



# Modern image inpainting methods

---

Юрий Гитман

*Video Group*

*CS MSU Graphics & Media Lab*



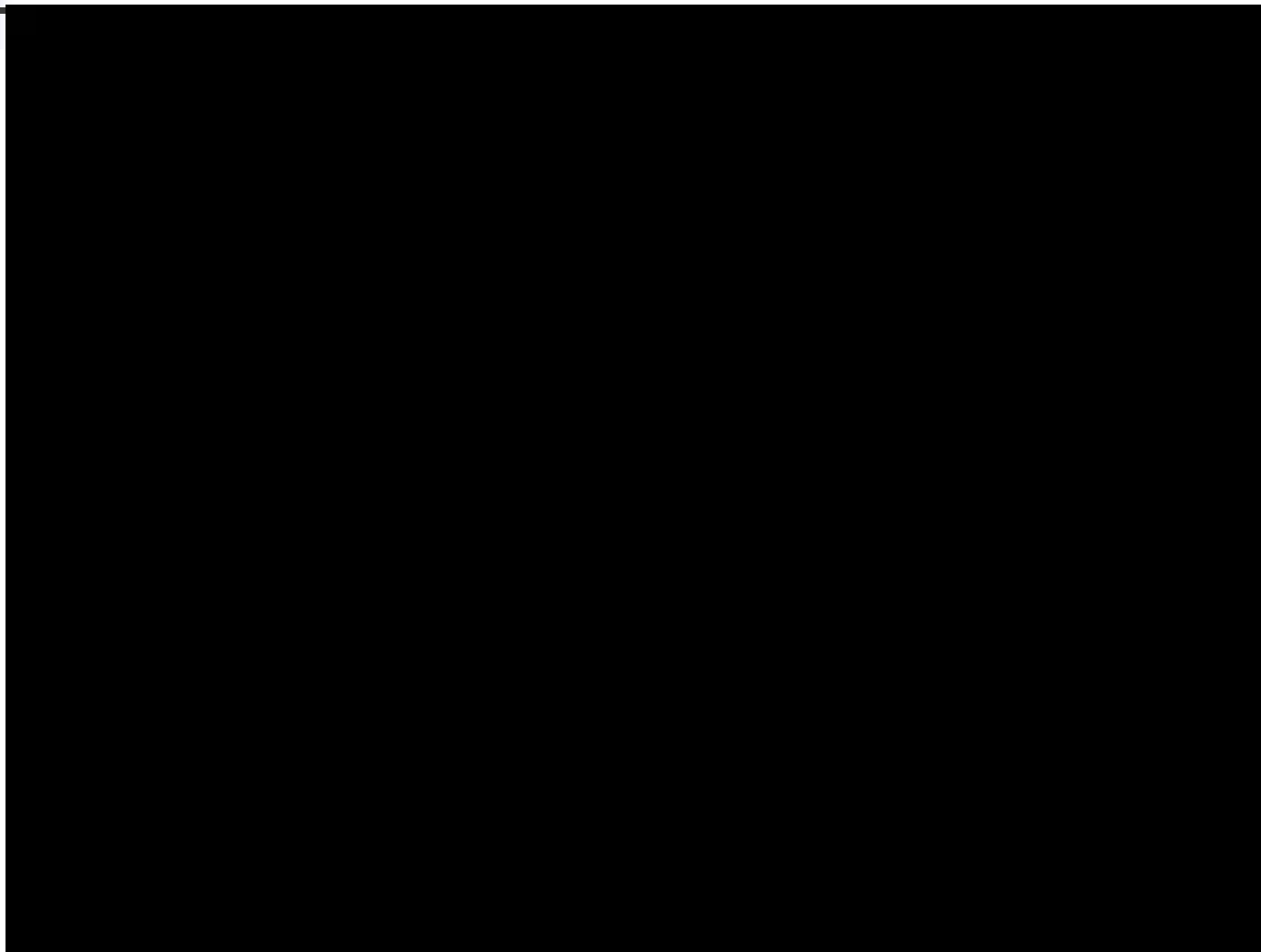
# Содержание

---

- **Введение**
- Inpainting with third-order optimal PDE
- Методы композиции блоков
- Оптимизации
- Свой алгоритм
- Результаты тестирования



# Photoshop CS6 tool



# Примеры (не photoshop!)

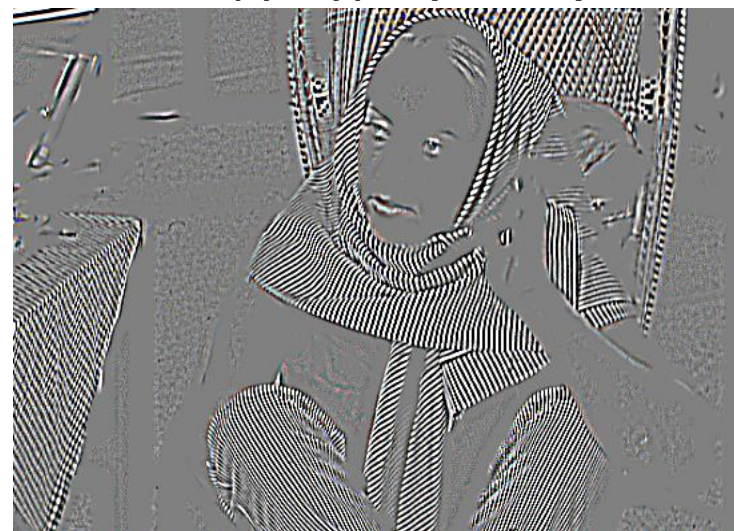
## Синтез и восстановление текстур



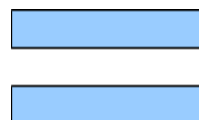
# Изображение



Структура (sketch)



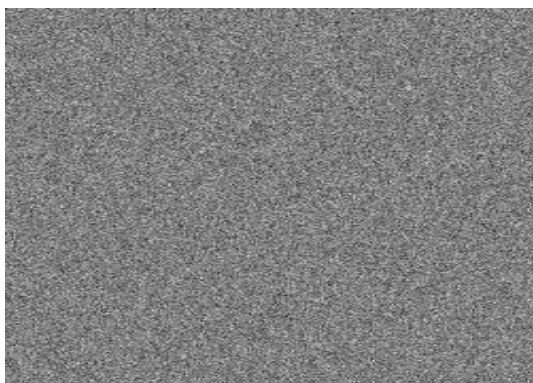
Текстура (texture)



