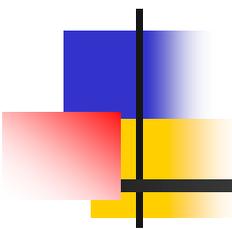
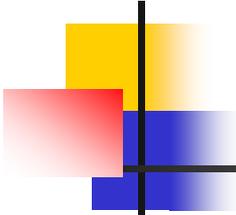


Настоящее и Будущее 3D видео



Ватолин Дмитрий Сергеевич

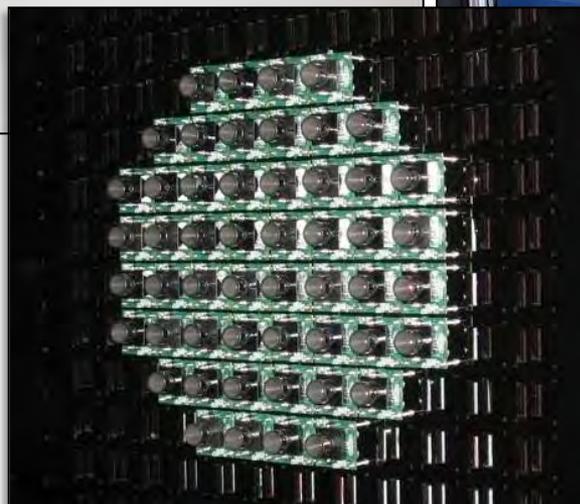
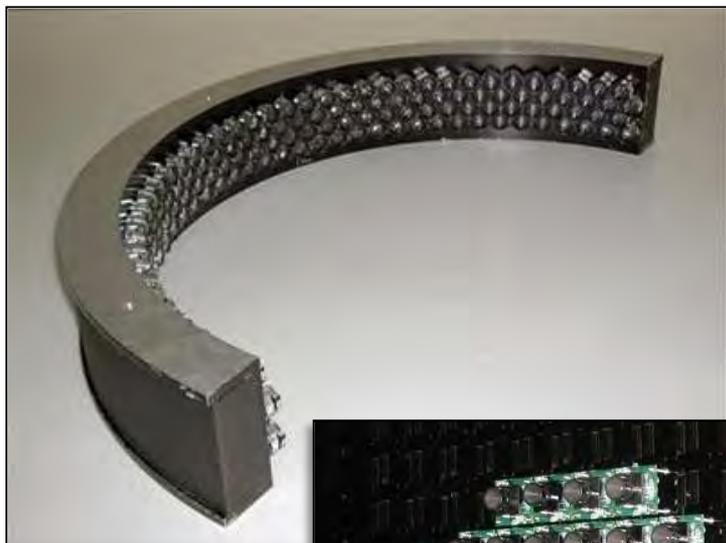
CS MSU Graphics & Media Lab



Содержание

- **Современный 3D-контент**
 - **Многокамерные системы**
 - Matrix Bullet Time
 - Toshiba Timesculpture
- **Создание 3D-контента**
 - Создание карт глубины
 - Методы обработки карт глубины
 - Построение новых ракурсов

Многокамерные системы



Многокамерные системы

Линейная система



100 камер

Многокамерные системы

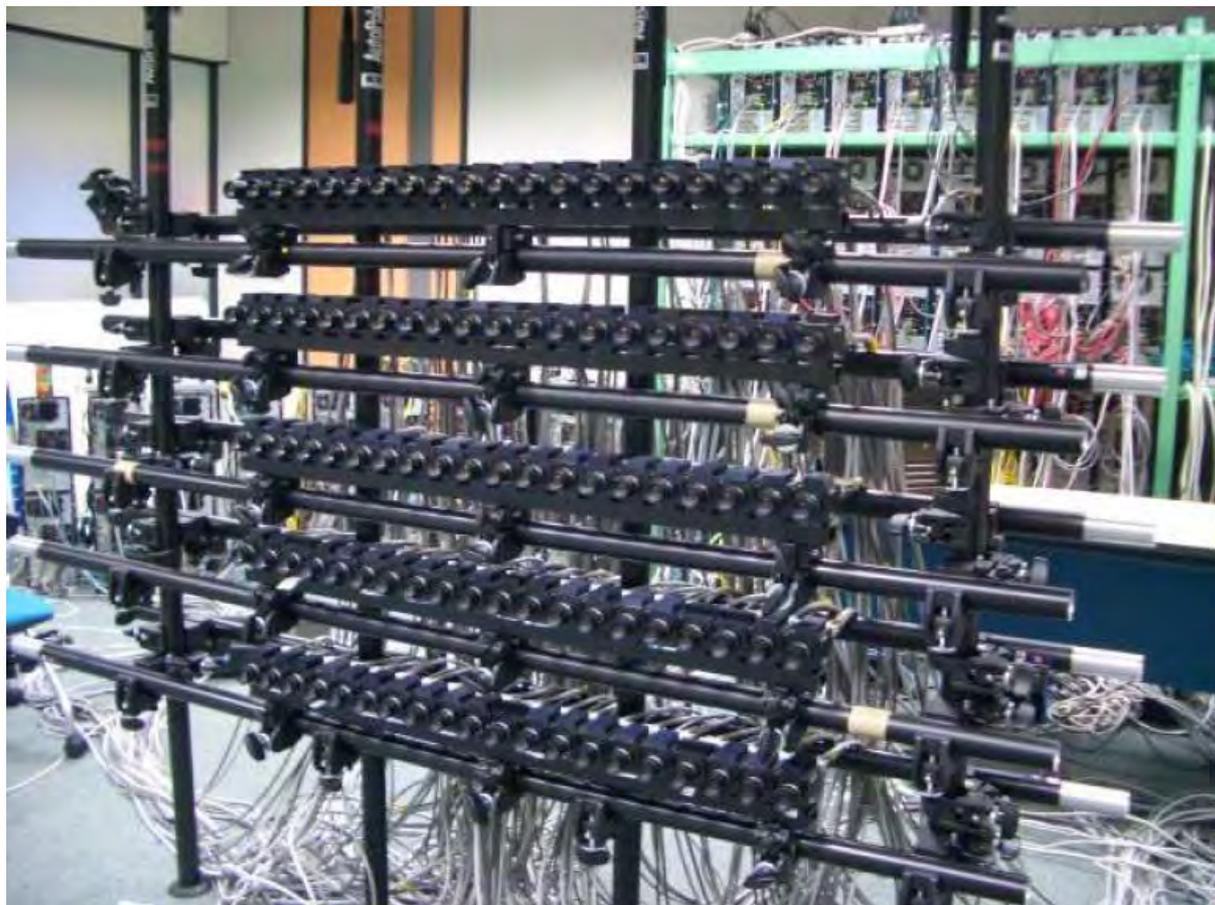
Фокусированная система



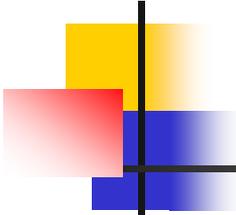
100 камер, окружность 5м

Многокамерные системы

Блочная система



5 × 20 камер



Содержание

- **Современный 3D-контент**
 - Многокамерные системы
 - **Matrix Bullet Time**
 - Toshiba Timesculpture
- **Создание 3D-контента**
 - Создание карт глубины
 - Методы обработки карт глубины
 - Построение новых ракурсов

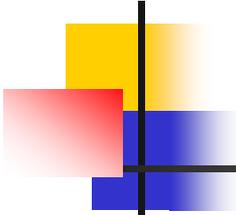
Современный 3D-контент

Matrix Bullet Time



0026





Содержание

- **Современный 3D-контент**
 - Многокамерные системы
 - Matrix Bullet Time
 - **Toshiba Timesculpture**
- **Создание 3D-контента**
 - Создание карт глубины
 - Методы обработки карт глубины
 - Построение новых ракурсов

Современный 3D-контент

Toshiba Timesculpture



Toshiba Timesculpture

Инженерные работы

- 200 серийных камер Toshiba Gigashot HD
- 1.5 месяца монтажных работ



informitv.com



Toshiba Timesculpture

The making of (1)



Съемка каждого объекта в отдельности (видеоряд "Rachel")

Toshiba Timesculpture

The making of (2)



Обработка каждого кадра практически вручную (маска объекта "Rachel")

Toshiba Timesculpture

Результат



Итог в цифрах:

- 2 500 000 уникальных кадров
- 20 Тбайт рабочего материала
- 1 минута ИТОГОВОГО ролика



Содержание

- Современный 3D-контент
 - Многокамерные системы
 - Matrix Bullet Time
 - Toshiba Timesculpture
- **Создание 3D-контента**
 - **Создание карт глубины**
 - Методы обработки карт глубины
 - Построение новых ракурсов

Создание 3D-контента

- Многокамерные системы – слишком дорогостоящий инструмент создания 3D
- Альтернатива – создание 3D-контента из 2D
- Основной шаг – вычисление пространственного представления сцены
- Для этого строятся карты глубины кадров

Карты глубины

Каждой точке изображения сопоставляется «глубина»,
удаленность реальной точки сцены от зрителя



Карта глубины получена с помощью ручной разметки кадров

Карты глубины

Автоматические способы построения

- Depth Effects for Landscape Scenes
- Depth from Motion
- Depth from Focus
- Depth from Geometry
- Stereo to Multiview conversion
(Optical-Flow based)

Depth Effects



Автоматически построенная карта глубины для исходного видео

Depth Effects

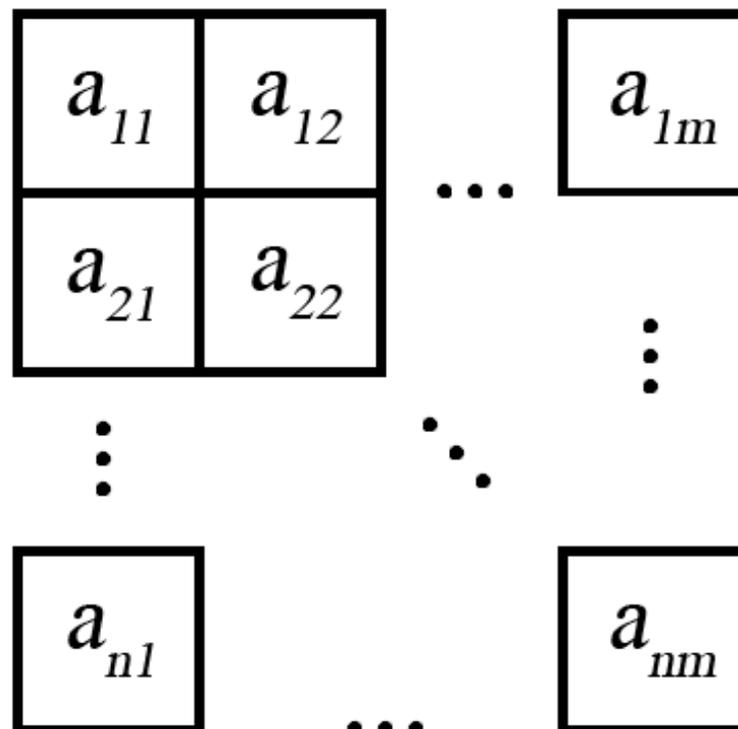
Идея алгоритма

- Во многих сценах передний план расположен внизу кадра, а дальний — вверху
- За исходную глубину принимается обычный градиент от чёрного к белому
- С помощью билатеральной фильтрации глубина распространяется по объектам

Линейная фильтрация

Оператор свертки (1)

- Применение линейного оператора свертки к изображению – замена пикселя взвешенным средним по окрестности
- Веса определяют ядро свертки – числовую матрицу

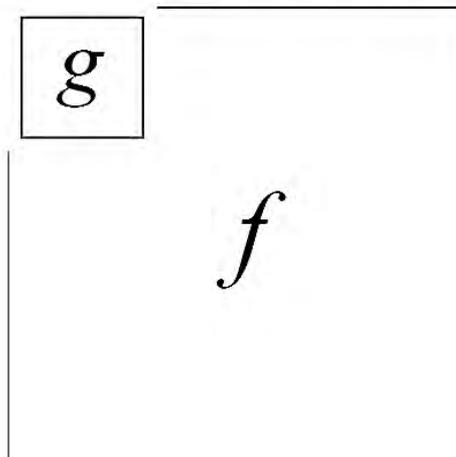


Линейная фильтрация

Оператор свертки (2)

- f – изображение $M \times N$, g – ядро $K \times L$
- Свертка f с помощью g – изображение $(f * g)$:

$$(f * g)[m, n] = \sum_{k, l} f[m - K/2 + k, n - L/2 + l] g[k, l]$$



Линейная фильтрация

Поведение на границе



Черный фон



Зеркалирование



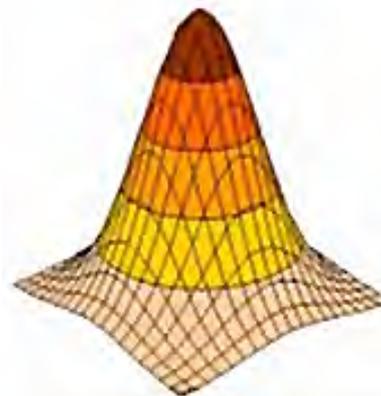
Дублирование

Линейная фильтрация

Фильтр Гаусса

- Веса в ядре рассчитываются из нормального распределения $N(0, \sigma^2)$

$$f(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



- Параметры фильтрации:
размер ядра d ,
дисперсия σ^2
- Эмпирически выбирается $d \sim 6\sigma$

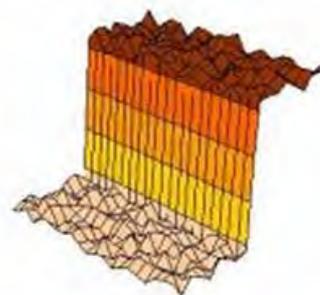
0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$

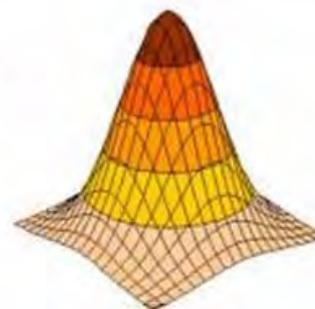
Линейная фильтрация

Билатеральная фильтрация

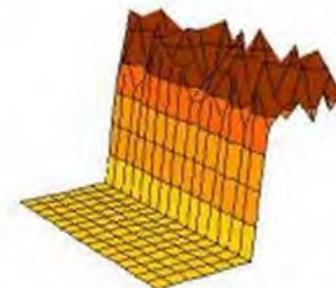
- Билатеральная – два параметра для расчета весов
- Веса исходного фильтра Гаусса модифицируем с учетом цветовой разницы центрального пикселя и пикселя окрестности



