+z}$$

где $r = \sqrt{a^2 + b^2}$ и $\varphi = \arg$

$$\int_I + \int_{C_R}$$




74
§ 2. ПР
где $r = \sqrt{a^2 + b^2}$ и $\varphi =$ ar
$\int + \int$
C_R

Сравним с результатом авторов метода?



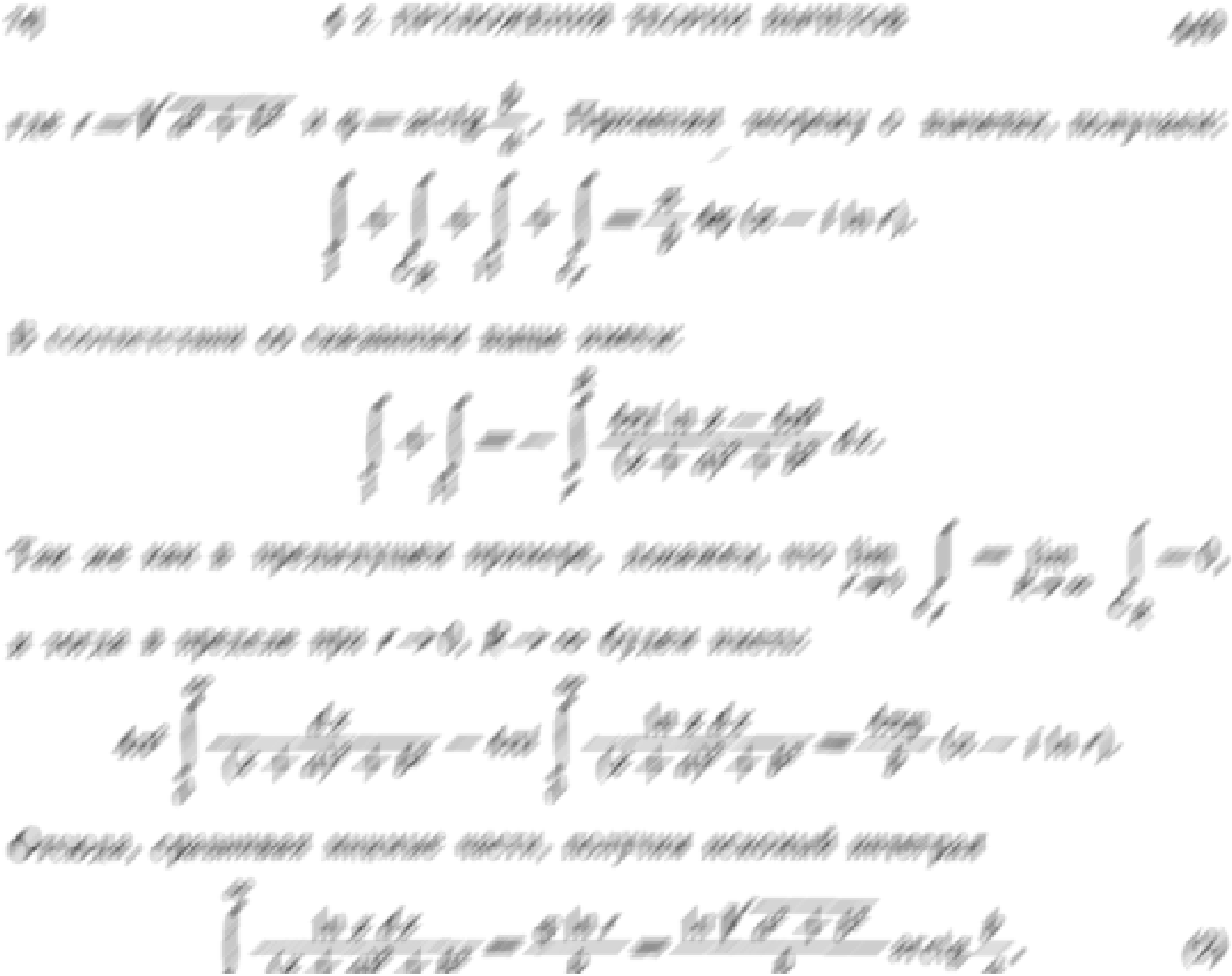


Рис.: Половина «теории вычетов»

где $r = \sqrt{a^2 + b^2}$ и $\varphi = \operatorname{arctg} \frac{b}{a}$. Применяя теорему о вычетах, получаем:

$$\int_{\text{I}} + \int_{C_R} + \int_{\text{II}} + \int_{c_r} = \frac{\pi}{b} 4\varphi (\pi - i \ln r).$$

В соответствии со сказанным выше имеем:

$$\int_{\text{I}} + \int_{\text{II}} = - \int_{r}^{R} \frac{4\pi i \ln x - 4\pi^2}{(x+a)^2 + b^2} dx.$$

Так же как в предыдущем примере, докажем, что $\lim\limits_{r \to 0} \int\limits_{c_r} = \lim\limits_{R \to \infty} \int\limits_{C_R} = 0$,