Исходное изображение

# Параметры текстур (1)

## Размер текстурного элемента (текстона)



Шум – нет текстонов



Овца – это текстон



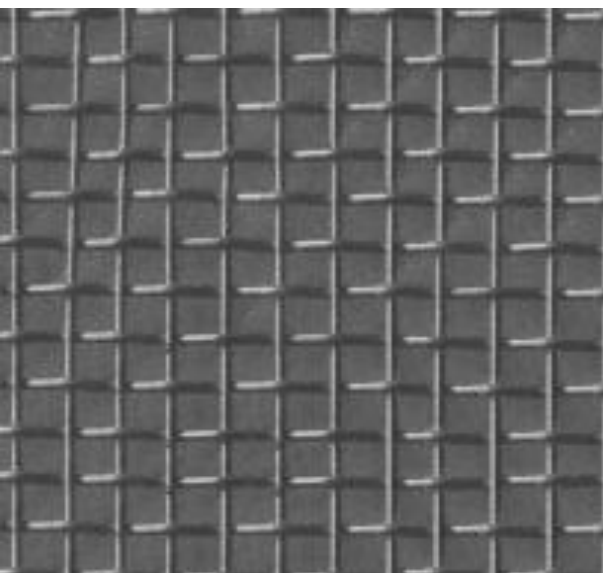
Элемент кривой – это текстон



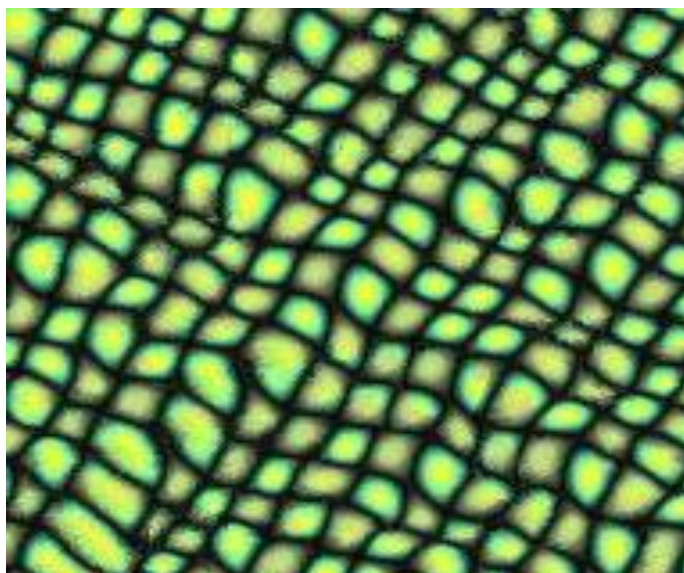
Каждая гора – это текстон

# Параметры текстур (2)

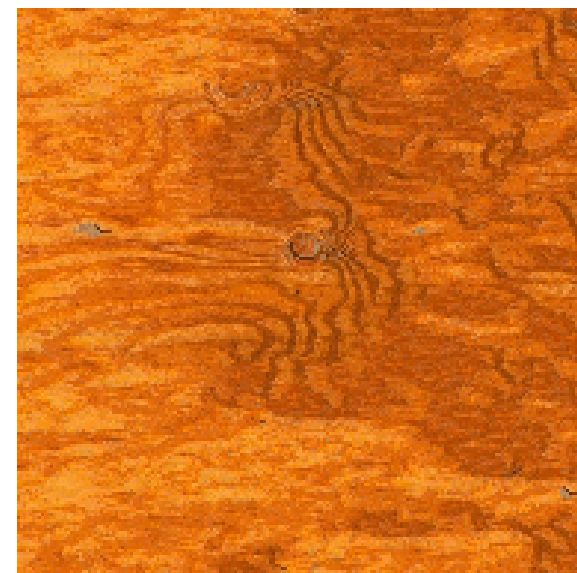
## Степень структурированности



Почти идеально повторяющиеся блоки



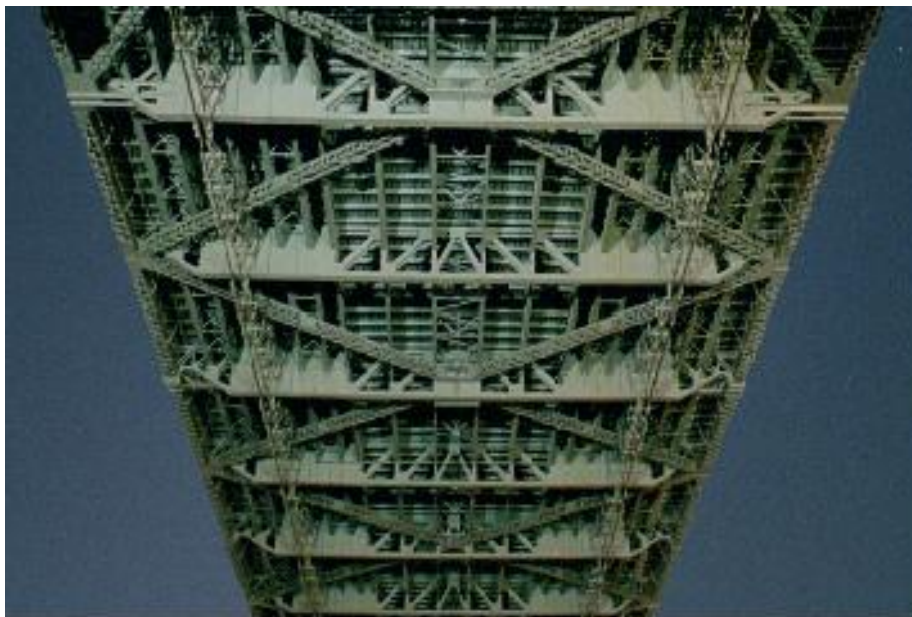
Структура есть, но не самоподобная



Хаотичная текстура

# Параметры текстур (3)

## Искажения перспективы





# Постановка задачи

- Задача: восстановить кусочно-связную область изображения
- Предположения:
  - Структурная часть кусочно-гладкая
  - Линии границ кусочно-гладкие
  - Изображение приближенно самоподобно
  - Есть необходимая для восстановления информация:
    - Семантическая — информация о природе объектов
    - Линии и цвета на границе восстанавливаемой области



# Области применения

---

- Удаление объектов
- Удаление царапин
- Изменение композиции изображения
- Построение панорамы
- Генерация стерео
- Синтез текстур

# Классификация методов

Восстановление структуры

Синтез текстуры

Комбинация синтеза текстур и восстановления структуры через декомпозицию

PDE-based методы (основанный на уравнения в частных производных)

Решение уравнения диффузии (модель передачи тепла)

Решение уравнений гидродинамика (модель поверхности воды)

Методы вариационного исчисления (минимизация интеграла)

Соединение разорванных линий

Построение аналитико-вероятностной модели

Попиксельная обработка

Обработка блоками

Обработка блоками в порядке геометрической значимости

Обработка блоками произвольной формы

Восстановление из градиента

Глобальный поиск максимального правдоподобия для системы блоков



# Содержание

---

- Введение
- **Inpainting with third-order optimal PDE**
- Методы композиции блоков
- Оптимизации
- Свой алгоритм
- Результаты тестирования

# Задача и предпосылки

- Цель:
  - Восстановить поврежденную область изображения
- Предпосылки:
  - Область достаточно велика
  - Вся необходимая информация содержится на границе
  - Изображение кусочно-постоянно

# Классический алгоритм

1. Линии, пресеченные на границе, восстанавливаются внутрь области
2. Замкнутые регионы заполняются цветом
3. Добавляется текстура

# Автоматический алгоритм

До сходимости:

1. Продлим линии краев  
Численное решение PDE  
(Partial differential equation),  
сохраняющего вдоль линий некоторую величину
2. Восстановим цвет  
Численное решение уравнения анизотропной  
диффузии

# Алгоритм

## Общее уравнение диффузии

$$\frac{\partial I}{\partial t} = \text{div}[T(x) \nabla I(x)]$$

$\nabla I$  — градиент изображения

$T(x)$  — тензор, определяющий направления и величину передачи тепла (интенсивности)

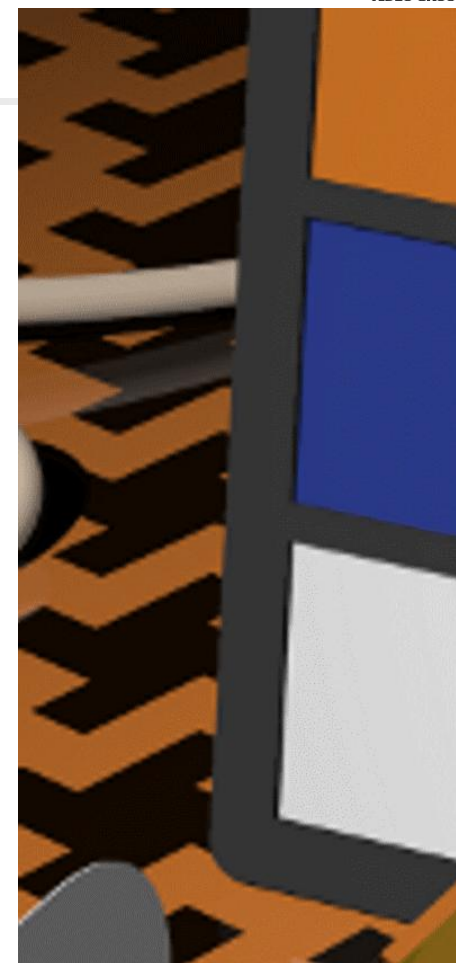
Соответственно, он должен гарантировать, что через края объектов цвет не переходит

Метод диффузии — самостоятельный. Может быть быстро реализован (несколько секунд) через аппарат термодинамики



# Алгоритм

## Результат



Результаты одного такого алгоритма на моем тесте

Marcelo Bertalmio, "Strong-Continuation, Contrast-Invariant Inpainting With a Third-Order Optimal PDE", IEEE transaction on image processing, 2006

# Алгоритм

## Принцип построения уравнений

- Общий принцип:

$$\frac{\partial I}{\partial t} = \dots \rightarrow (\text{SteadyState}) 0 = \dots$$

- В правой части обычно находится  $\nabla f \cdot \nabla^\perp I$

$f$  — инвариант, который хотим сохранить

$I$  — функция изображения

$\nabla^\perp I$  — направление линий краев

# Алгоритм

## Выбор порядка уравнения

- Для любого уравнения ниже третьего порядка, линии будут продолжены с границ, как прямые
- Так произойдет, если выбирать уравнение из условия:

$$I(x_0) = \frac{I(x_0 + h\nabla^\perp I(x_0)) + I(x_0 - h\nabla^\perp I(x_0))}{2}$$

$I$  — функция изображения

$h$  — малое приращение

$x_0$  — произвольная точка

# Алгоритм

## Выбор уравнения

$$\nabla I(x_0) = \frac{\nabla I(x_0 + h\nabla^\perp I(x_0)) + \nabla I(x_0 - h\nabla^\perp I(x_0))}{2}$$

Разложение в ряд Тейлора:

$$\nabla [k(s)|\nabla I|^3] \nabla^\perp I = 0 \rightarrow I_t = \nabla [k(s)|\nabla I|^3] \nabla^\perp I$$

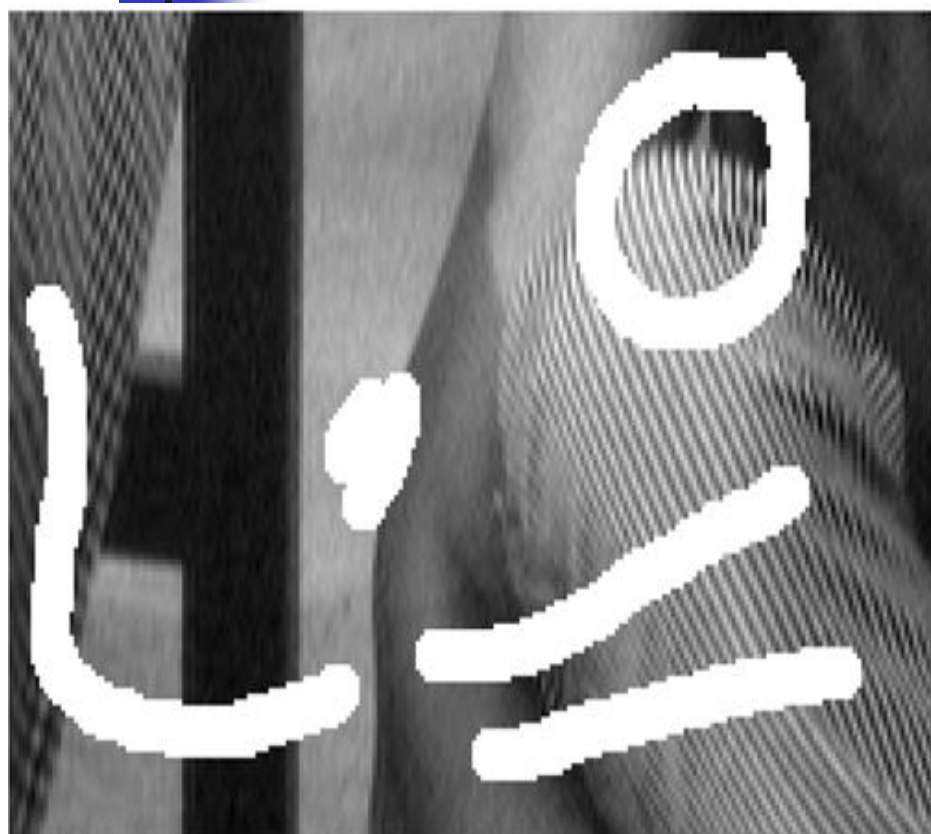
$k(s) = \gamma''(s)$  — кривизна линии (здесь  $\gamma(s)$  — параметризация кривой)

Следовательно, уравнение оптимально, если мы хотим сохранять направление линий.