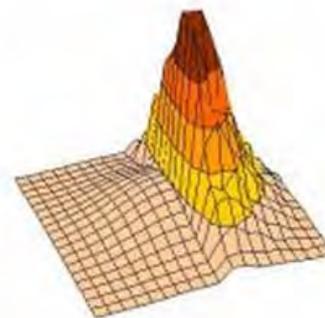
Исходное изображение



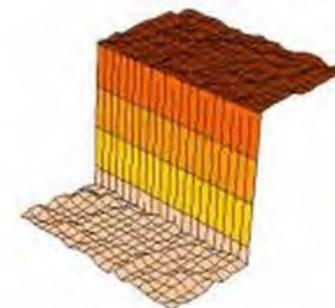
Функция Гаусса



Функция в домене интенсивности для центрального пикселя



Изменение функции Гаусса с учётом интенсивности



Результат работы билатерального фильтра

Линейная фильтрация

Кросс-билатеральная фильтрация

Веса билатерального ядра для исходного изображения применяются к карте глубины

$$D(x_0, y_0) = \frac{1}{2\pi\sigma_s\sigma_c} * \sum_{(x_1, y_1) \in \Omega} D(x_1, y_1) * e^{-\frac{(x_0-x_1)^2+(y_0-y_1)^2}{2\sigma_s^2} - \frac{(I(x_0, y_0)-I(x_1, y_1))^2}{2\sigma_c^2}}$$

$D(x, y)$ – глубина точки (x, y)

$I(x, y)$ – цвет точки (x, y)

Ω – область ядра свертки

σ_s – параметр пространственного усреднения

σ_c – параметр цветового усреднения

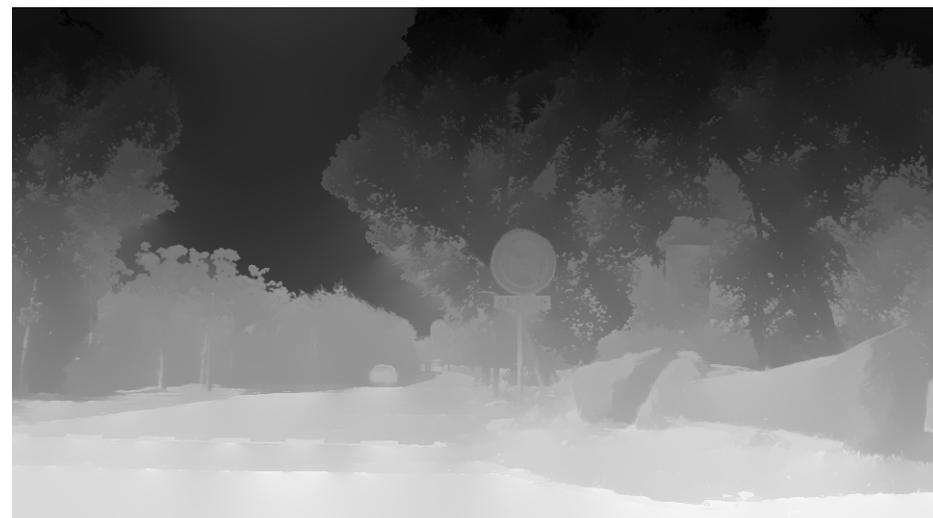
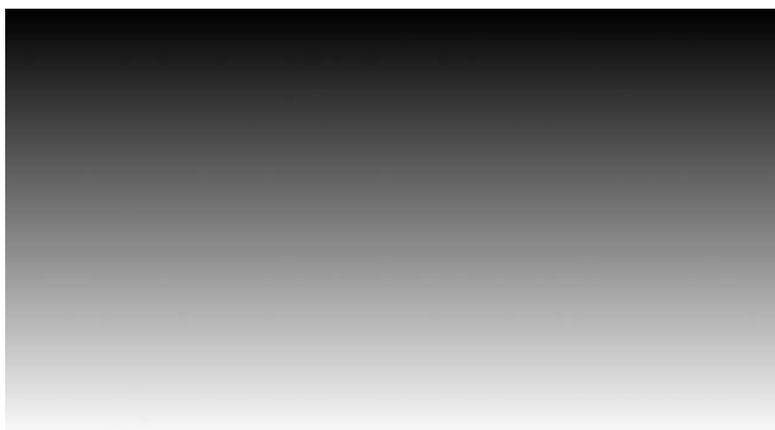
Depth Effects

Идея алгоритма



+

=



Depth Effects

Результаты (1)



Исходное изображение

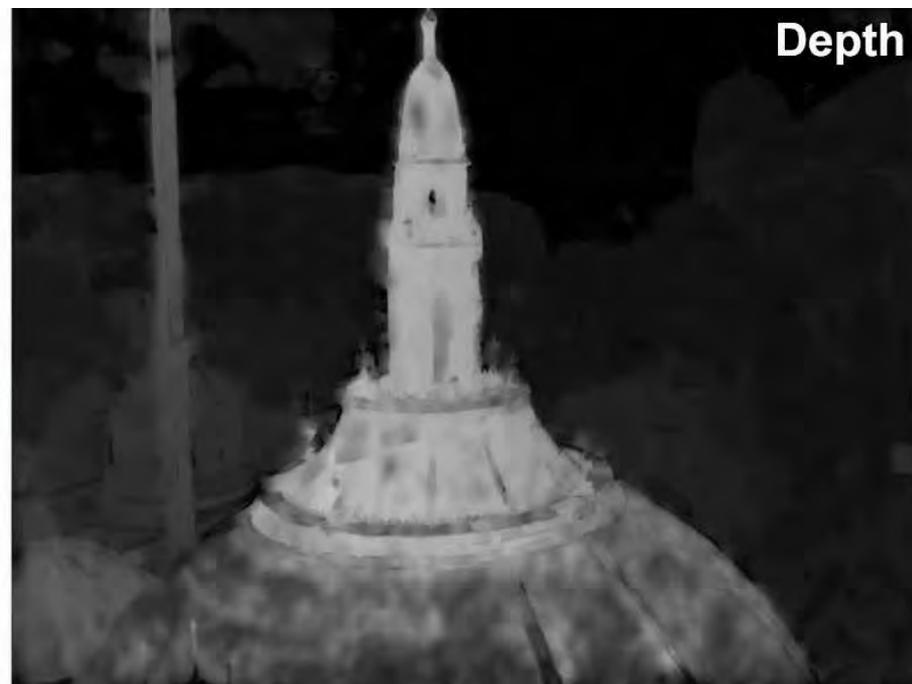
Depth Effects

Результаты (2)



Карта глубины

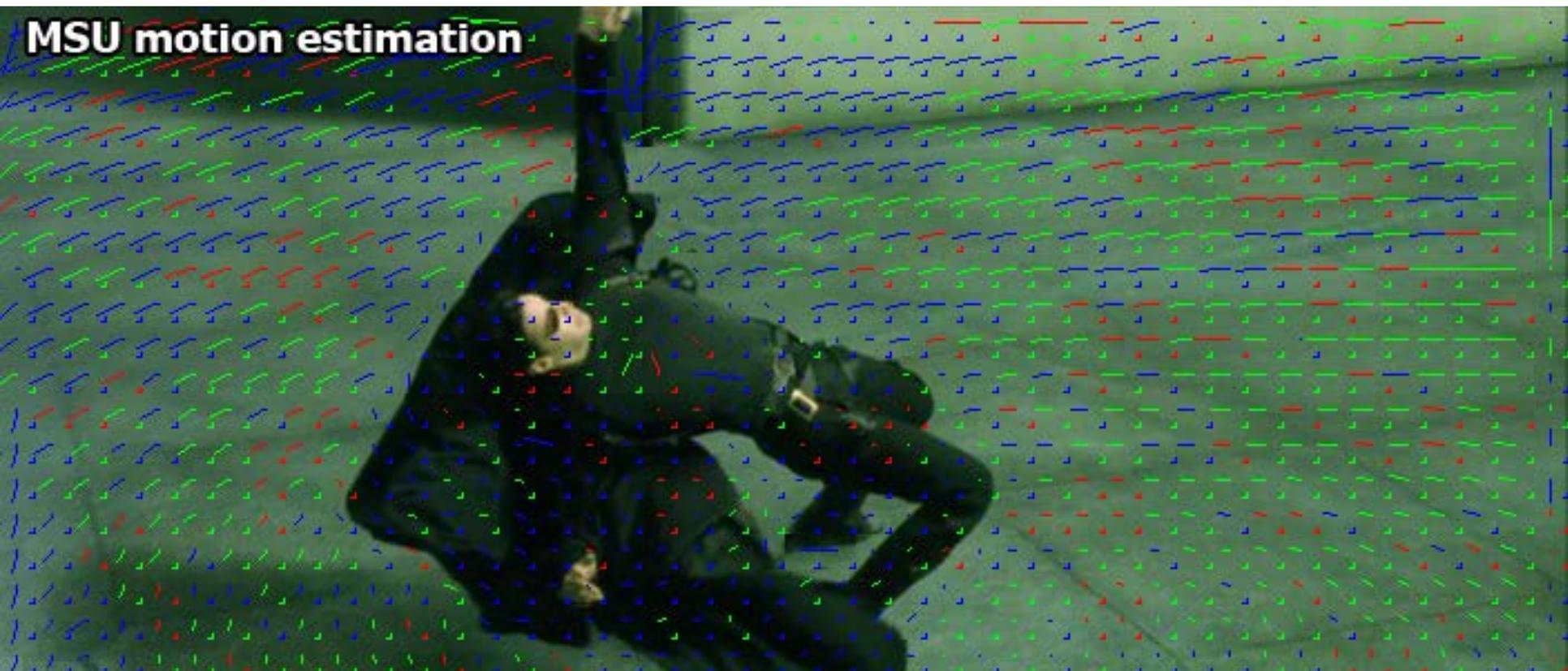
Depth from Motion



Автоматически построенная карта глубины для исходного видео

Depth From Motion

Компенсация движения

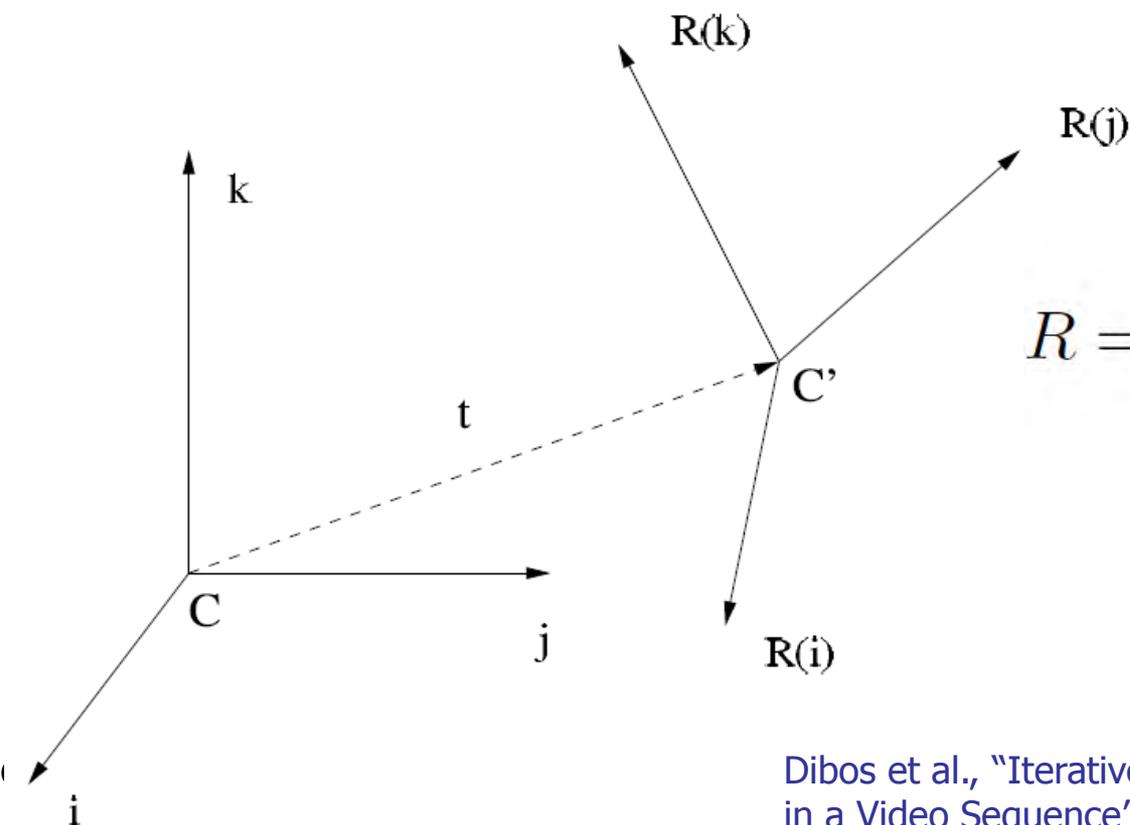


Оценка движения между соседними ракурсами

Depth From Motion

Определение движения камеры

Из анализа движения восстанавливается проективное преобразование кадров и движение камеры



$$R = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \quad t = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}$$

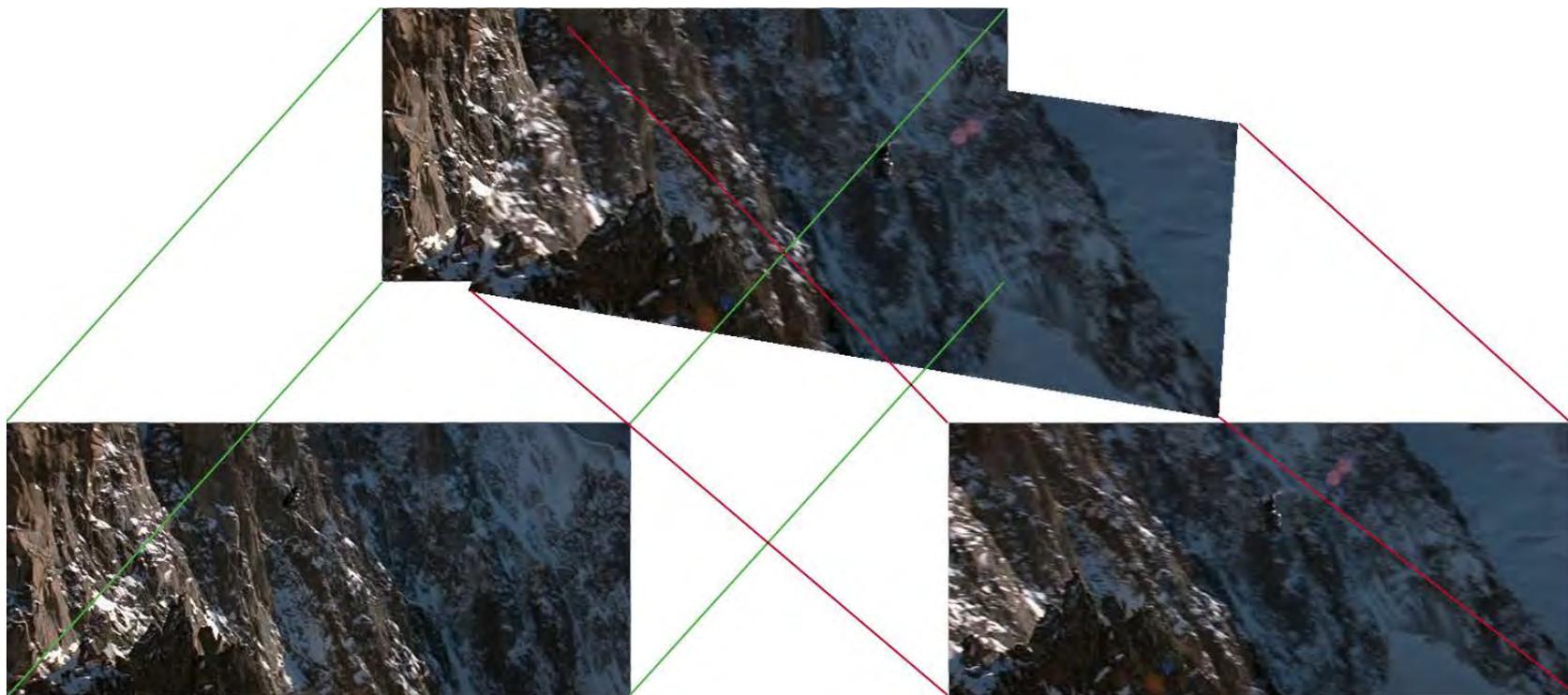
R – матрица поворота

t – вектор сдвига

Depth From Motion

Сопоставление кадров

На этапе сопоставления кадров повторно производится компенсация движения



Depth From Motion

Результат (Видео)