и тогда в пределе при $r \to 0$, $R \to \infty$ будем иметь:

$$4\pi^2 \int_0^{\infty} \frac{dx}{(x+a)^2 + b^2} - 4\pi i \int_0^{\infty} \frac{\ln x \, dx}{(x+a)^2 + b^2} = \frac{4\pi\varphi}{b} (\pi - i \ln r).$$

Отсюда, сравнивая мнимые части, получим искомый интеграл

$$\int_0^{\infty} \frac{\ln x \, dx}{(x+a)^2 + b^2} = \frac{\varphi \ln r}{b} = \frac{\ln \sqrt{a^2 + b^2}}{b} \operatorname{arctg} \frac{b}{a}. \qquad (2)$$

Рис.: И никакого фотошопа





Рис.: Пикассо (800x532)





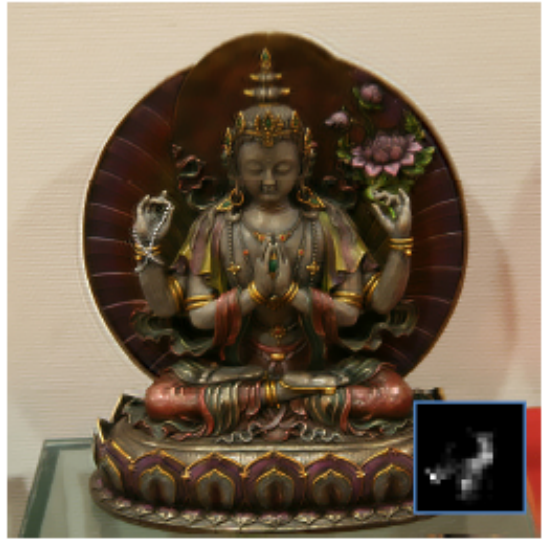
Рис.: Красиво, но как хватает памяти? :)



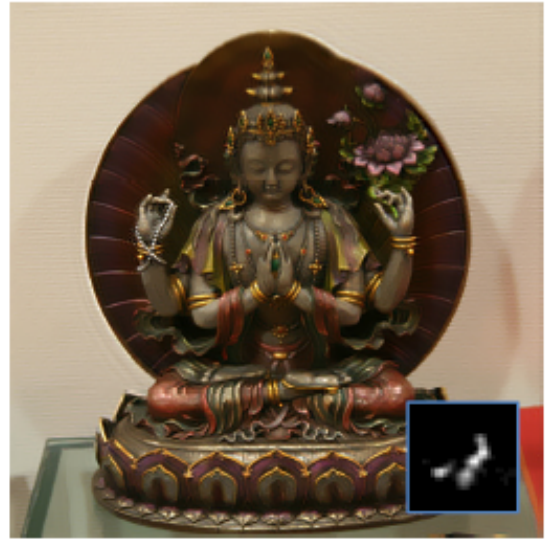


(a) blurry input

(b) [Yuan et al. 2007]



(c) [Shan et al. 2008]



(d) our method

Figure 11: *Comparison with previous deblurring methods.*

Рис.: Из Sunghyun Cho, Seungyong Lee — Fast Motion Deblurring, proc. SIGGRAPH ASIA, 2009



Image	Size		Processing time (sec.)		
	Image	Kernel	A	B	C
a	972×966	65×93	2.188	3.546	5.766
b	1024×768	49×47	1.062	1.047	2.125
c	483×791	35×39	0.343	0.235	0.578
d	858×558	61×43	0.406	0.297	0.703
e	846×802	35×49	0.516	0.406	0.922

Table 1: *Processing times of the deblurring examples in Fig. 10. A: kernel estimation. B: final deconvolution. C: total processing time.*

Image	Size		Processing time (sec.)		
	Image	Kernel	A	B	C
Picasso	800×532	27×19	360	20	0.609
statue	903×910	25×25	762	33	0.984
night	836×804	27×21	762	28	0.937
red tree	454×588	27×27	309	11	0.438

Table 2: *Processing time comparison. A: [Shan et al. 2008]. B: our method with C++ implementation. C: our method with GPU acceleration.*

Рис.: Из Sunghyun Cho, Seungyong Lee — Fast Motion Deblurring, proc. SIGGRAPH ASIA, 2009



Содержание

Проблема смазанных изображений

Современный подход

- Немного математики
  - Приближение скрытого изображения **L**
  - Вычисление приближения градиента ∂**L** — Ψ
  - Обновление **L**
  - Нахождение ядра свертки **f**
- Алгоритм для реализации
- Достоинства и недостатки алгоритма
- Почему GPU?

Обзор проекта

- Поставленные цели
- Полученный результат
- Продолжение



Литература

- Shan Q., Jia J., Agarwala A. — High-quality Motion Deblurring from a Single Image.// ACM Transactions on Graphics (SIGGRAPH), 2008.
- Sunghyun Cho, Seungyong Lee — Fast Motion Deblurring // Proceedings of SIGGRAPH ASIA, 2009.
- Tony F. C., Shen J. — Theory and Computation of Variational Image Deblurring, 2005.
- Гонсалес Р., Вудс Р. — Цифровая обработка изображений.// М.: Техносфера, 2005.
- Tony F. C., Shen J. — Theory and Computation of Variational Image Deblurring, 2005.



Спасибо за внимание