То же уравнение в терминах частных производных:

$$I_t = \frac{I_y^2 [-I_y I_{xxx} + 3I_x I_{xxy}] + I_x^2 [-I_x I_{yyy} + 3I_y I_{yyx}]}{[I_x^2 + I_y^2]^{\frac{3}{2}}}$$

# Результат 1

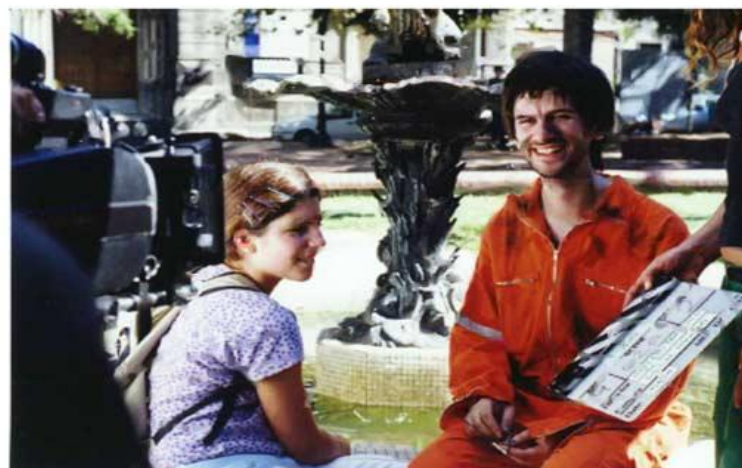


Input



Result

# Результат 2

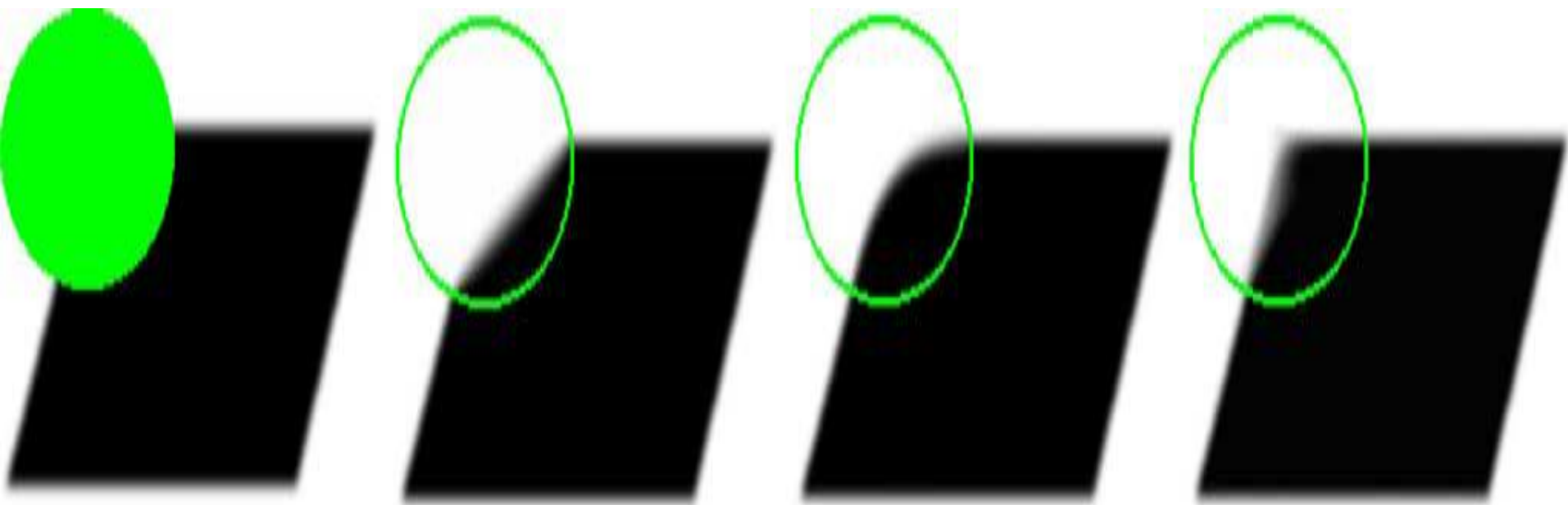


Inputs

Results

Marcelo Bertalmio, "Strong-Continuation, Contrast-Invariant Inpainting With a Third-Order Optimal PDE", IEEE transaction on image processing, 2006

# Результат 3



Input

Integral minimization  
result

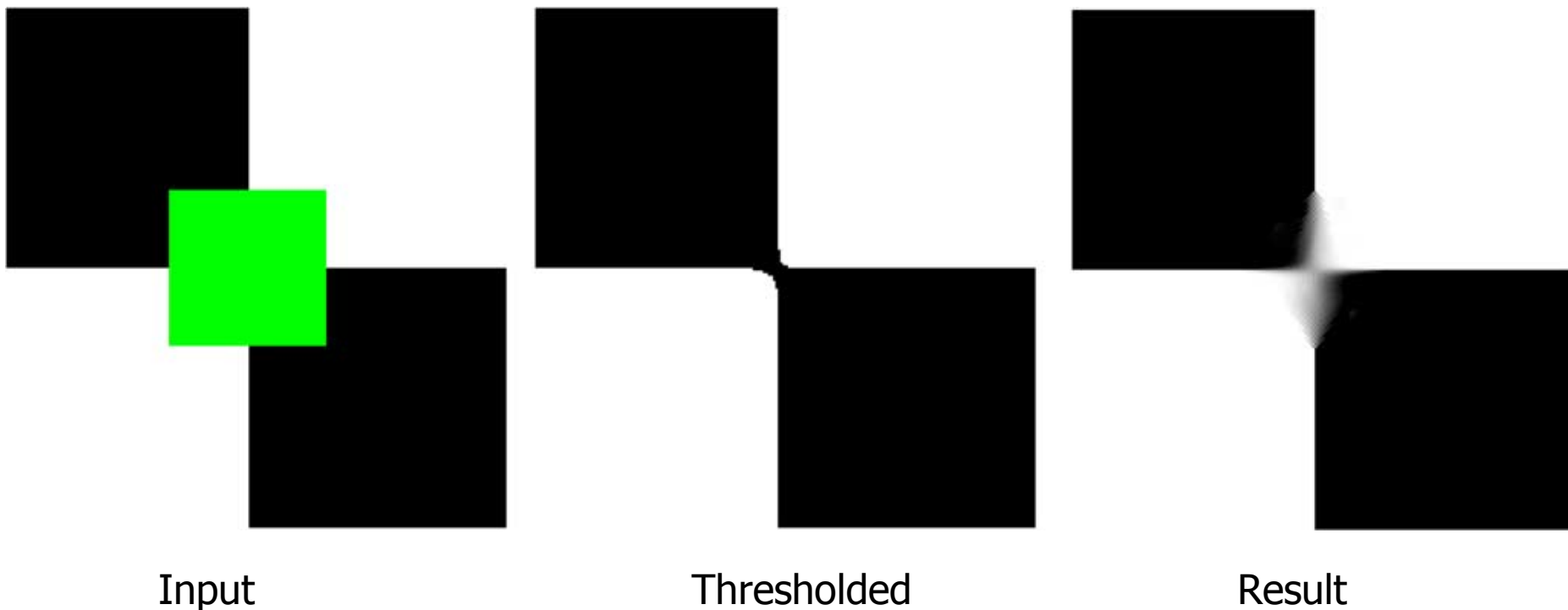
Classic inpainting  
equation

This algorithm result

Marcelo Bertalmio, "Strong-Continuation, Contrast-Invariant Inpainting With a Third-Order Optimal PDE", IEEE transaction on image processing, 2006