Чем больше смещение точки между кадрами, тем ближе она к зрителю

Depth From Motion

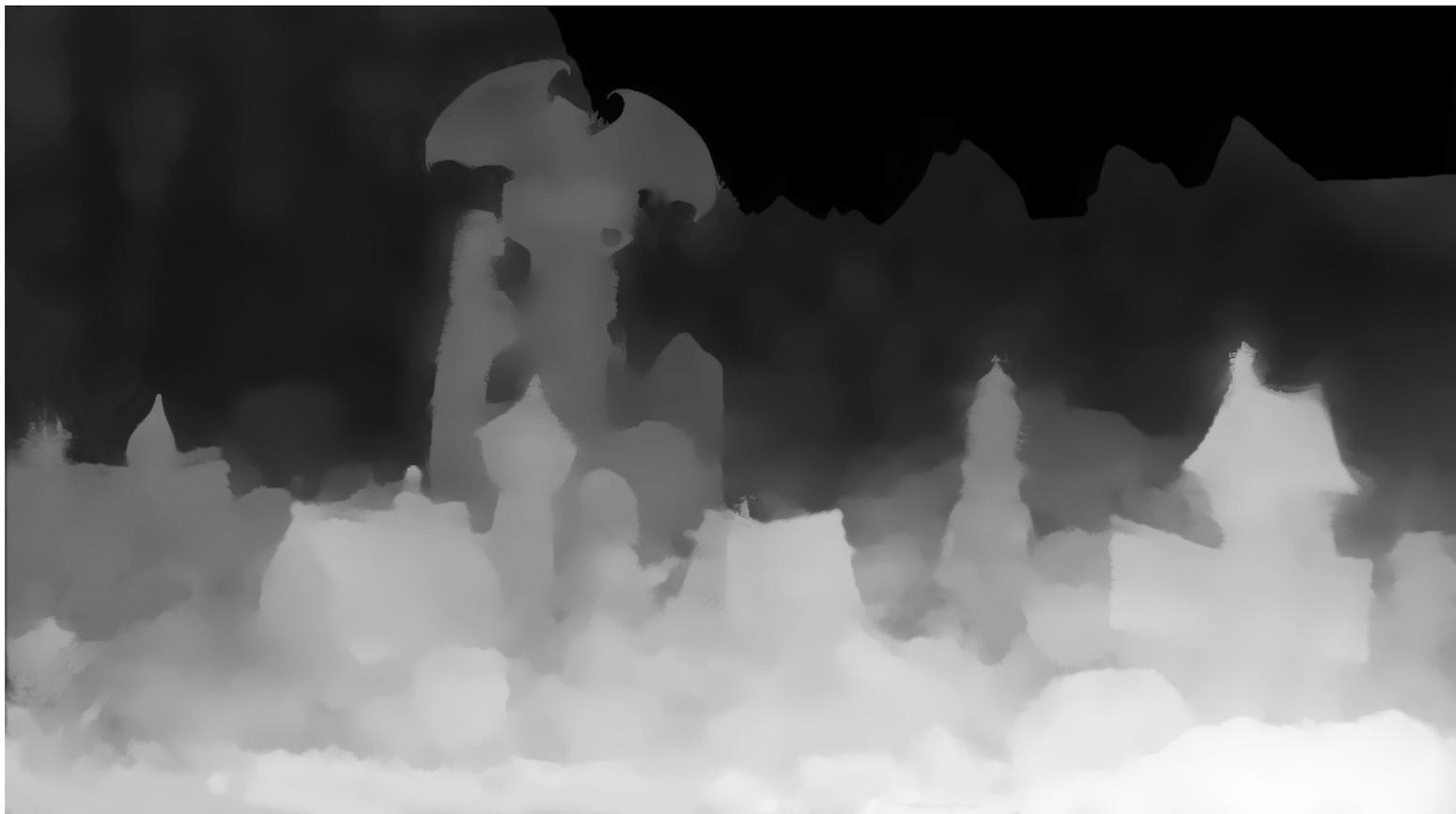
Результат (Исходный кадр)



Чем больше смещение точки между кадрами, тем ближе она к зрителю

Depth From Motion

Результат (Карта глубины)



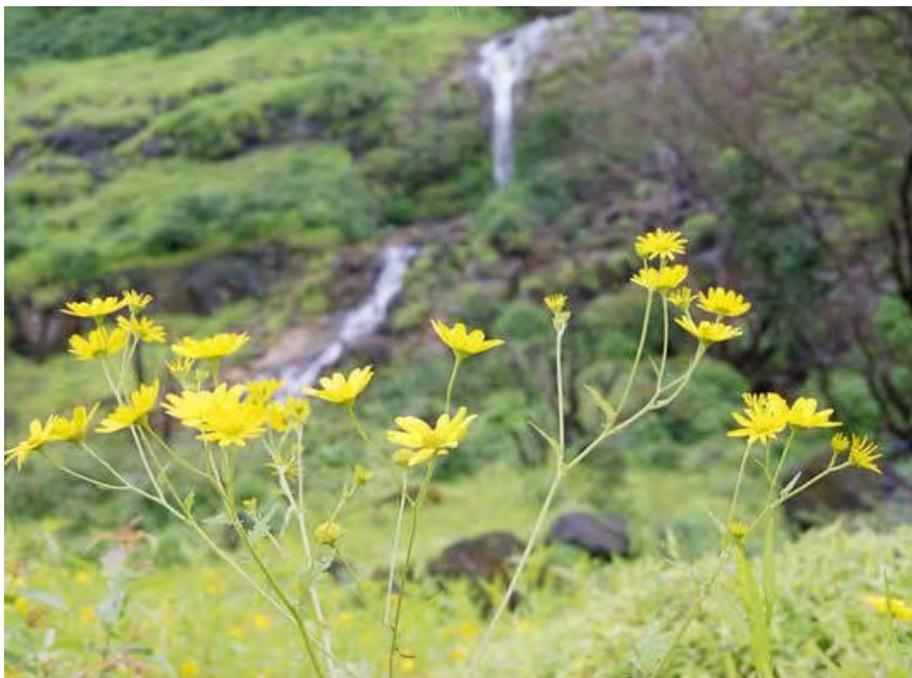
Чем больше смещение точки между кадрами, тем ближе она к зрителю

Depth From Motion

Результаты

**Processing results
in sequence with plane-parallel
camera motion**

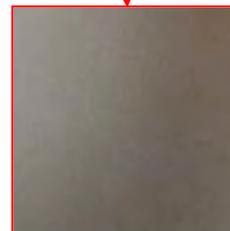
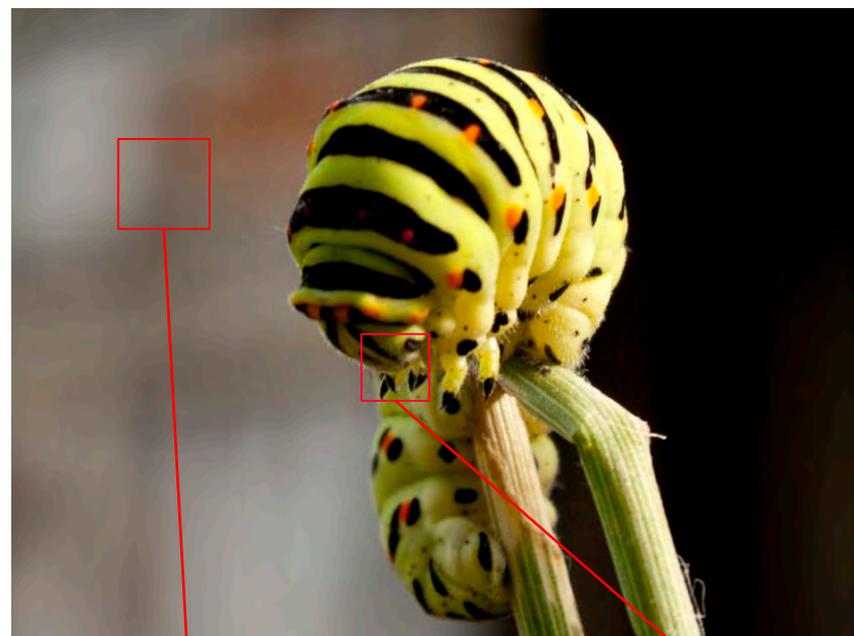
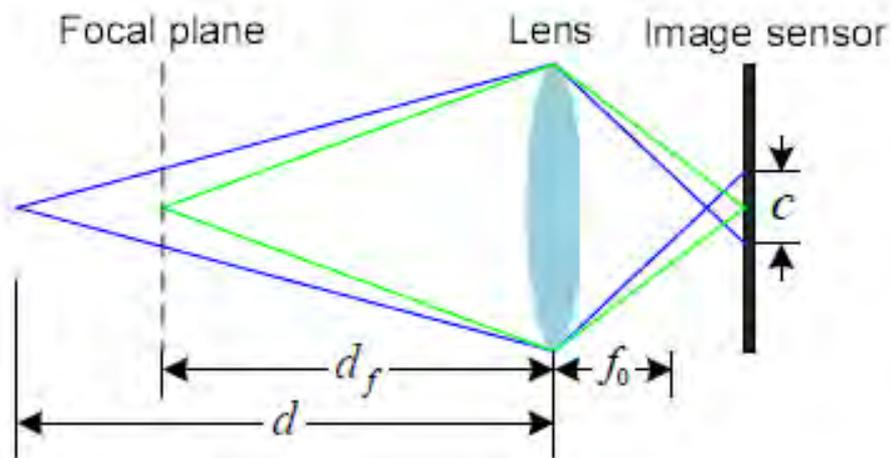
Depth from Focus



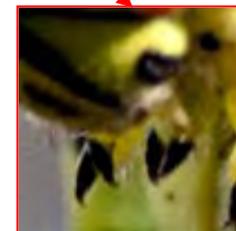
Автоматически построенная карта глубины для исходного кадра

Главная идея

Расфокусированное изображение



Размыто (дальше)

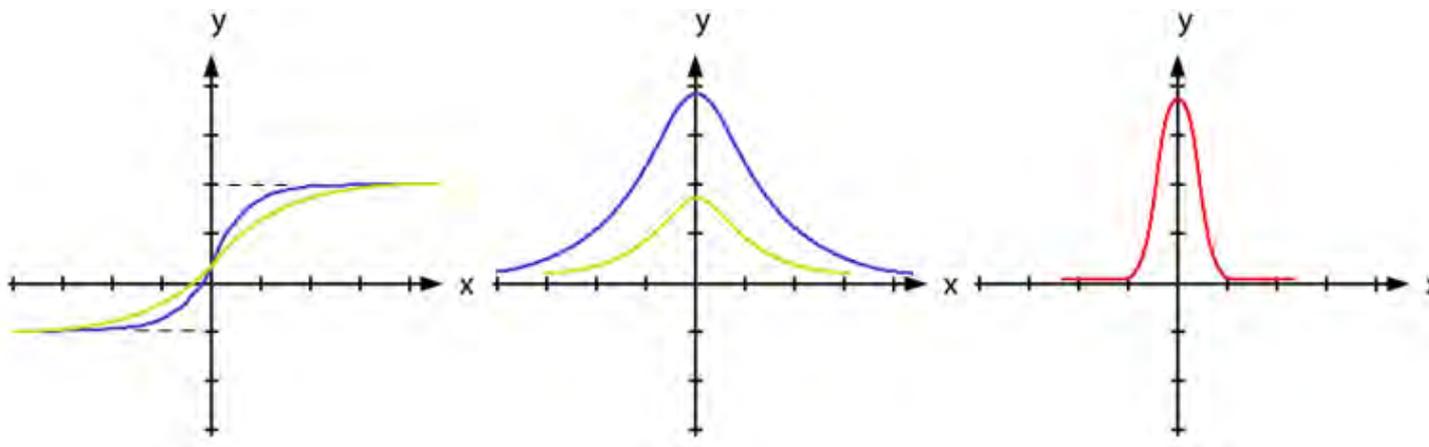


Сфокусировано (ближе)

Depth from Focus

Первичные данные

Изначальные данные о расфокусированности изображения получаем из анализа границ объектов



Исходная и размытая границы объекта

Производные функций границ

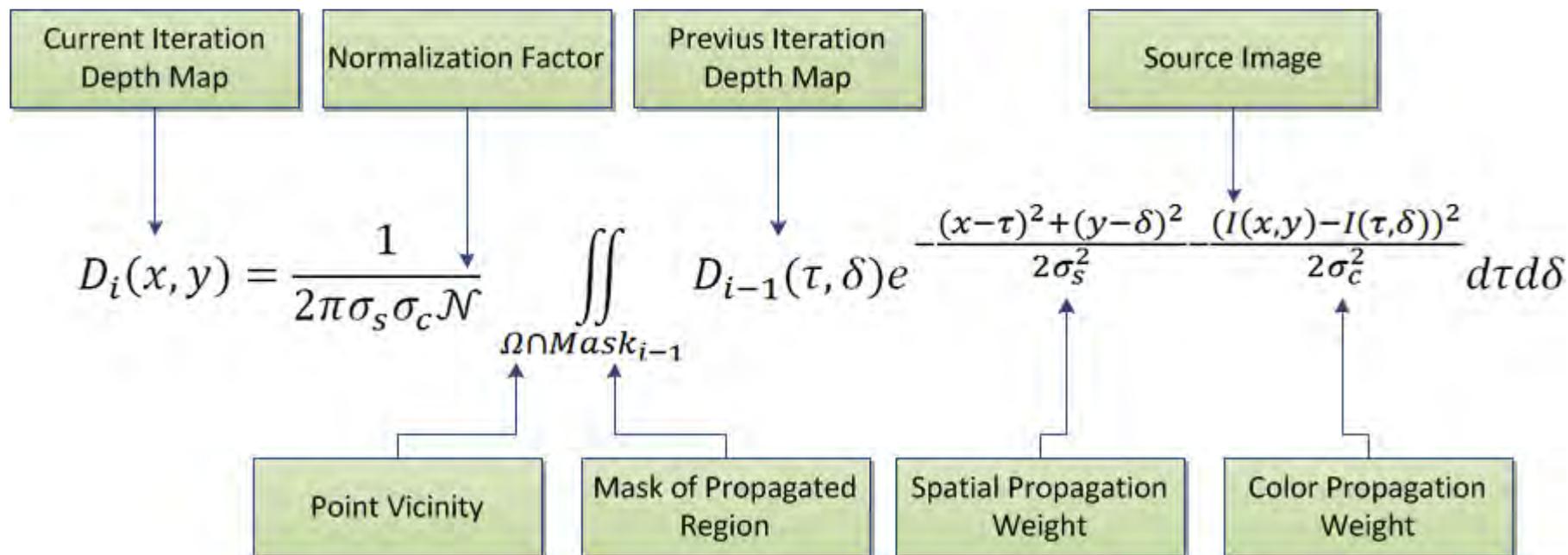
Соотношение производных функций границ

- исходная граница объекта
- Размытая граница объекта
- Соотношение производных границ