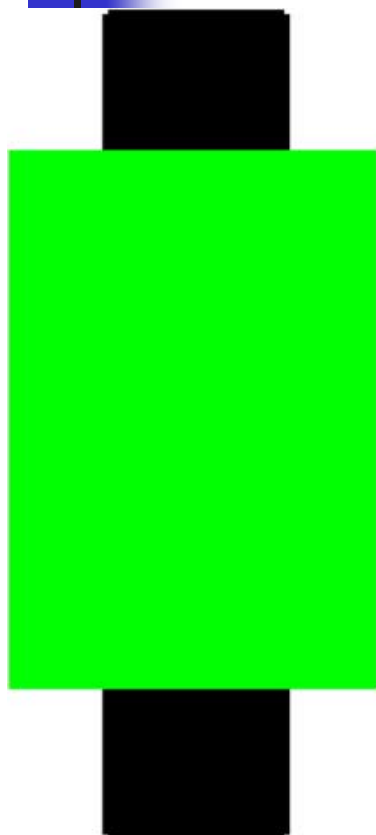


# Результат 4

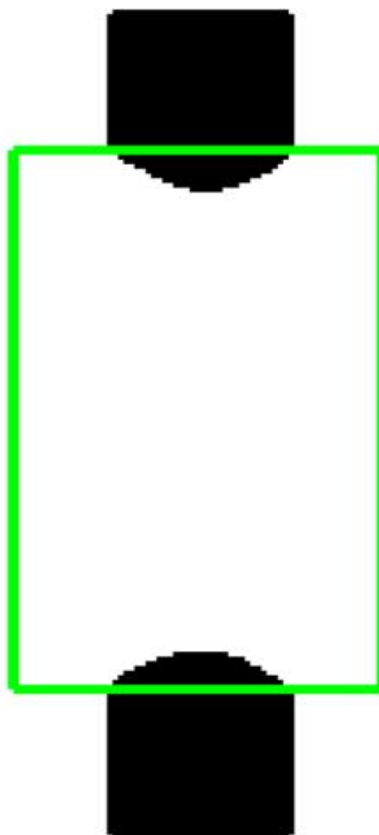




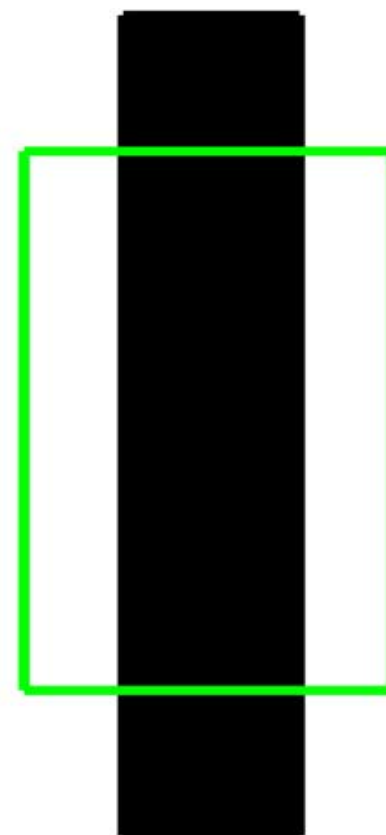
# Результат 5



Input



Classic equation  
result



This algorithm result

Marcelo Bertalmio, "Strong-Continuation, Contrast-Invariant  
Inpainting With a Third-Order Optimal PDE", IEEE transaction 25  
on image processing, 2006

# Third order optimal PDE

- Достоинства:
  - Инвариантность к контрасту
  - Возможность восстанавливать тонкие структуры и острые углы
  - Меньший blur, чем в других методах
- Недостаток:
  - Ничего не известно про время работы



# Содержание

---

- Введение
- Inpainting with third-order optimal PDE
- Методы композиции блоков
  - **Exemplar-based inpainting**
  - Belief propagation for inpainting
- Оптимизации
- Свой алгоритм
- Результаты тестирования

# Exemplar-based inpainting

## Актуальность:

- Методы, основанные на PDE (partial differential equations), не восстанавливают текстуру
- Из-за диффузии они склонны к размытию



A. Criminisi\*, P. Pérez and K. Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting", IEEE transaction on image processing, 2004

# Exemplar-based inpainting

## Идея

- Идея: использовать самоподобие в текстурах изображения
- Можно восстанавливать значение точки из точек со схожей окрестностью
  - Очень долгое время работы
  - Не восстанавливаются области, где много структуры
- Решение: восстанавливать изображение целыми блоками

A. Criminisi\*, P. Pérez and K. Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting", IEEE transaction on image processing, 2004

# Exemplar-based inpainting

## Идея

- Покроем область блоками
- Приоритет определяет порядок восстановления

$$Priority = \frac{N}{S} |\nabla^\perp I \cdot \vec{n}|$$

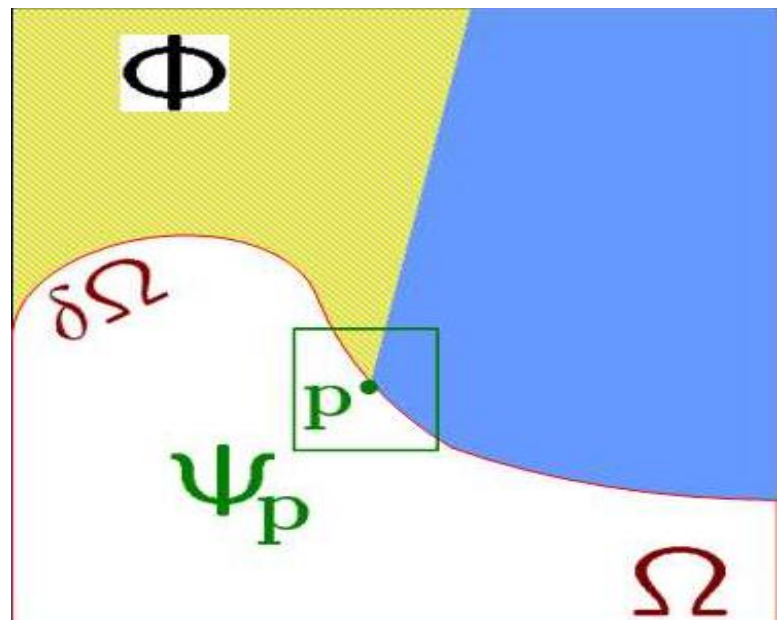
$N$  — количество известных в окрестности пикселей

$S$  — общее число пикселей в окрестности

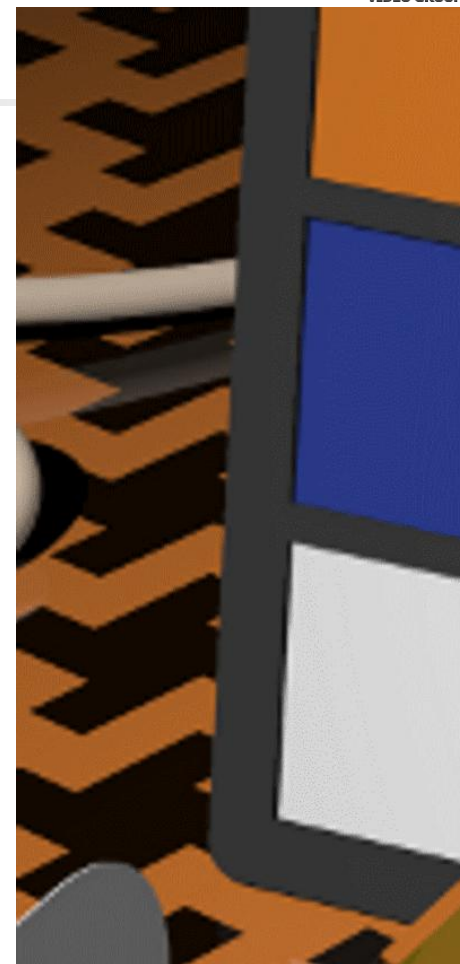
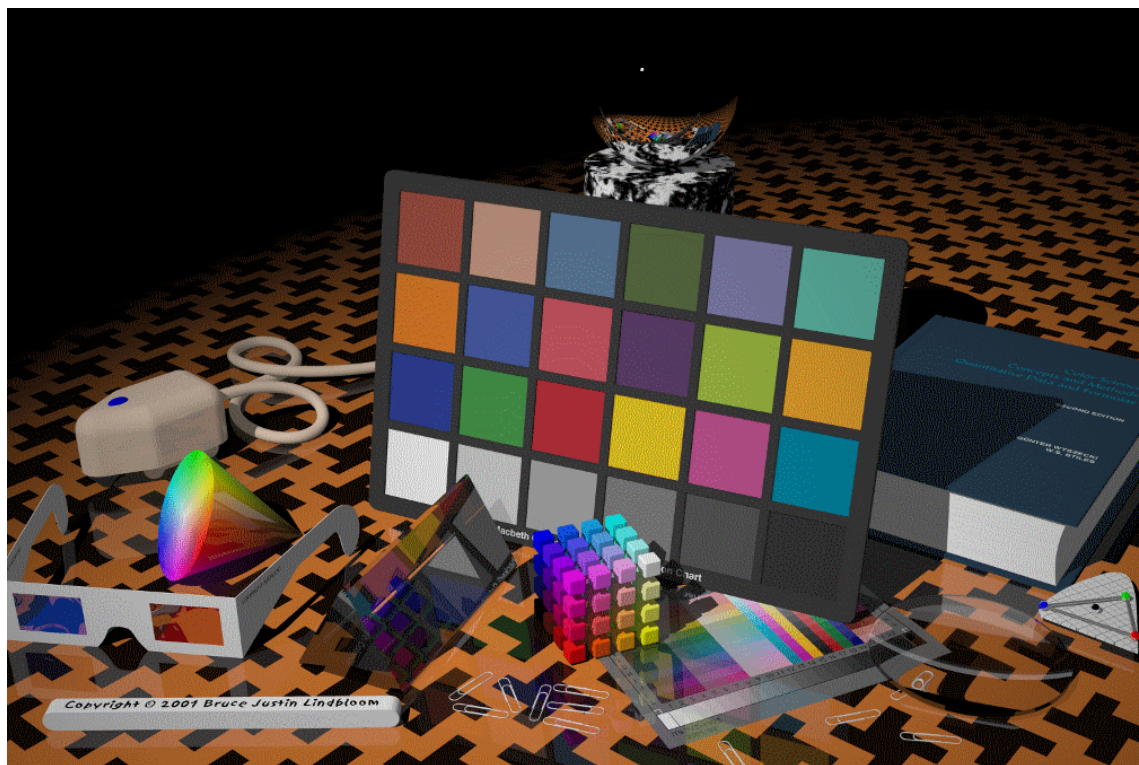
$I$  — функция изображения

$\vec{n}$  — единичная нормаль к границе

Таким образом, можно использовать и самоподобие линий



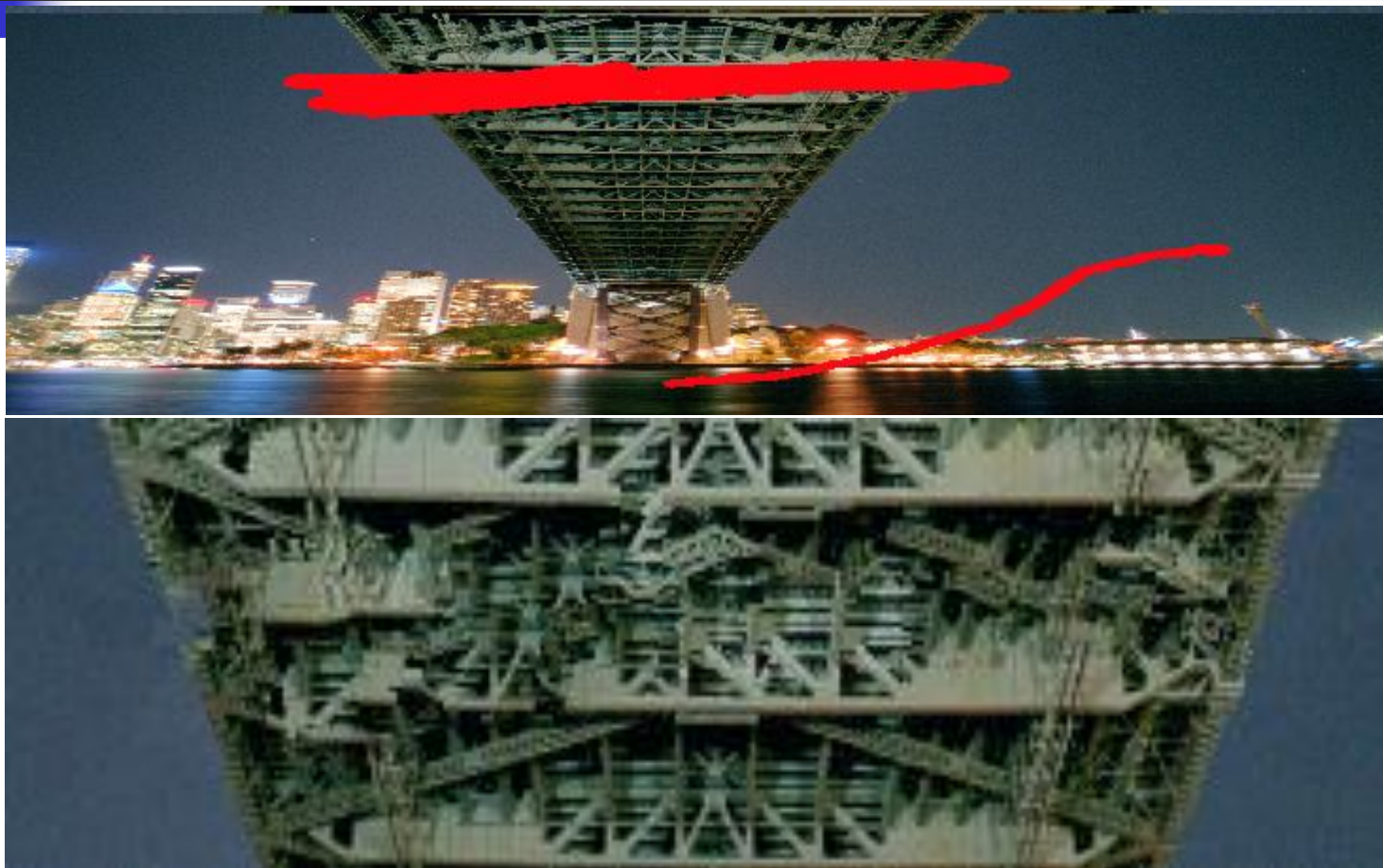
# Результат 1



Тестовое изображение и результат

A. Criminisi\*, P. Pérez and K. Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting", IEEE transaction on image processing, 2004

# Результат 2



Тестовое изображение и результат

A. Criminisi\*, P. P´rez and K. Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting", IEEE transaction on image processing, 2004





# Содержание

---

- Введение
- Inpainting with third-order optimal PDE
- Методы композиции блоков
  - Exemplar-based inpainting
  - **Belief propagation for inpainting**
- Оптимизации
- Свой алгоритм
- Результаты тестирования

# Глобальный подход

- Идея: оптимизировать систему соответствий “блок области — образец” сразу для всех пар
- Можно найти максимальное правдоподобие
- Возможные алгоритмы:
  - Expectation Minimization
  - Belief Propagation

# Belief Propagation

## Общая задача

### Классический алгоритм:

1. Пусть есть ациклический граф, в котором каждая вершина может принимать одно из  $k$  состояний
2. Известны вероятности  $\phi(x_k)$  :  
для вершины  $x$  принять состояние  $k$   
и  $\psi(x_k, y_q)$  — совместное распределение  
на  $x$  и  $y$

# Belief Propagation

## Общая задача

### Классический алгоритм:

#### 3. Максимизируем функцию правдоподобия

$$\max P = -\min [\log P] = \min_{x_1, \dots, x_n} \left[ \sum_{i=1}^n \phi_i(x_i) + \sum_{\{k,j\} \in E} \psi_{j,k}(x_j, x_k) \right]$$

- $E$  — множество ребер графа
- $\psi_{j,k}(x_j, x_k)$  —  $(-\log)$  совместного распределения на две вершины
- $\phi_i(x_i)$  —  $(-\log)$  вероятностного распределения на одну вершину
- $P$  — правдоподобие
- $x_i$  — состояние вершины  $i$

Можно решить как задачу динамического программирования в силу ацикличности. При этом перерасчет значений динамической матрицы называется передачей сообщений.

# Belief Propagation

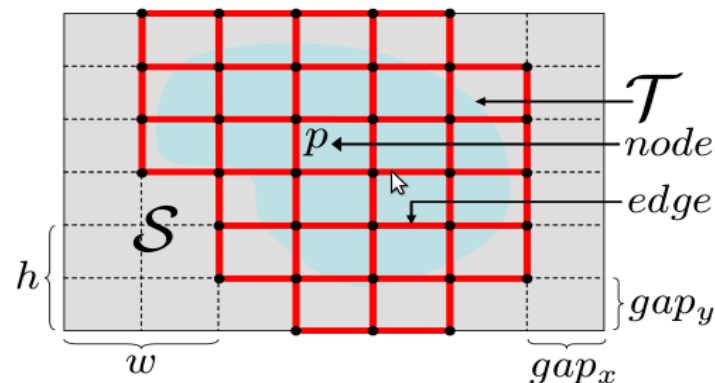
## Формулировка для inpainting

- Покроем область сетью пересекающихся блоков. Блоки — это вершины графа, перекрытия задают ребра
- Норма перекрытия двух блоков, или блока и начального приближения задает вероятности
- Граф не ациклический, но можно рассчитывать на хороший локальный минимум
- **Алгоритм квадратично зависит от количества состояний. Что делает невозможным использование алгоритма**

### Решения:

- Priority message scheduling
- Dynamic pruning

$h, w$  — ширина и высота блока  
 $gap_x, gap_y$  — ширина и высота перекрытия  
 $S, \mathcal{T}$  — изображение и восстанавливаемая область



# Belief Propagation

## Методы оптимизации

Итеративно будем выполнять алгоритм, как для ациклического графа

- Dynamic pruning — будем хранить только наиболее вероятные состояния
- Priority message scheduling — будем выполнять передачу сообщений от вершин с наименьшим числом состояний

При этом такими вершинами оказываются блоки на границах объектов. Их определенность выше

# Результат 1



Originals

Restoration order  
(Blacks are earlier)

Results

# Результат 2



Originals

Restoration order  
(Blacks are earlier)

Results



# Belief Propagation (1)

- Для изображений  $256*170$   
От нескольких секунд до двух минут,  
в зависимости от параметров и скорости  
сходимости
- Для моих тестов  $\sim 1000*800$   
От десяти до двадцати минут  
Впрочем, я связался с автором реализации  
(не автор статьи). И он с моей подачи нашел  
в ней ошибки. Так что результаты пока  
не в счет.

# Belief Propagation (2)

- Достоинство:  
Более высокое качество, чем у просто  
exemplar-based методов
- Недостаток:  
Долгое время работы



# Содержание

---

- Введение
- Inpainting with third-order optimal PDE
- Методы композиции блоков
- **Оптимизации**
  - Seamless block matching
  - Blocks deformation leading by feature-map
- Свой алгоритм
- Результаты тестирования

# Методы оптимизации

## Качество

- Расширять базу поиска за счет поворотов и отражений
- Копировать только высокие частоты (восстанавливать освещение отдельно)
- Использовать дополнительно карту краев при сравнении блоков
- Соединять блоки по линии минимального разреза
- Применять к блокам деформации по завершении работы алгоритма

# Методы оптимизации

## Время

- Использовать при поиске блоки-кандидаты
- Выполнять поиск наилучшего блока в пространстве частот
- Спроектировать окрестности блоков в пространство меньшей размерности (метод главных компонент)
- Работать в разных разрешениях, от низкого к высокому



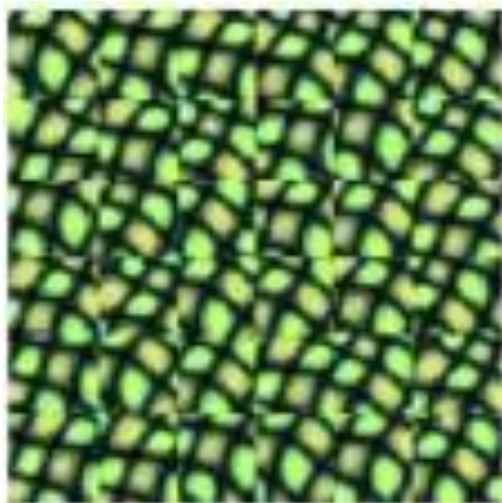
# Содержание

---

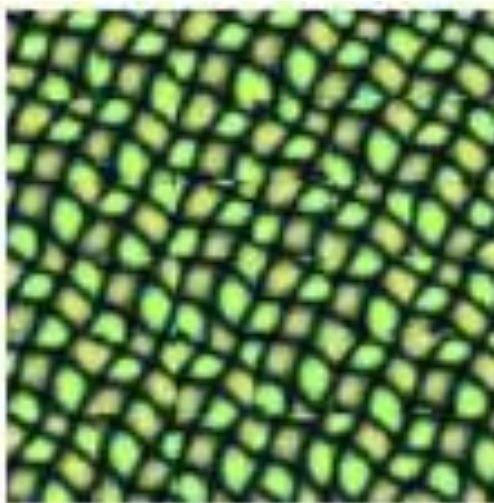
- Введение
- Inpainting with third-order optimal PDE
- Методы композиции блоков
- Оптимизации
  - **Seamless block matching**
    - Blocks deformation leading by feature map
- Свой алгоритм
- Результаты тестирования