Depth from Focus

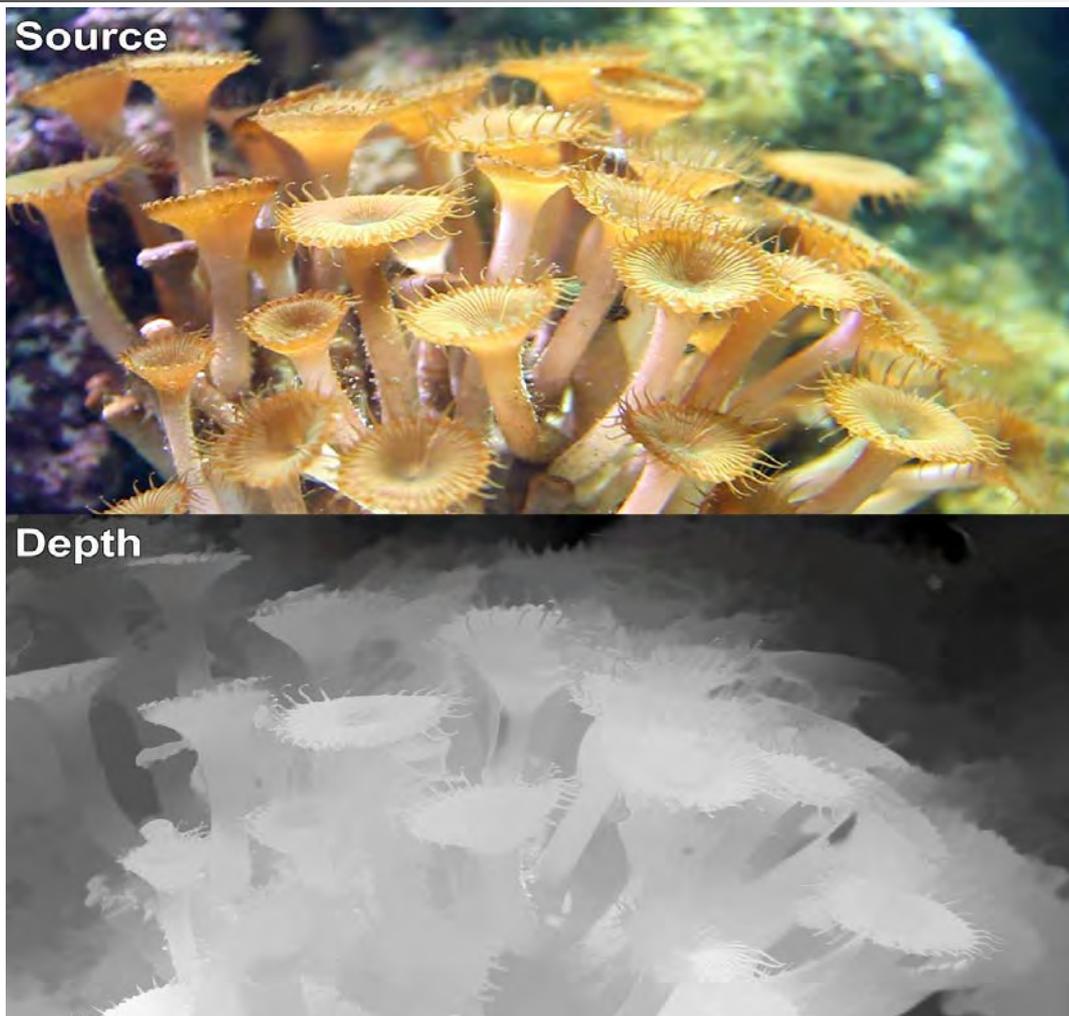
Распространение информации

Для всего кадра информация итеративно распространяется по принципу:



Depth from Focus

Результаты (1)



Depth from Focus

Результаты (2)



Исходное изображение

Depth from Focus

Результаты (3)



Карта глубины

Depth from Focus

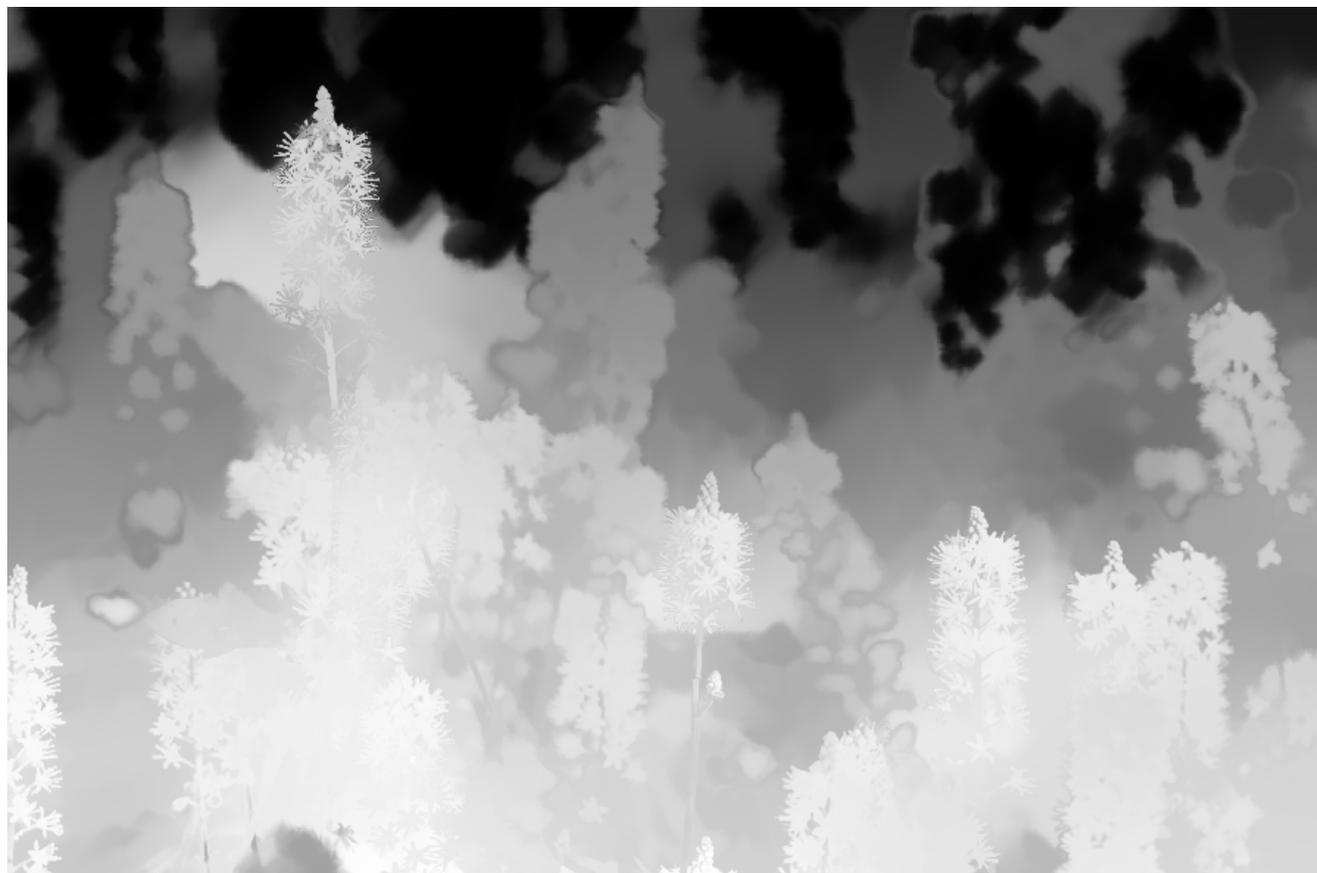
Результаты (4)



Исходное изображение

Depth from Focus

Результаты (5)



Карта глубины

Depth from Focus

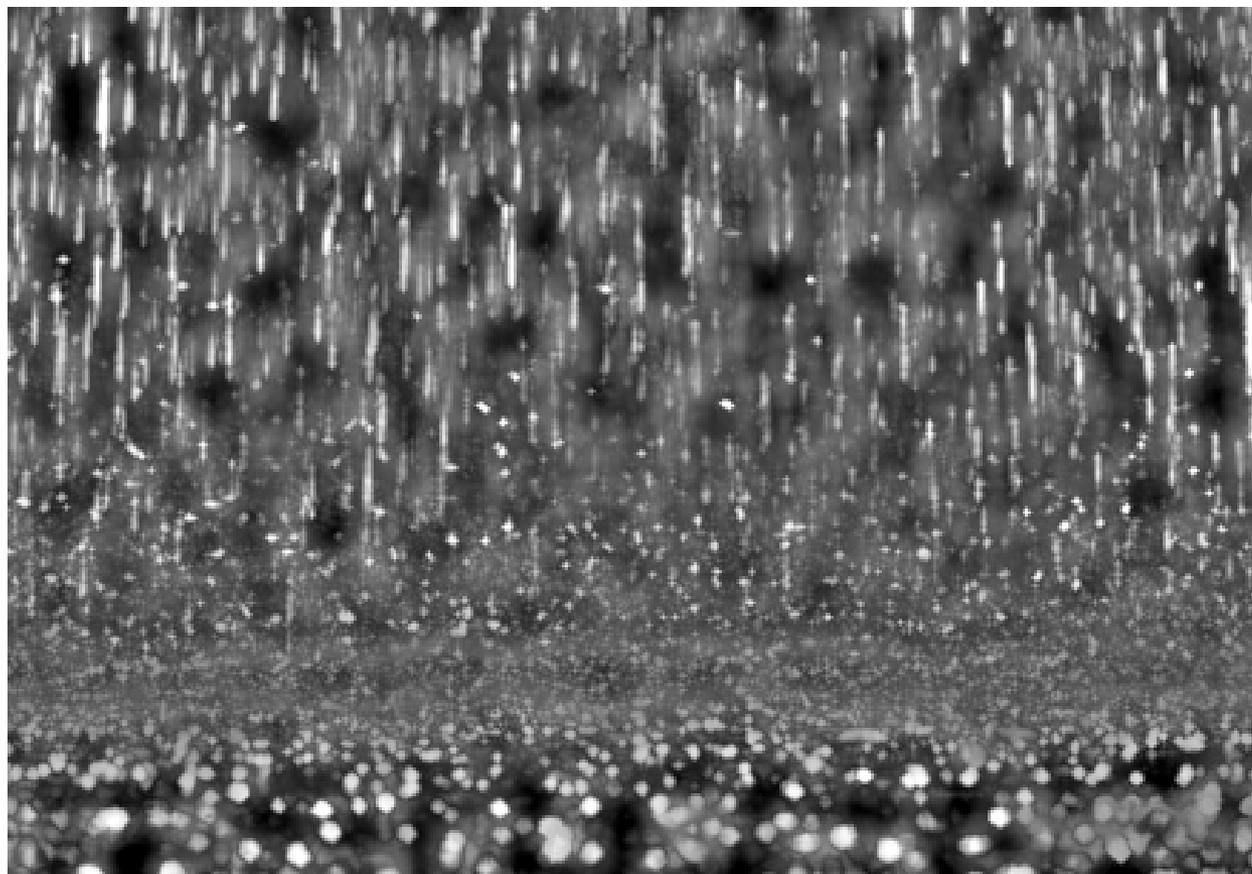
Результаты (6)



Исходное изображение

Depth from Focus

Результаты (7)



Карта глубины

Depth from Focus

Результаты (8)



Исходное изображение

Depth from Focus

Результаты (9)



Автоматически построенная маска снега

Depth from Focus

Результаты (10)



Результат в объеме

Depth from Geometry

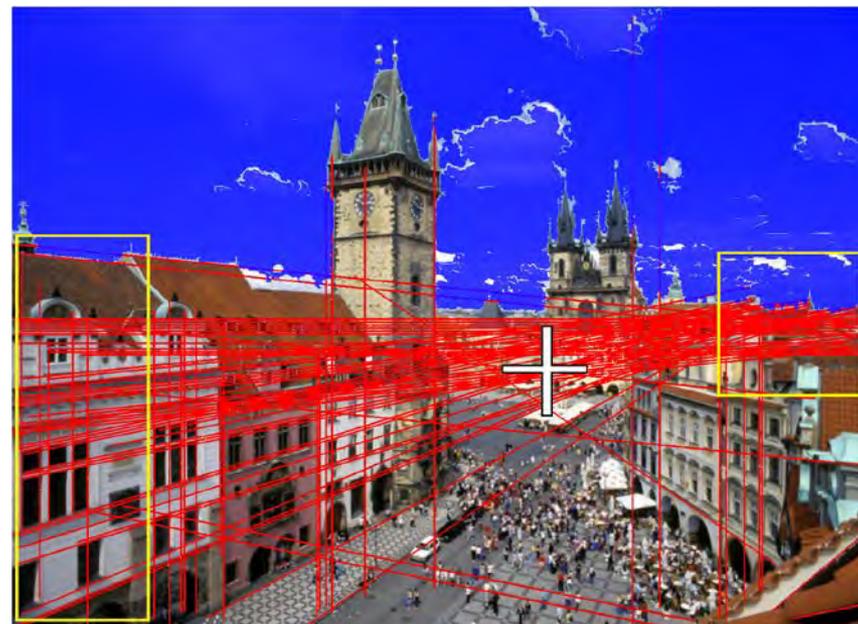


Автоматически построенная карта глубины для исходного кадра

Depth from Geometry

Анализ геометрии сцены

- Сегментация неба
- Обнаружение прямых линий на изображении
- Определение характера распределения прямых на изображении



Depth from Geometry

Результаты (1)



Исходное изображение

Depth from Geometry

Результаты (2)



Карта глубины

Depth from Geometry

Результаты (3)



Исходное изображение

Depth from Geometry

Результаты (4)



Карта глубины

Stereo to Multiview

Задача – для исходной стереопары вычислить карту глубины для дальнейшего построения большего количества ракурсов



Автоматически построенная карта глубины для исходного кадра

Optical Flow

Общая задача

Для двух последовательных кадров:

$I_1(\mathbf{x})$ — первый кадр $\mathbf{x} = (x, y)^\top \in \Omega$

$I_2(\mathbf{x})$ — второй кадр $\Omega \subset \mathbb{R}^2$ — область изображения

Требуется найти векторное поле $\mathbf{u}(\mathbf{x})$,
задающее соответствие:

$$I_1(\mathbf{x} + \mathbf{u}(\mathbf{x})) = I_2(\mathbf{x}), \forall \mathbf{x} \in \Omega$$

$$\mathbf{u}(\mathbf{x}) = (u_1(\mathbf{x}), u_2(\mathbf{x}))^\top : \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