# Произвольные блоки

Алгоритм: по области пересечения блоков найдем минимальный путь. Он будет границей между ними  
Задача может быть решена методами динамического программирования



Random copying



Searching for the  
best sample

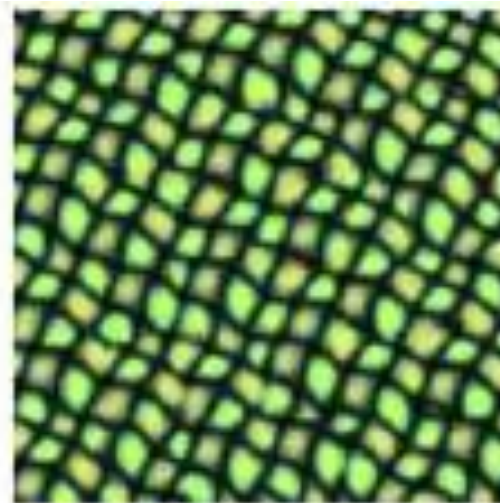


Image quilting



# Содержание

---

- Введение
- Inpainting with third-order optimal PDE
- Методы композиции блоков
- Оптимизации
  - Seamless block matching
  - **Blocks deformation leading by feature map**
- Свой алгоритм
- Результаты тестирования





# Feature map

---

Актуальность:

Даже при блоках произвольной формы  
важные текстурные линии могут  
не совпадать

Решение:

Преобразуем блоки, используя карту  
линий, так, чтобы линии на карте совпали  
Кроме того, можно учитывать соответствие  
карт при поиске наилучшего блока

# Feature map

## Построение

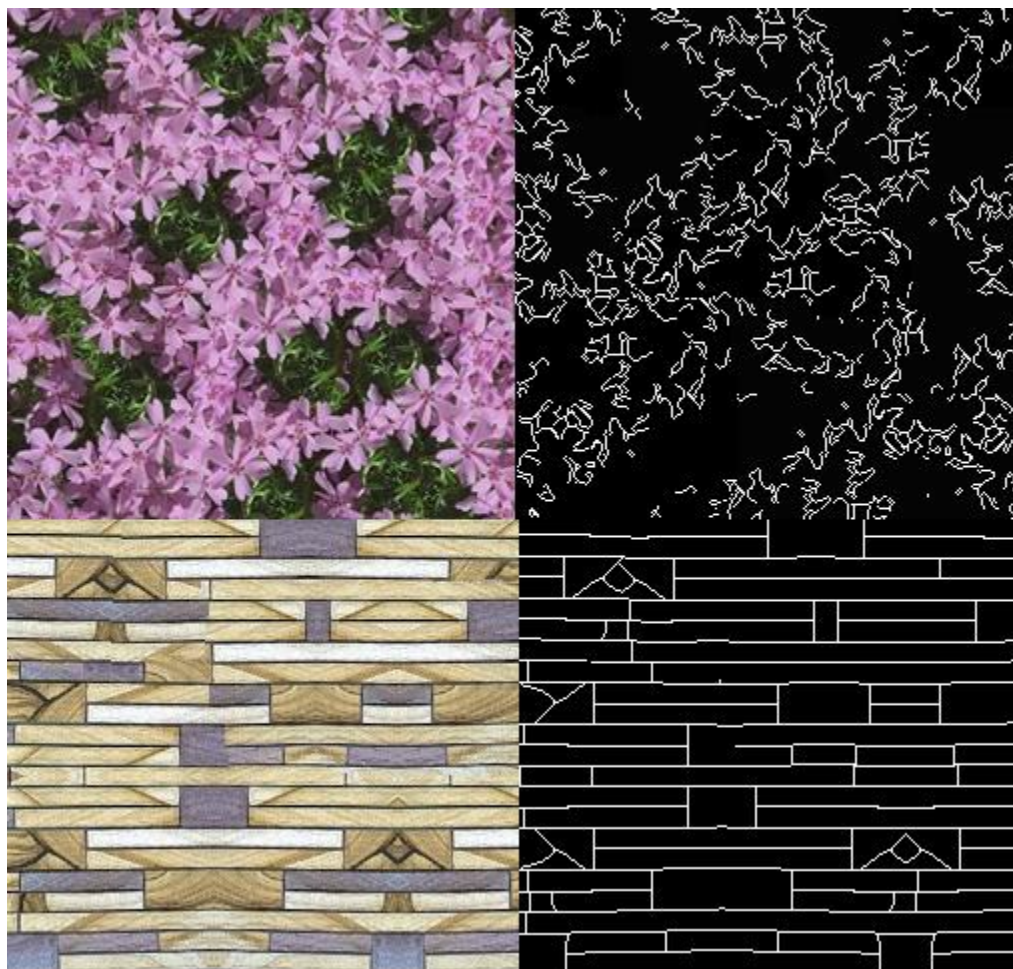


Построим по изображению карту линий (feature map):

1. Применим к изображению билатеральную фильтрацию (Параметры — Spatial  $\sigma$ : 2.0, Color  $\sigma$ : 10.0)
2. Отсечение по глобальному порогу  
Многие ребра распадутся на кусочки
3. Используя полученную на прошлом шаге карту выполним для градиентов адаптивное отсечение по порогу.  
Таким образом мы восстановим непрерывность карты
4. Выполним сужение линий (Pavlidis, T. 1982. Algorithms for Graphics and Image Processing. Computer Science Press)  
Линии будут иметь толщину в один пиксель, но не потеряется СВЯЗНОСТЬ

# Feature map

## Примеры



Originals

Feature-map

# Feature matching

- Вычислим feature map для каждого из перекрывающихся блоков
- Найдем для каждой линии одного блока наилучшее соответствие среди линий другого

$$gdist(f_i, f_j) = \|f_i - f_j\|_2 + \tau \|v_i - v_j\|_2$$

$v_i, v_j$  – множество касательных к двум линиям

$\tau$  – весовой коэффициент

$f_i, f_j$  – множество точек двух линий

Время – линейное от количества пикселей в карте (Level set method, Sethian, 1999)

# Feature deformation

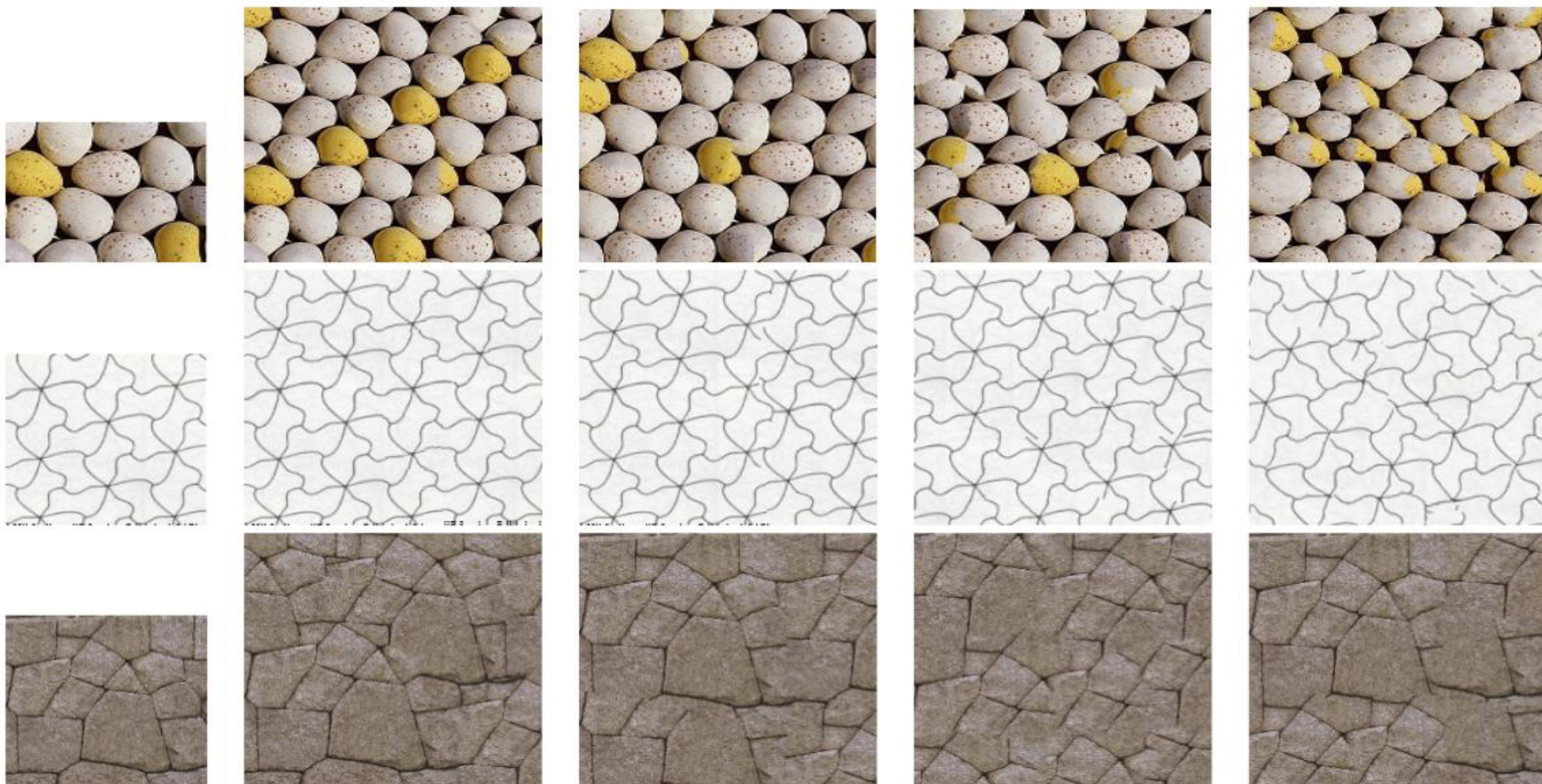
- Преобразуем блоки так, чтобы линии совпали
  - Thin-plate splines (Meinguet 1979; Turk and O'Brien 1999)
    - Решение оптимально, но может не существовать
  - Shepard's method (Hoschek and Lasser 1993)

Чтобы деформации не накапливались, потребуем также неизменности противоположного пересечению угла

# Feature map

## Сравнение

Блоки дополнительно деформировались после Graphcut



Input

Feature map

Graphcut

Quilting

Texton mask



# Содержание

---

- Введение
- Inpainting with third-order optimal PDE
- Методы композиции блоков
- Оптимизации
- **Свой алгоритм**
- Результаты тестирования



# Свой алгоритм

---

1. Разложить изображение на текстуру и структуру
2. Восстановить структуру, решив соответствующее уравнение в частных производных
3. Восстановить текстуру, используя в качестве начального приближения полученное на прошлом шаге изображение
4. Коррекция границ блоков



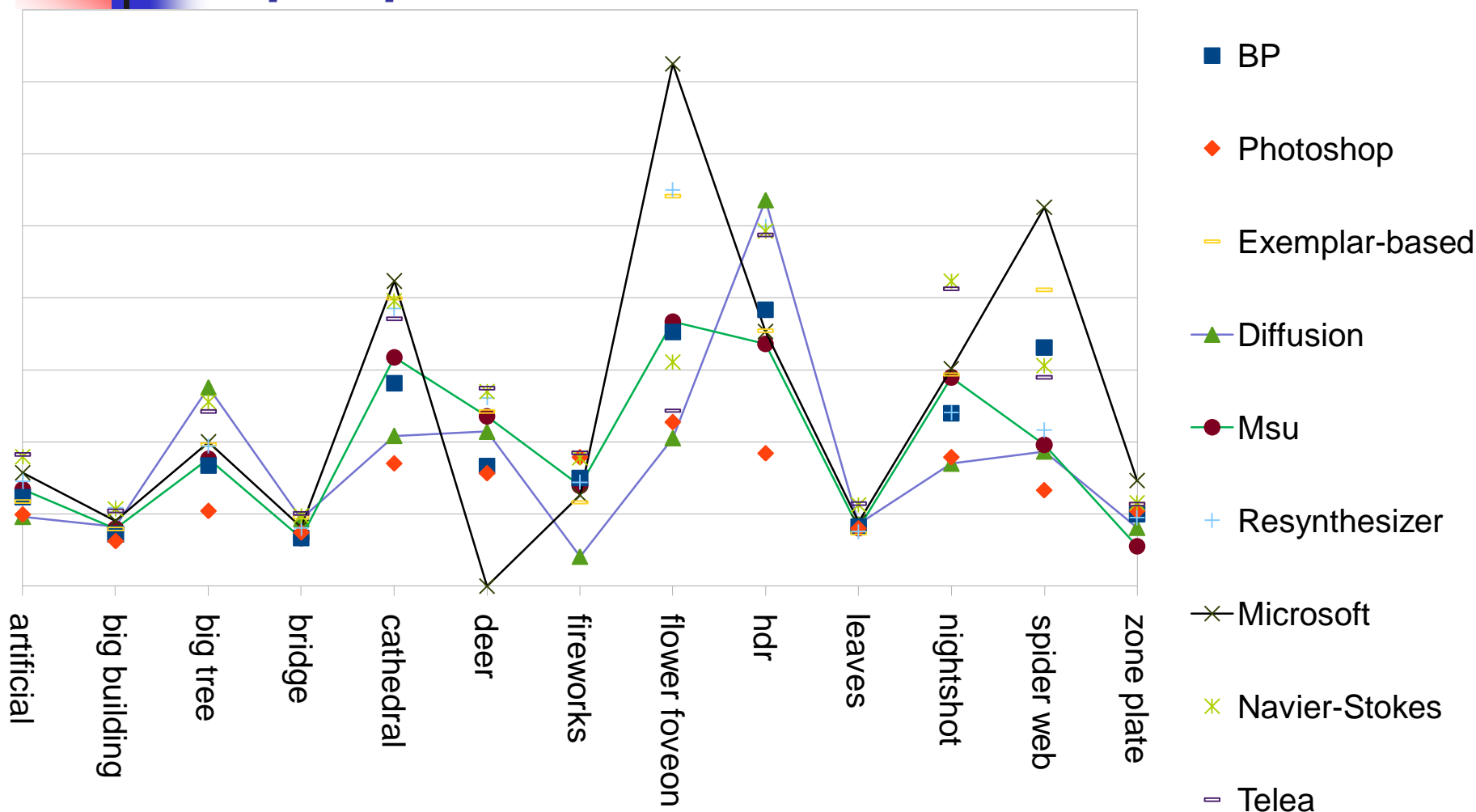


# Содержание

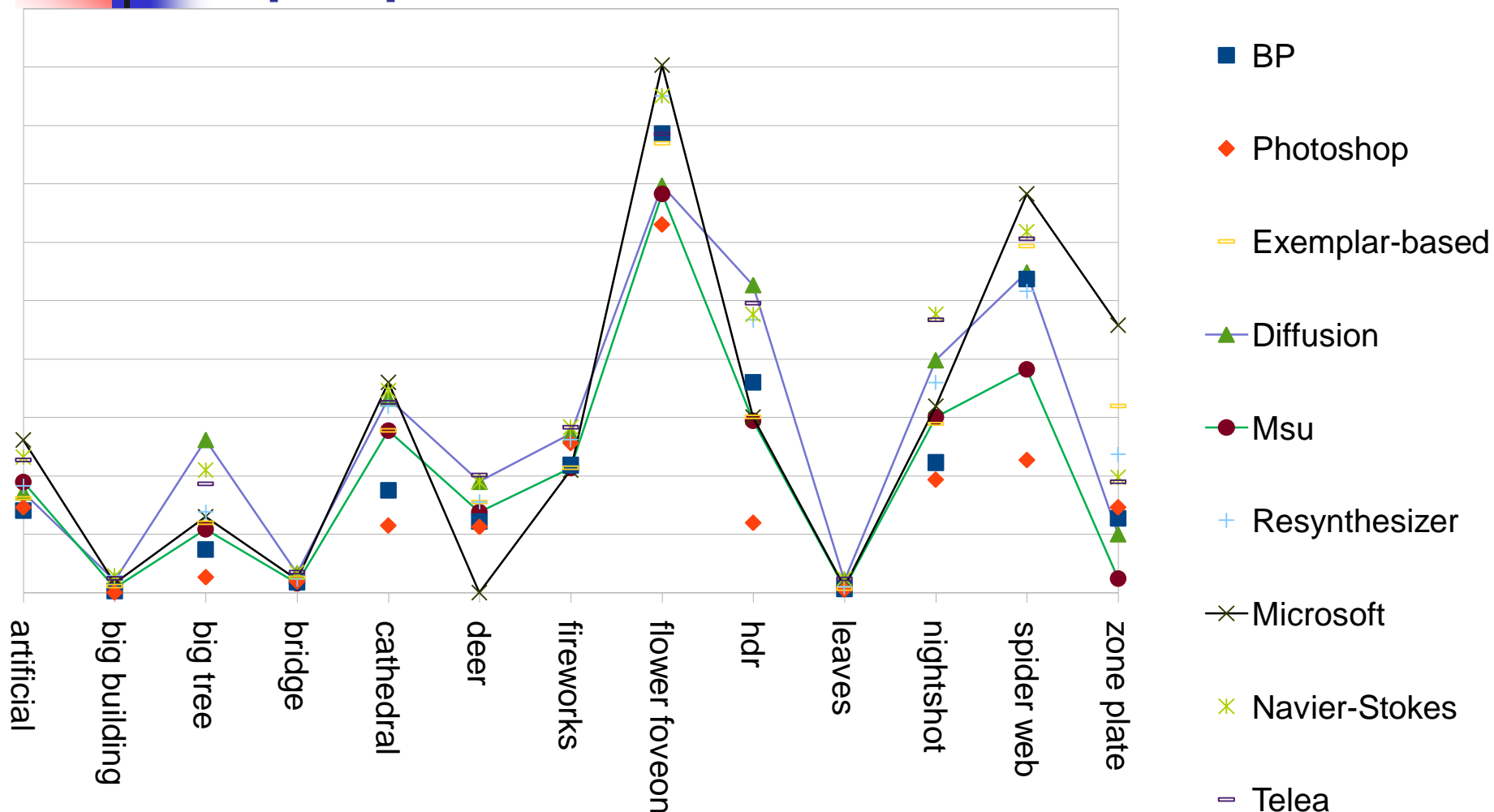
---

- Введение
- Inpainting with third-order optimal PDE
- Методы композиции блоков
- Оптимизации
- Свой алгоритм
- **Результаты тестирования**