Optical Flow

Основная идея

- Приближение сигнала рядом Тейлора:

$$I(x, y, t) \approx I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t$$

$$\frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t = 0$$

$\nabla I^T \vec{V} = -I_t$ — главное уравнение Optical Flow

- Минимизация функции энергии:

$$E(\mathbf{u}) = \sum_{\mathbf{x} \in \omega} g(\mathbf{x}) \left(\nabla I(\mathbf{x})^T \mathbf{u} + I_t(\mathbf{x}) \right)^2$$

Пример Optical Flow

Кадр 1



Пример Optical Flow

Кадр 2



Пример Optical Flow

Поле оптического потока



Тон и насыщенность цвета определяют направление и длину вектора движения точки

Stereo to Multiview

Результаты (1)



Исходный кадр левый ракурс

Stereo to Multiview

Результаты (2)



Исходный кадр правый ракурс

Stereo to Multiview

Результаты (1)



Optical Flow

Stereo to Multiview

Результаты (1)



Карта глубины

Содержание

- Современный 3D-контент
 - Многокамерные системы
 - Matrix Bullet Time
 - Toshiba Timesculpture
- **Создание 3D-контента**
 - Создание карт глубины
 - **Методы обработки карт глубины**
 - Построение новых ракурсов

Карты глубин

Фильтрация

Даже при ручной разметке кадров возникают неточности карты глубины:

- «Картонки» (отсутствие рельефа и объема)
- Неправильные границы
- Несогласованность глубины соседних кадров



Пространственная Билатеральная фильтрация (1)



+

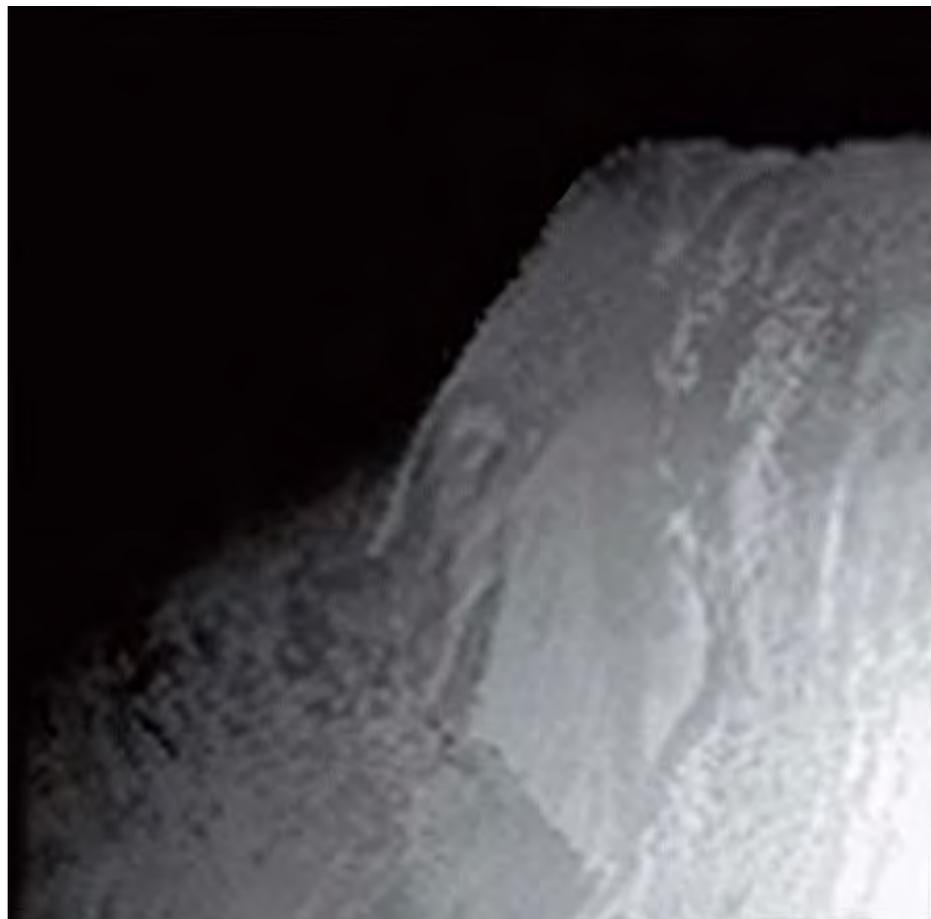
=



фильтрация карты глубины на основе исходного кадра

Пространственная Билатеральная фильтрация (2)

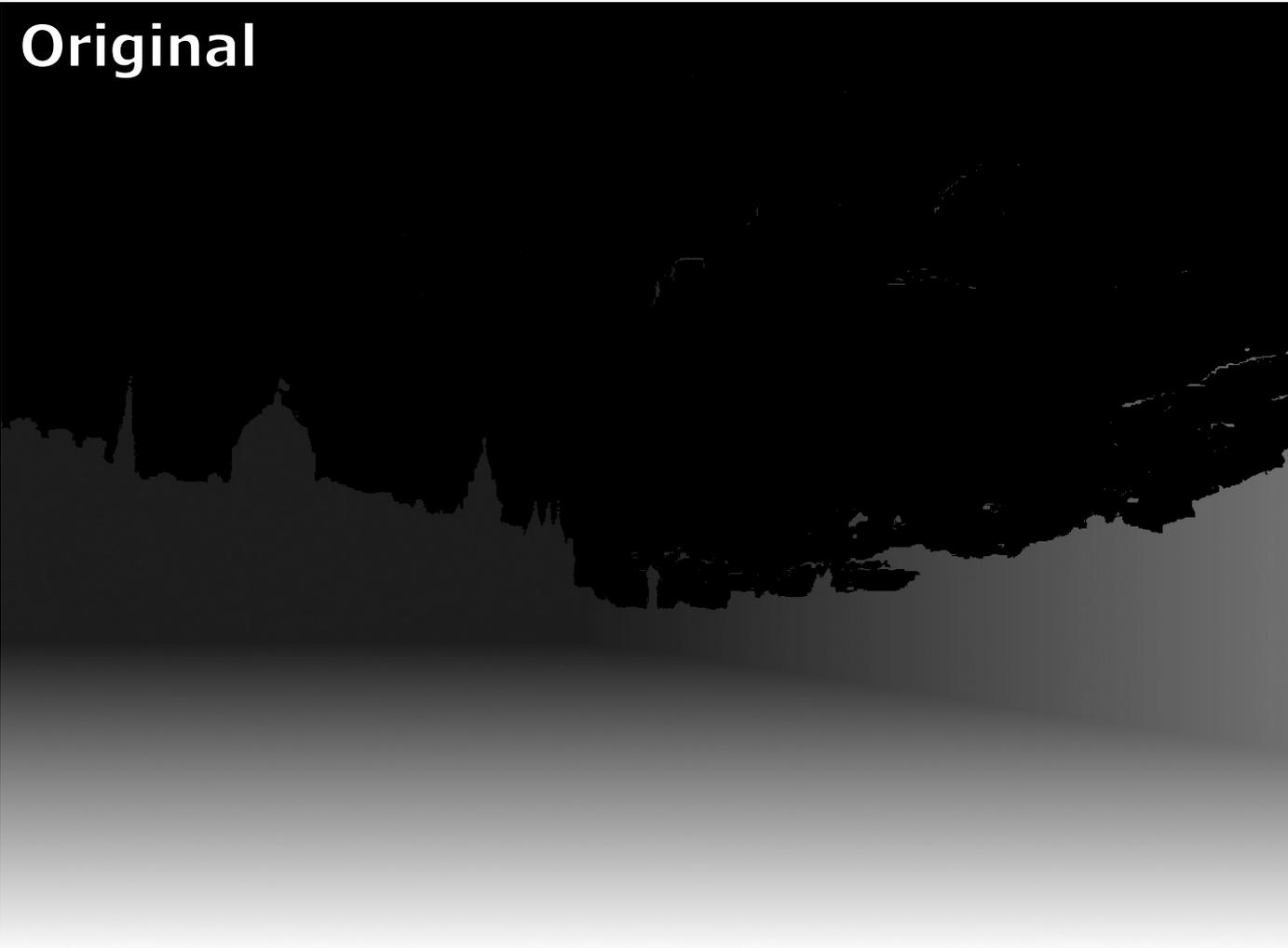
- Сглаживание глубины ровных по цвету областей
- Подчеркивание разницы на цветовых границах



Пространственная

Результаты

Original



Пространственная

Результаты

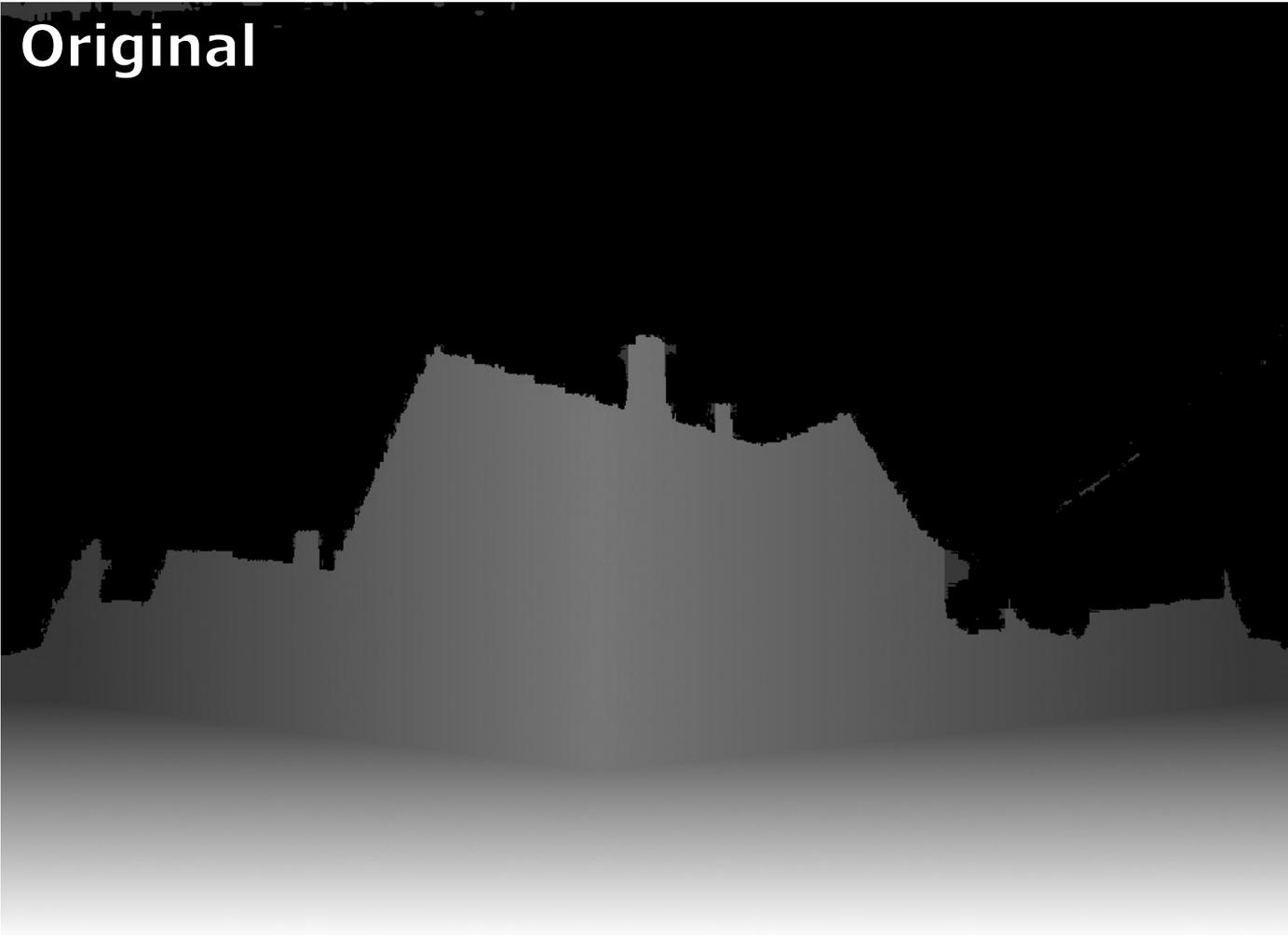
Filtered



Пространственная

Результаты

Original



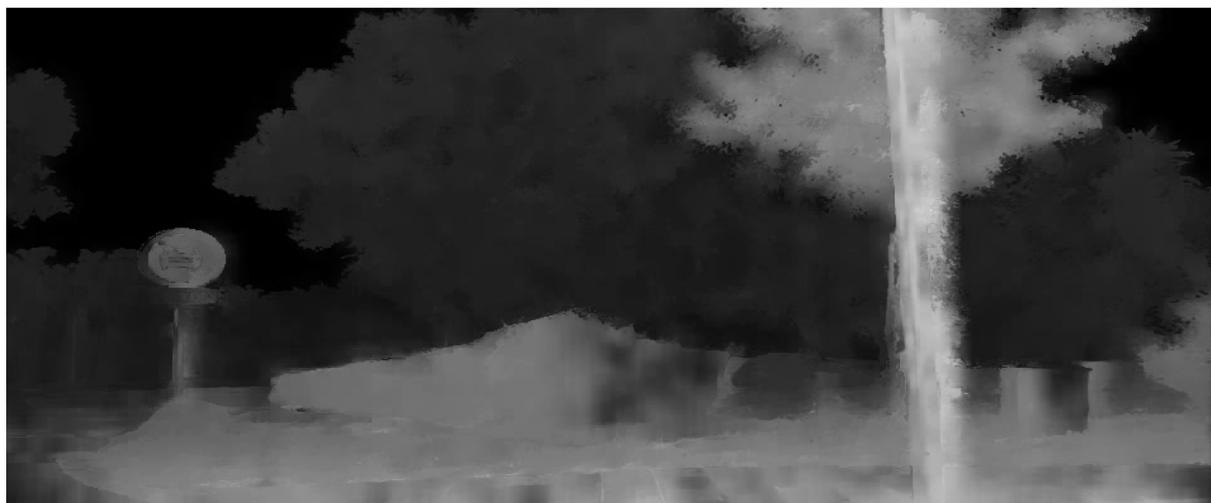
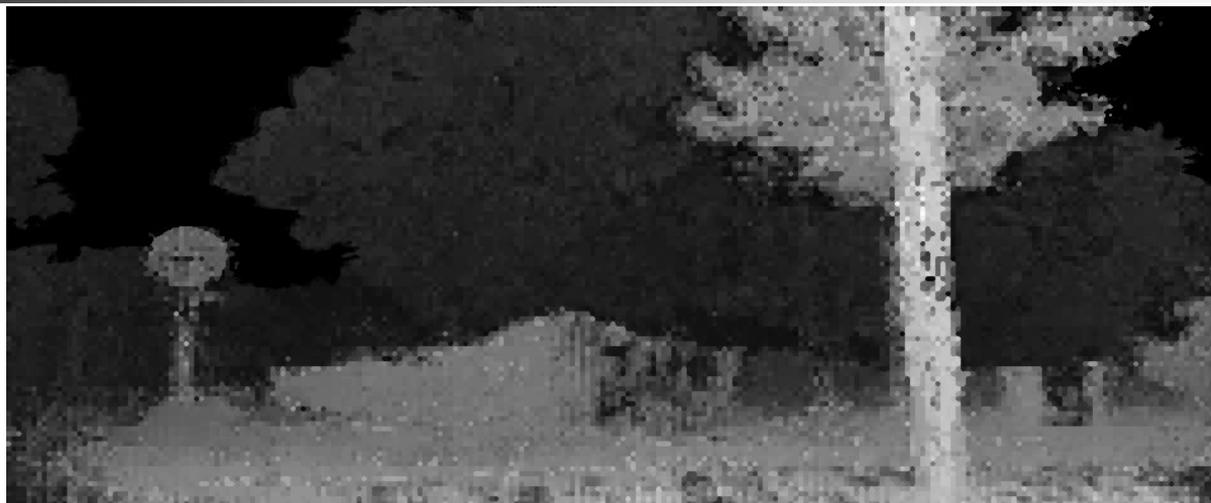
Пространственная

Результаты

Filtered



Пространственная Результаты



Пространственная Результаты



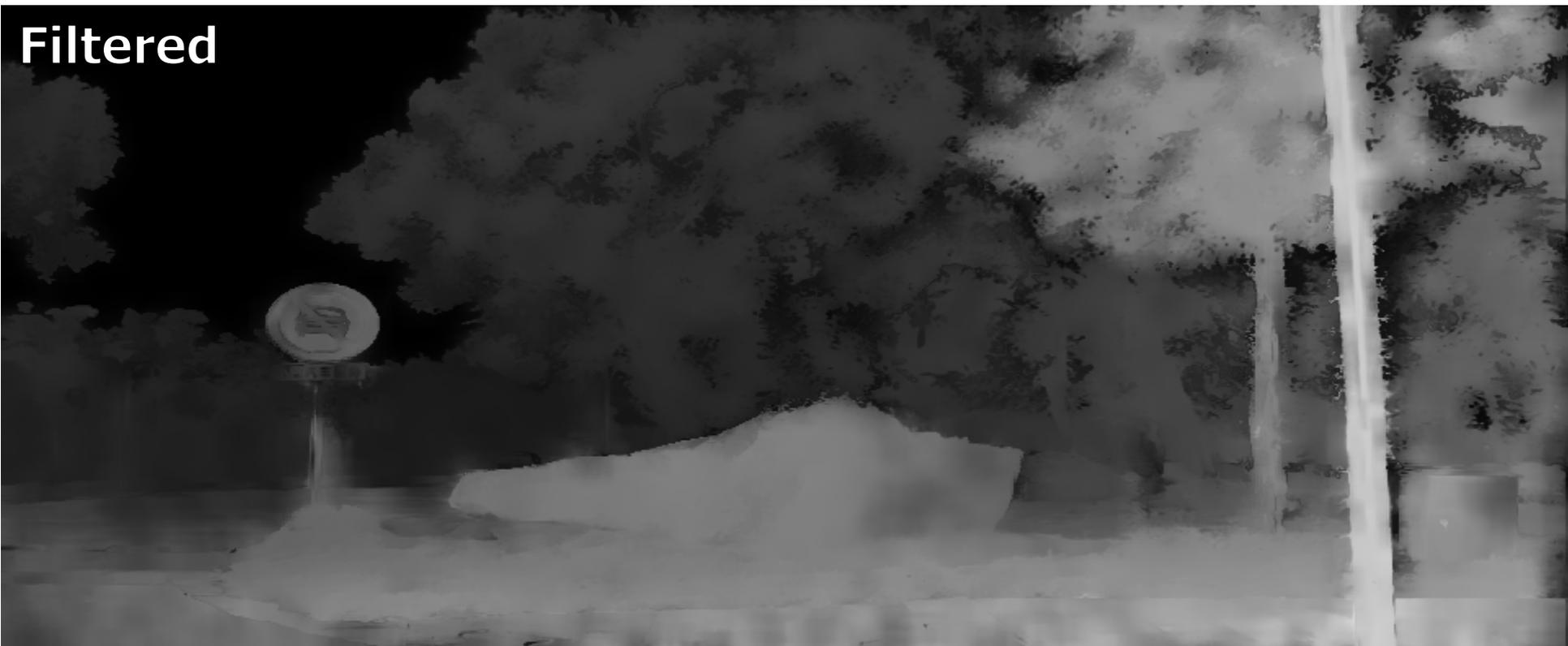
Original



Пространственная Результаты



Filtered

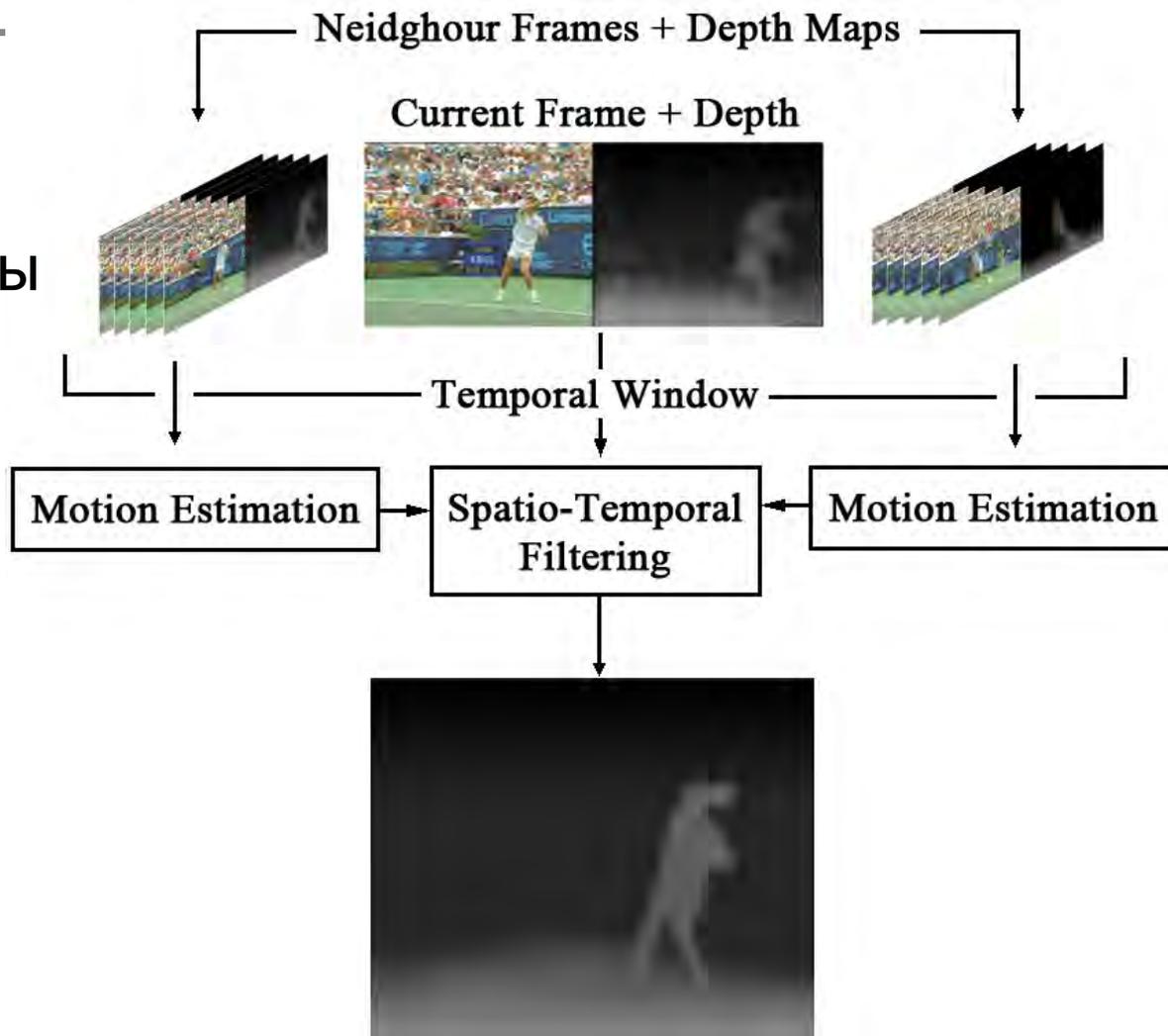


Временная фильтрация

Схема работы

Сглаживание
изменения глубины
во времени

- Нормализация, устранение «мерцания»
- Удаление «артефактов»



Временная фильтрация

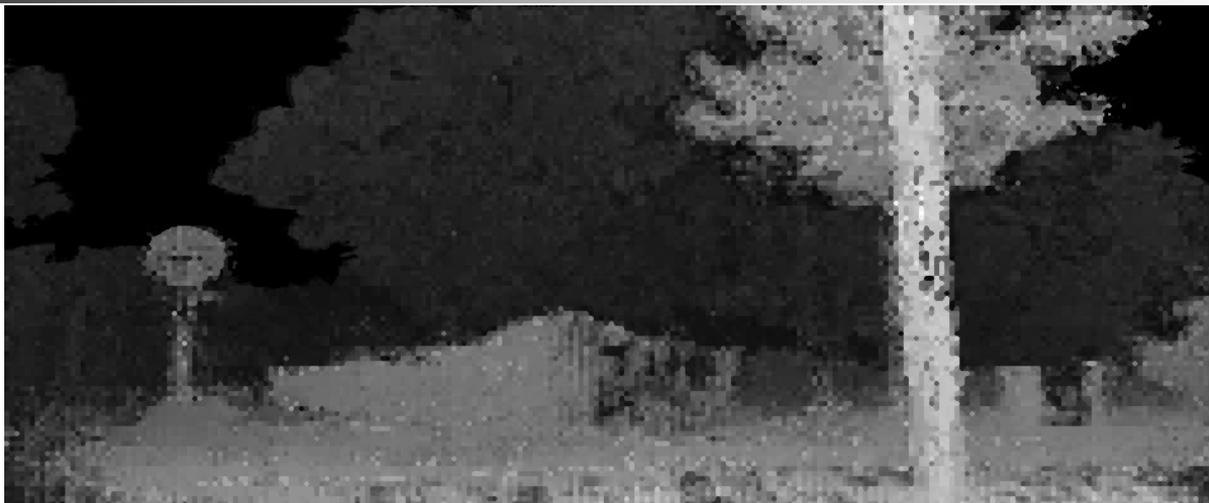
Spatio-Temporal Filtering

- Пиксель определяется тремя координатами – (x, y, t) положение в пространстве и времени
- Рассматриваем окрестность по всем трем координатам, с учетом компенсации движения

$$D(x_0, y_0, t_0) = \frac{1}{\sqrt{(2\pi)^3 \sigma_s \sigma_c \sigma_t}} * \\ * \sum_{(x_1, y_1, t_1) \in \Omega} D(x_1, y_1, t_1) * \\ * e^{-\frac{(x_0 - x_1)^2 + (y_0 - y_1)^2}{2\sigma_s^2} - \frac{(I(x_0, y_0, t_0) - I(x_1, y_1, t_0))^2}{2\sigma_c^2} - \frac{(t_0 - t_1)^2}{2\sigma_t^2}}$$

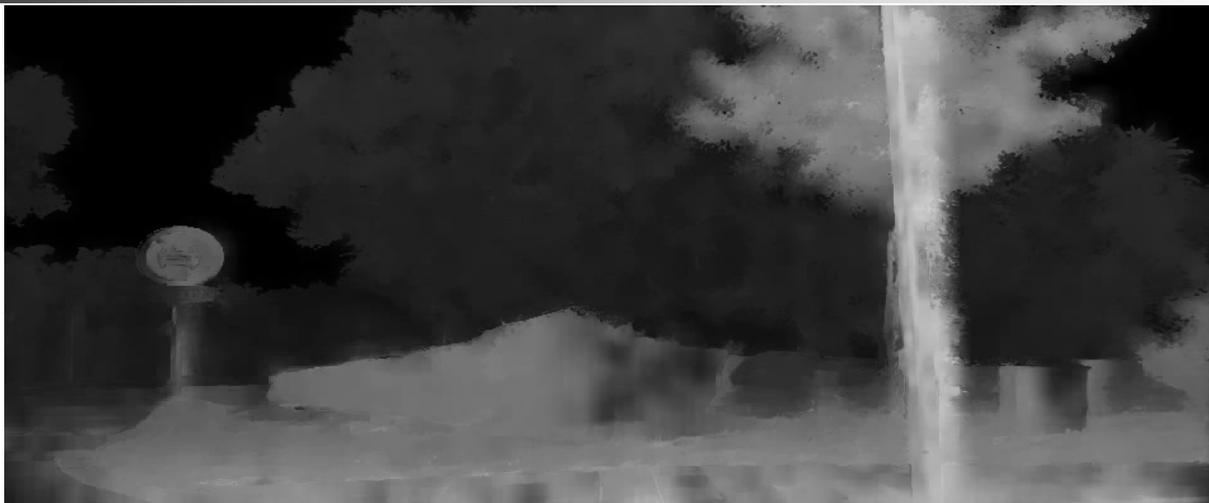
Временная фильтрация

Результаты



Временная фильтрация

Сравнение



Временная фильтрация

Сравнение



Original

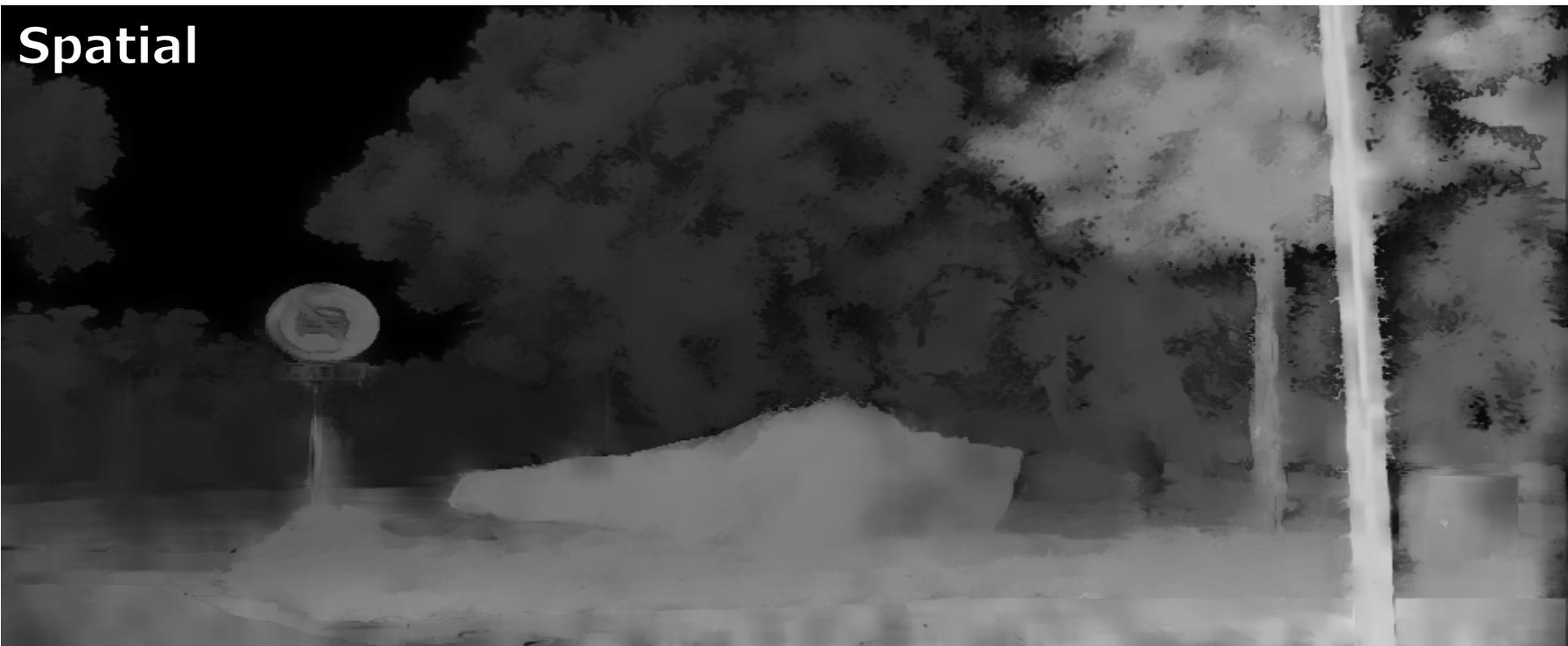


Временная фильтрация

Сравнение



Spatial



Временная фильтрация

Сравнение



Spatio-temporal



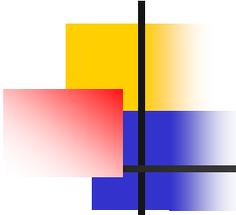
Временная фильтрация

Сравнение



Original





Содержание

- **Современный 3D-контент**
 - Многокамерные системы
 - Matrix Bullet Time
 - Toshiba Timesculpture
- **Создание 3D-контента**
 - Создание карт глубины
 - Методы обработки карт глубины
 - **Построение новых ракурсов**