# График значений PSNR



# График значений SSIM



# Таблицы значений PSNR и SSIM



## PSNR

BP	32.7340	28.9508	34.8511	28.4139	38.4916	34.8271	34.1002	40.0722	40.6574	29.9139	37.3633	39.6333	31.2451
Photoshop	31.1841	27.9163	31.5222	29.1532	34.9635	34.3797	35.3086	37.0004	35.5203	29.6618	35.3122	33.2333	31.5356
Exemplar-based	32.3442	29.5895	35.9693	30.9012	40.9306	37.4371	32.2757	43.0608	40.0990	29.0493	38.7773	41.1361	32.1046
Diffusion	30.9719	29.9058	38.3431	30.7347	36.3907	36.5815	24.9746	36.2839	42.9952	30.1752	34.9606	35.6216	29.7648
Msu	33.3073	29.6613	35.2125	28.3692	39.3317	37.2385	33.5679	40.3404	39.7282	29.6825	38.6857	35.9437	27.0750
Resynthesizer	33.8617	31.8361	35.9024	29.7342	40.6878	37.9655	33.8093	43.1799	42.4942	29.2645	37.3906	36.6510	30.8831
Microsoft	34.4192	30.5231	36.0935	29.8582	41.3461	—	32.8805	45.1073	40.0853	30.4318	38.9781	42.8584	33.9142
Navier-Stokes equation	35.3224	31.6809	37.8131	31.0477	40.8727	38.1899	35.2735	39.1881	42.4104	32.0670	41.3435	39.0662	32.2391
Telea method	35.4571	31.5489	37.4456	31.3030	40.4231	38.3141	35.5583	37.4722	42.3304	32.1974	41.1649	38.6914	32.1663

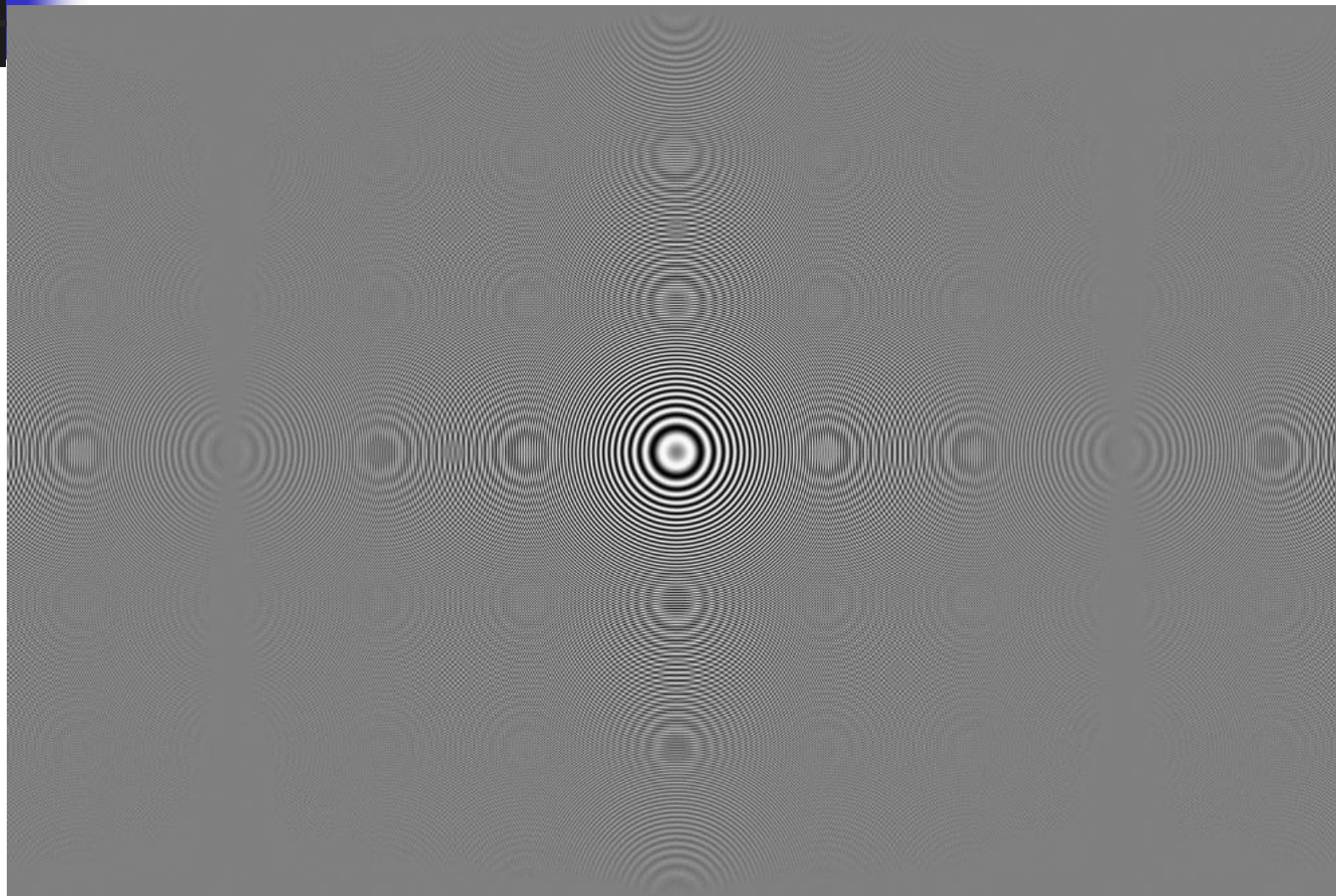
## SSIM

BP	0.9889	0.9698	0.9857	0.9787	0.9900	0.9882	0.9911	0.9975	0.9936	0.9735	0.9912	0.9956	0.9884
Photoshop	0.9891	0.9607	0.9806	0.9790	0.9879	0.9878	0.9919	0.9964	0.9881	0.9730	0.9905	0.9913	0.9891
Exemplar-based	0.9896	0.9762	0.9881	0.9804	0.9923	0.9894	0.9910	0.9974	0.9927	0.9735	0.9925	0.9961	0.9930
Diffusion	0.9899	0.9793	0.9920	0.9816	0.9932	0.9904	0.9922	0.9969	0.9955	0.9794	0.9941	0.9957	0.9872
Msu	0.9904	0.9748	0.9876	0.9781	0.9923	0.9888	0.9910	0.9968	0.9926	0.9749	0.9927	0.9939	0.9801
Resynthesizer	0.9902	0.9810	0.9888	0.9799	0.9930	0.9894	0.9920	0.9979	0.9949	0.9756	0.9936	0.9954	0.9915
Microsoft	0.9920	0.9781	0.9885	0.9796	0.9936	—	0.9909	0.9982	0.9927	0.9755	0.9930	0.9968	0.9948
Navier-Stokes equation	0.9914	0.9809	0.9909	0.9819	0.9934	0.9904	0.9924	0.9979	0.9950	0.9801	0.9950	0.9963	0.9906
Telea method	0.9913	0.9802	0.9903	0.9820	0.9931	0.9907	0.9924	0.9975	0.9952	0.9800	0.9949	0.9962	0.9904

Синим и зеленым отмечены первый и второй результат по тесту



# Microsoft (smart erase) 13



# Telea method (OpenCV)



# Diffusion (gmic) 9



# Exemplar-based (Exinpaint) 7





# Литература

1. Marcello Bertalmio, Guillermo Sapiro, Vicent Caselles, Coloma Ballester, "Image inpainting", siggraph 2000
2. Jean-Luc Starck, Michael Elad, David Donoho, "Image decomposition via the combination of sparse representation and variational approach", IEEE transactions on image processing 2005
3. Nikos Comodakis, Georgios Tziritas, "Image completion using efficient Belief Propagation via priority scheduling and dynamic pruning", IEEE transactions on image processing 2007
4. Vivek Katra, Amo Schodl, irfan Essa, Greg Turk, Aaron Bobick, "Graphcut textures: Image and video synthesis using Graph Cut", siggraph 2003
5. Jianbing Shen, Xiaogang Jin, Chuan Zhou, Charlie Wang, "Gradient-based image completion by solving the Poisson equation", Computers & graphics 2007
6. Qing Wu, Yizhou Yu, "Feature matching and deformation for texture synthesis", ACM transactions on graphics 2004
7. Sylvain Lefebvre, Hugues Hoppe, "Appearance-space texture synthesis", ACM transactions on graphics 2006
8. Marie-Flavie Auclair-Fortier, Djemel Ziou, "Global approach for solving evolutive heat transfer for image denoising and inpainting", IEEE transactions on image processing 2006
9. David Tschumperle, "Fast Anisotropic Smoothing of Multi-Valued Images using Curvature-Preserving PDE's", International journal of computer vision 2006
10. Antoni Buades, Triet Le, Jean-Michel Morel, and Luminita Vese, "Fast Cartoon + Texture Image Filters", IEEE transactions on image processing 2010

# Литература

11. Isik Baris Fidaner, "A survey on variational image inpainting, texture synthesis and image completion" 2008.
12. Alexei Efros and Thomas Leung, "Texture synthesis by non-parametrical samplings", seventh international conference on computer vision 1999.
13. Vivek Kwatra, Irfan Essa, Aaron Bobick, Nipun Kwatra, "Texture optimization for example-based synthesis" 2005.
14. Marcelo Bertalmio, "Strong-Continuation, Contrast-invariant Inpainting with third-order optimal PDE", IEEE transactions on image processing 2006
15. Marcello Bertalmio, Luminita Vesse, Guillermo Sapiro, Stanley Osher, "Simultaneous structure and texture image inpainting", IEEE transactions on image processing 2003
16. Criminisi, Perez, Toyama, "Region filling and object removal by exemplar-based image inpainting" IEEE transactions on image processing 2004.
17. Frederic Cao, Yann Gousseau, Simon Masnou, Patrick Perez, "Geometrically guided exemplar-based inpainting", SIAM journal on imaging sciences 2011
18. Patrick Perez, Michel Gangnet, Andrew Blake, "Poisson image editing", siggraph 2003
19. Sylvain Lefebvre, Hugues Hoppe, "Parallel controllable texture synthesis", ACM transactions on graphics 2005
20. Alexei Efros, William Freeman, "Image quilting for texture synthesis and transfer", siggraph 2001

# Лаборатория компьютерной графики и мультимедиа



Видеогруппа — это:

- Выпускники в аспирантурах Англии, Франции, Швейцарии (в России в МГУ и ИПМ им. Келдыша)
- Выпускниками защищено 5 диссертаций
- Наиболее популярные в мире сравнения видеокодеков
- Более 3 миллионов скачанных фильтров обработки видео