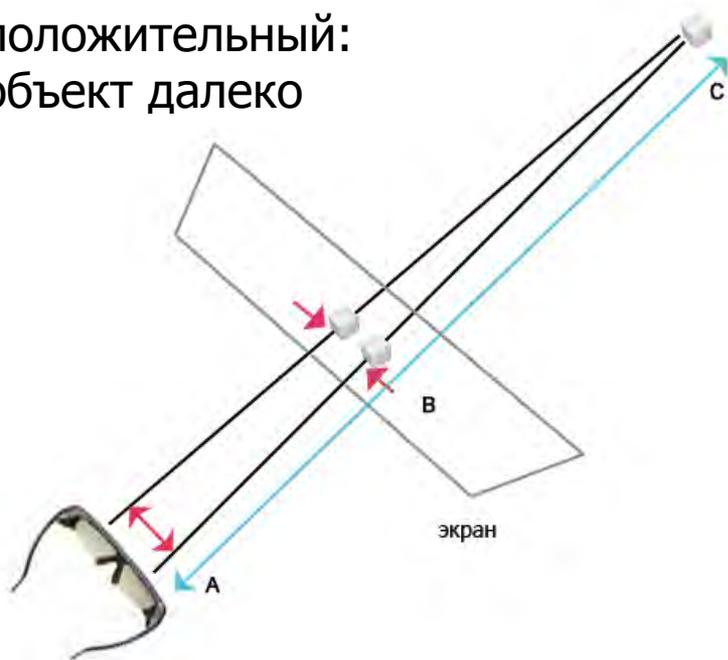
Построение новых ракурсов

- На входе имеется исходное изображение и карта глубины
- Требуется преобразовать исходный кадр согласно принципам бинокулярной стереографии (объектам, находящимся ближе, соответствует больший параллакс)

Бинокулярная стереография

Параллакс — расстояние между картинками для разных глаз на экране

положительный:
объект далеко



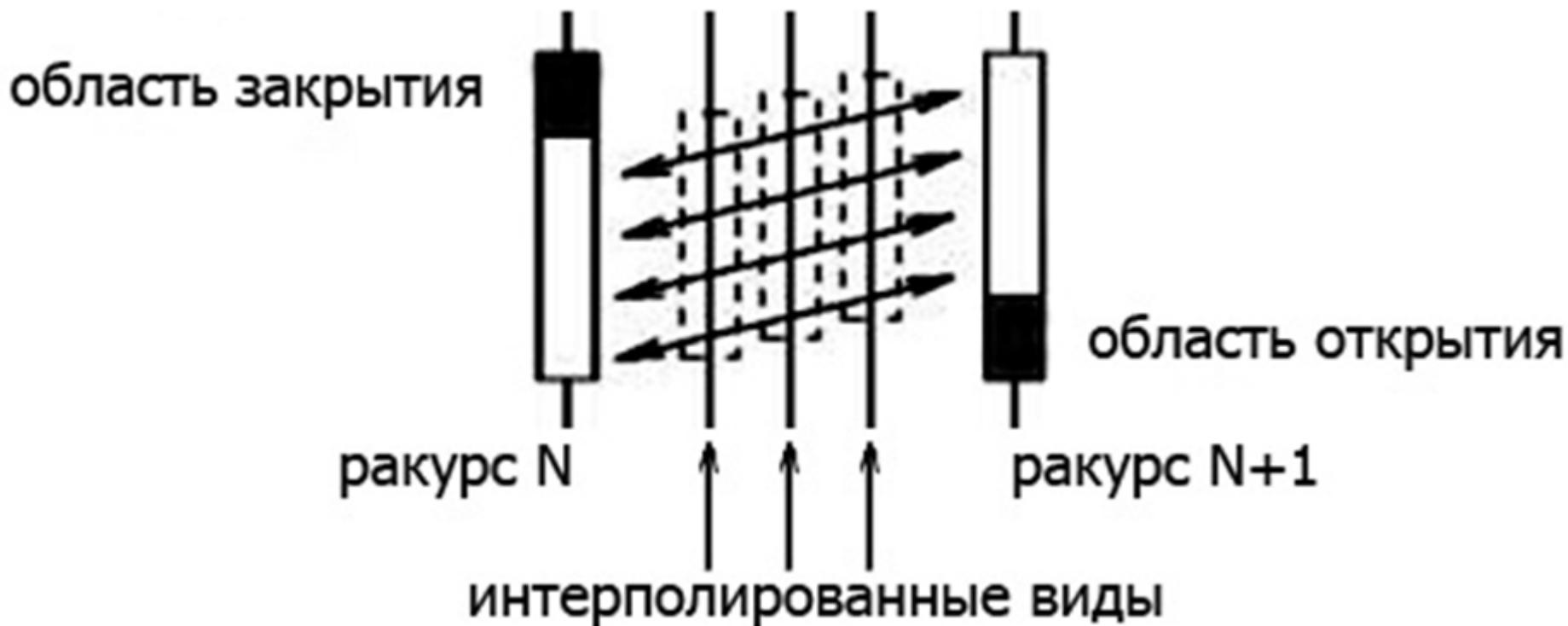
отрицательный:
объект рядом



Построение ракурсов

Интерполирование видов (1)

Смещаем пиксели в соответствии с их глубиной



Построение ракурсов

Интерполирование видов (2)



Построение новых ракурсов

Inpainting



- Появляются области, которых не было на исходном изображении (т.н. области открытия)
- Требуются алгоритмы качественного заполнения пустот (т.н. алгоритмы inpainting)

Inpainting

Основная идея (1)

Статическая экстраполяция объектов фона в области открытия:

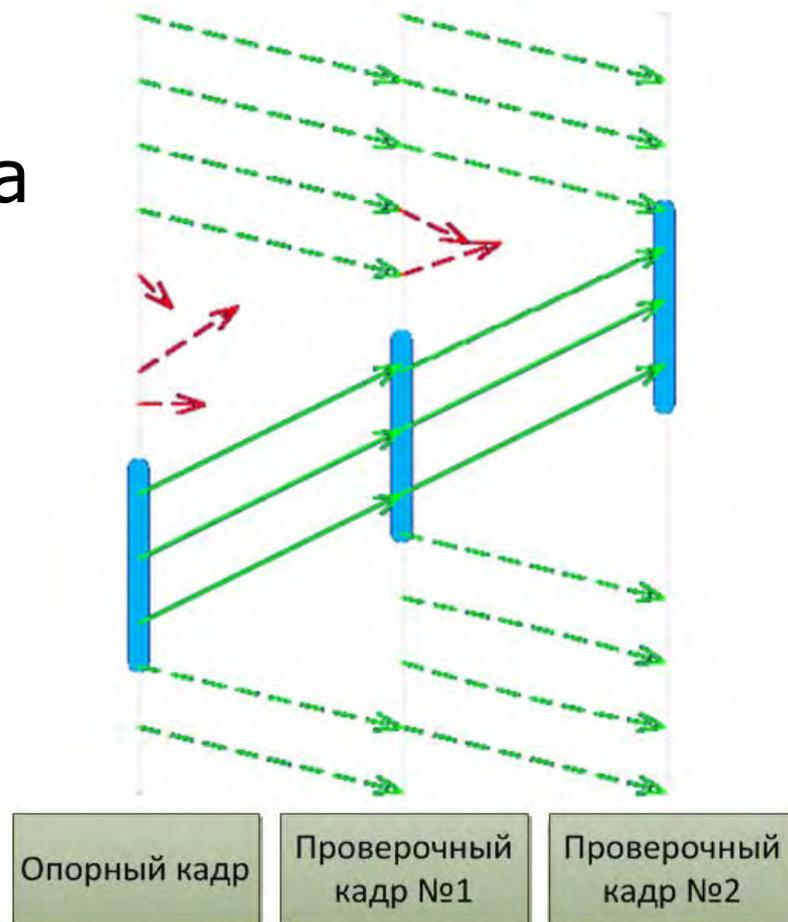
$$\forall (x, y) \in \Omega_u : \bar{I}_{x,y} = \frac{\iint_{C_{x,y} \setminus \Omega_u} W_{i,j} \bar{I}_{i,j} di dj}{\iint_{C_{x,y} \setminus \Omega_u} W_{i,j} di dj}$$

$$W_{i,j} = e^{-\frac{\|(\overline{x,y}) - \overline{(i,j)}\|^2}{\sigma_r^2}} e^{-\frac{(D_{bg,x,y} - D_{i,j})^2}{\sigma_d^2}}$$

Inpainting

Основная идея (2)

Для видео возможно
восстановить участки фона
за счет соседних кадров
с помощью анализа
результатов компенсации
движения



Inpainting

Результаты

Background reconstruction



Foreground and background motion,
complex background

Inpainting

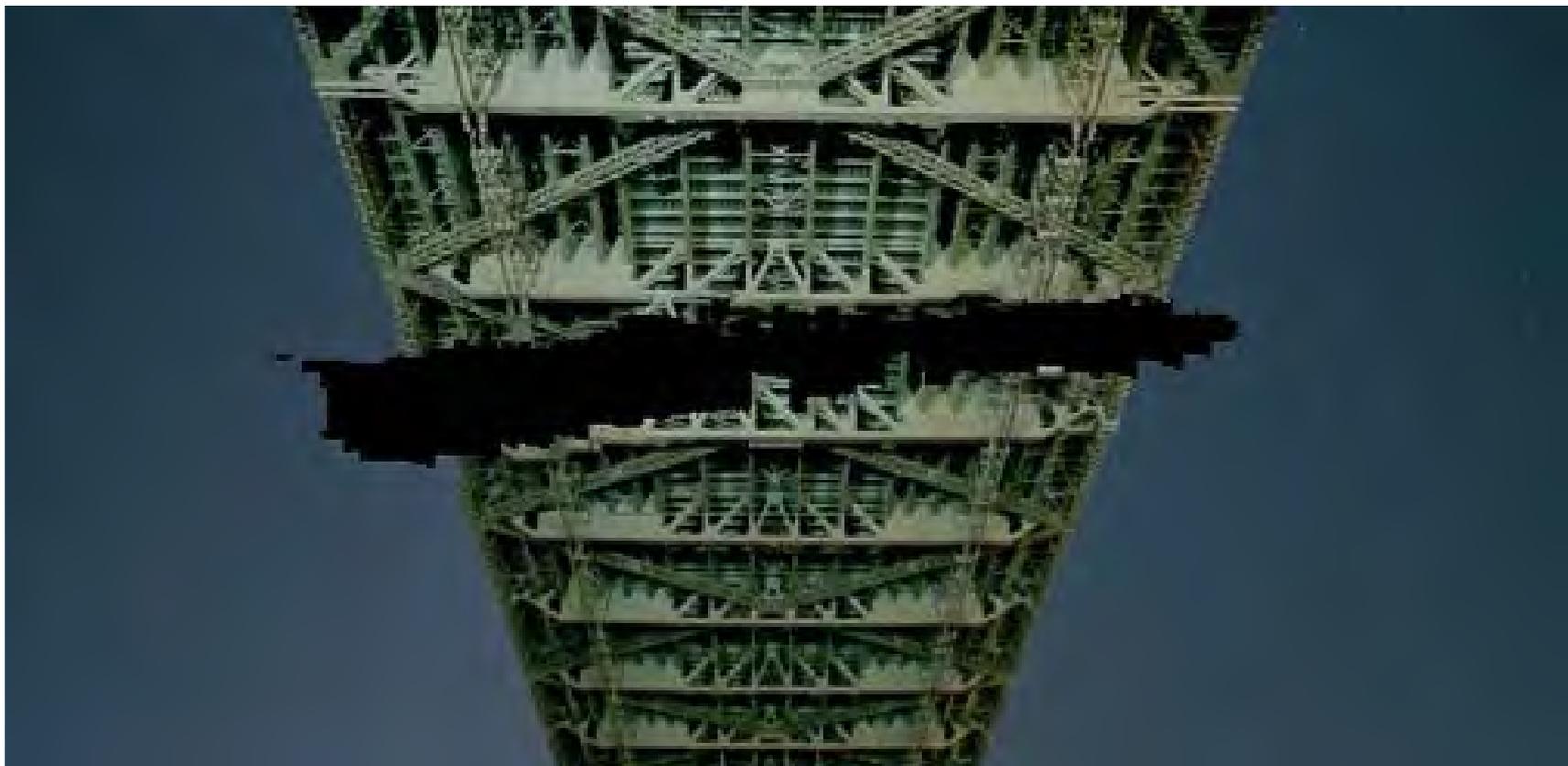
Альтернативная идея

Использование самоподобия в текстурах изображения:

- Можно восстанавливать значение точки из точек со схожей окрестностью
 - Очень долгое время работы
 - Не восстанавливаются сильно текстурированные области
- Решение – восстановление изображения целыми блоками

Exemplar-Based Inpainting

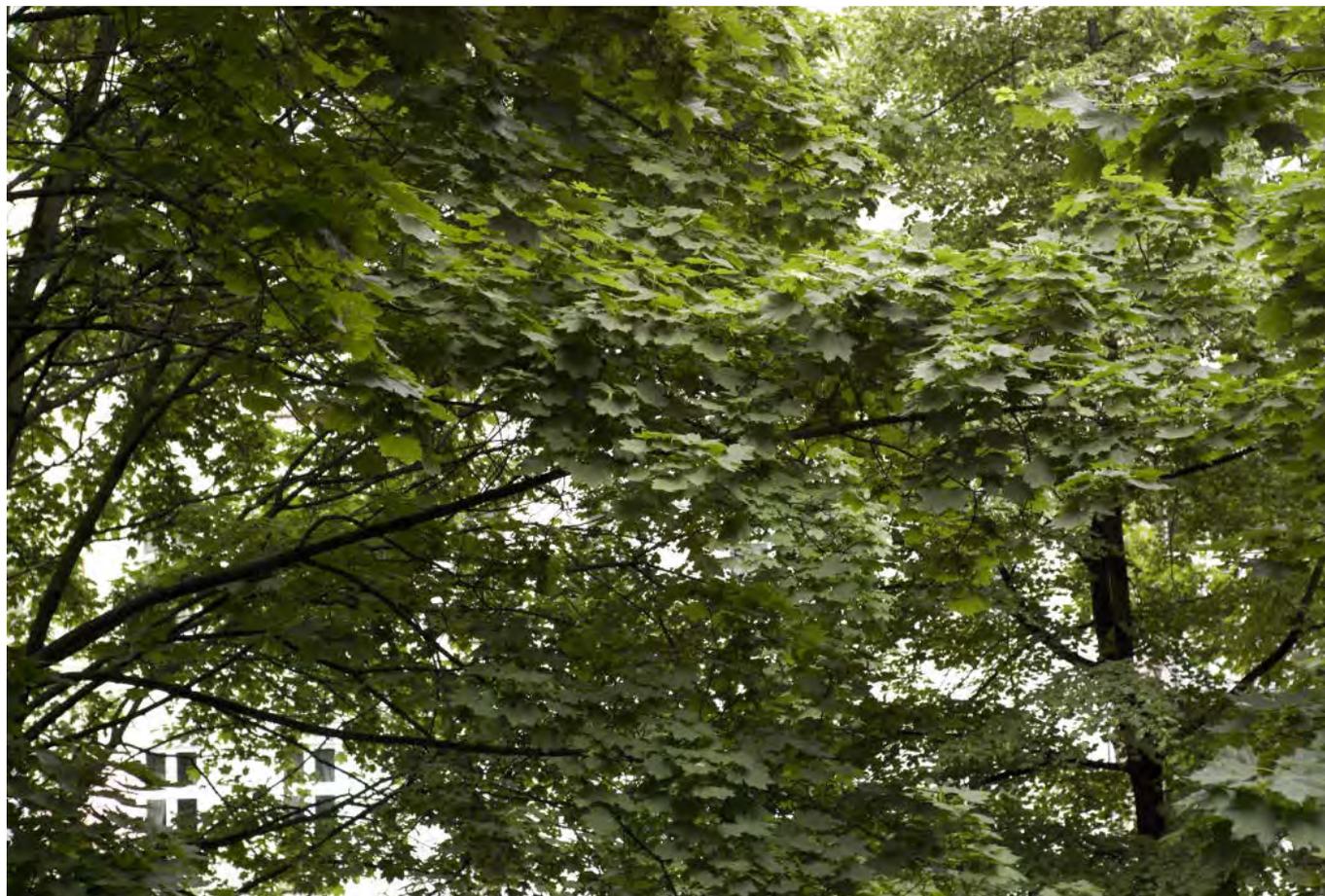
Результаты (1)



Заполнение экземплярами

Exemplar-Based Inpainting

Результаты (2)



Исходное изображение

Exemplar-Based Inpainting

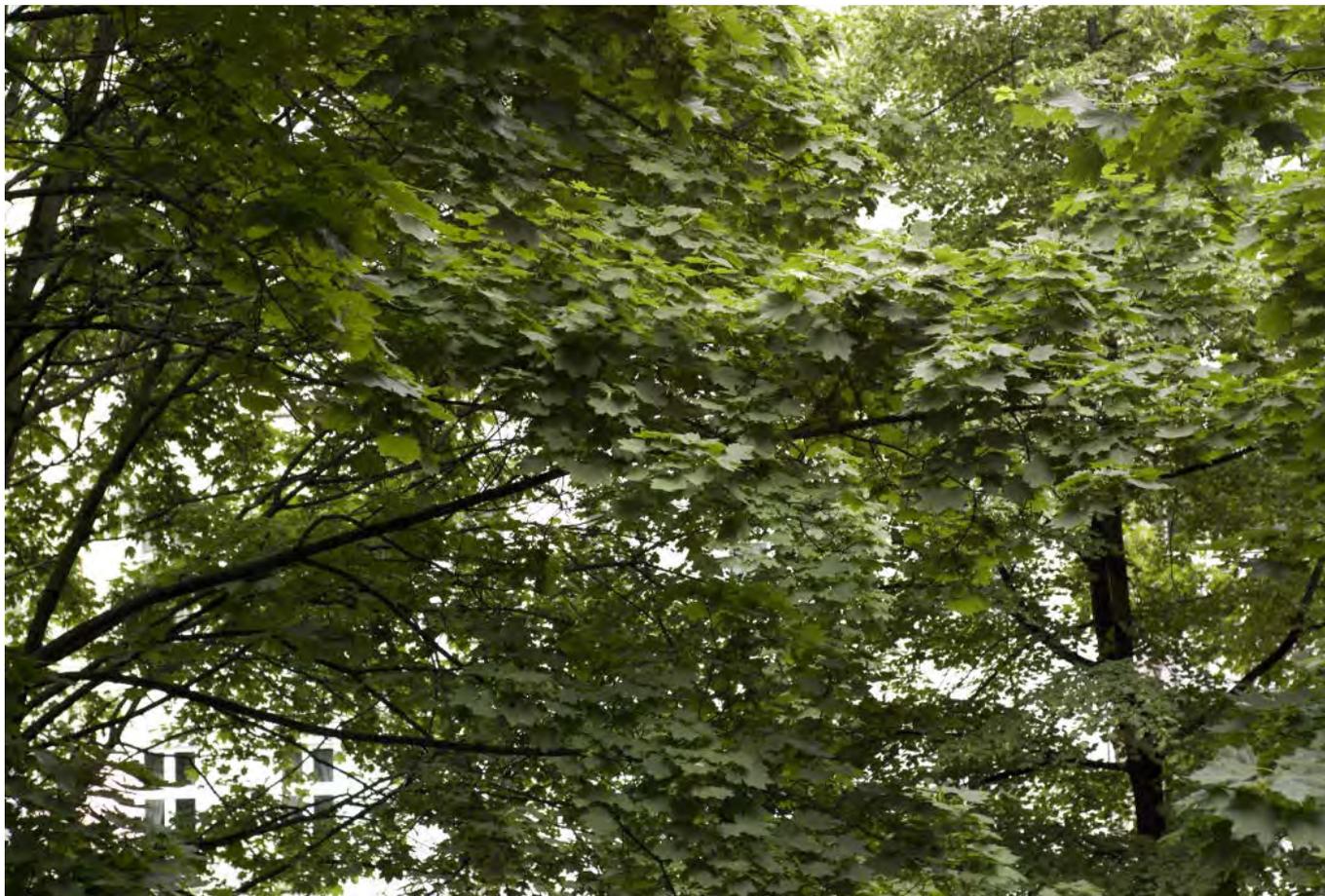
Результаты (3)



Замаскированные объекты

Exemplar-Based Inpainting

Результаты (4)



Заполнение удаленной области

Exemplar-Based Inpainting

Результаты (5)



Исходное изображение

Exemplar-Based Inpainting

Результаты (6)



Замаскированный объект

Exemplar-Based Inpainting

Результаты (7)

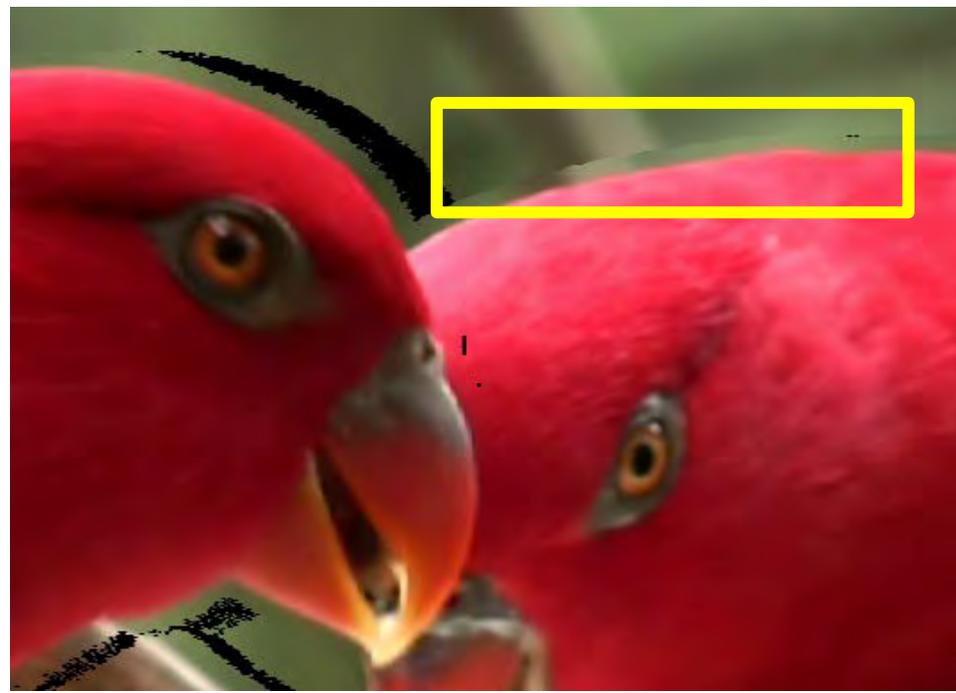
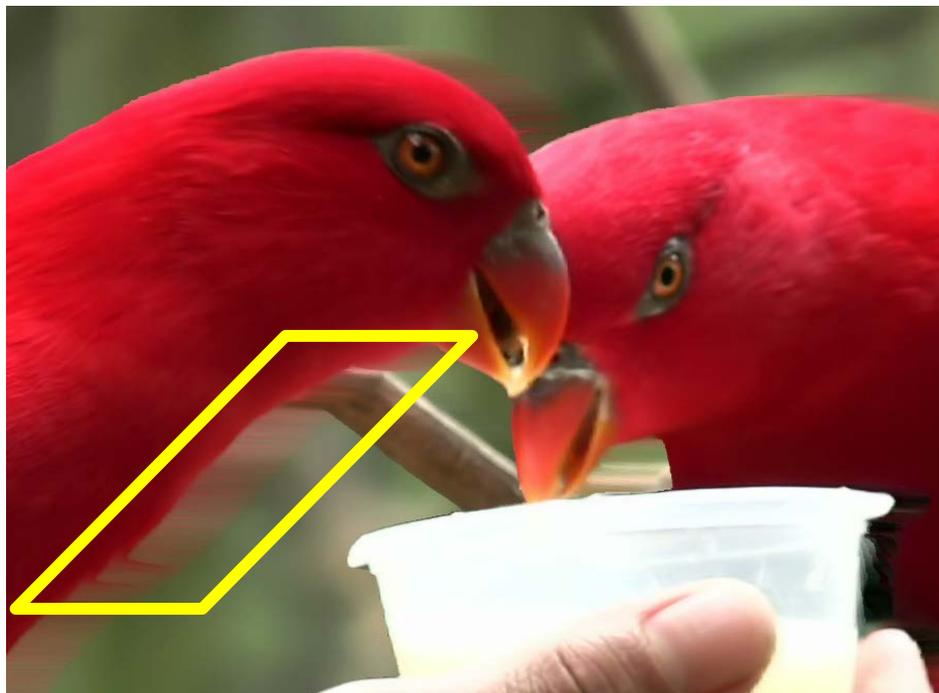


Заполнение удаленной области

Построение новых ракурсов

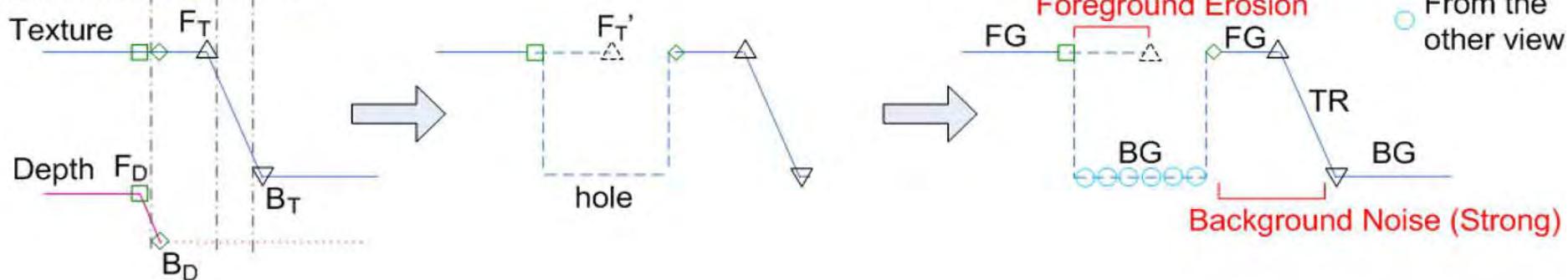
Неточности границ

Подобные проблемы на границах объектов пагубно сказываются на восприятии 3D



Возможные проблемы «Прилипание» объекта к фону (1)

Case 1: *FA FM TM BA*



FG — передний план

BG — задний план

TR — область перехода (TRansition area)

F_D , B_D , F_T , B_T — границы объектов на исходном изображении
и на карте глубины

FA , FM , TM , BA , BM — foreground/transition/background
alignment/misalignment

Возможные проблемы

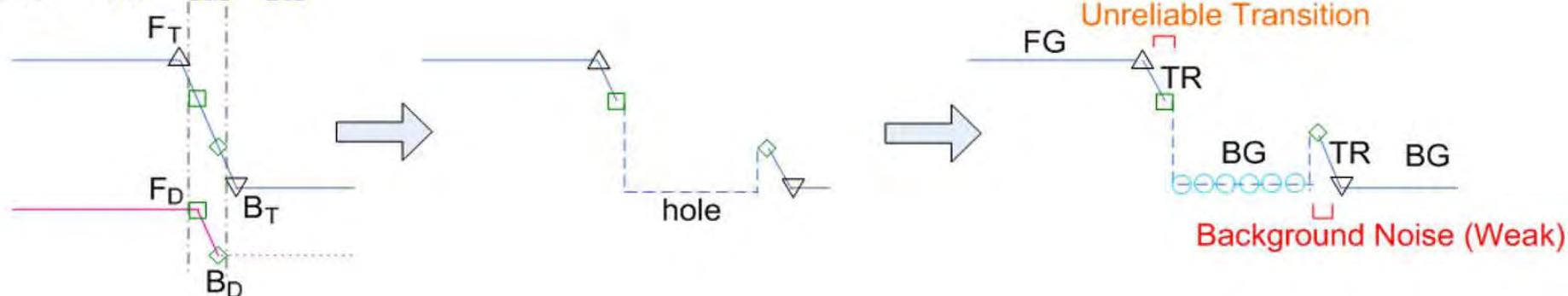
«Прилипание» объекта к фону (2)



Возможные проблемы

Разрыв размытой границы (1)

Case 2: *FA TM BA*



FG — передний план

BG — задний план

TR — область перехода (TRansition area)

F_D , B_D , F_T , B_T — границы объектов на исходном изображении и на карте глубины

FA , FM , TM , BA , BM — foreground/transition/background alignment/misalignment

Возможные проблемы

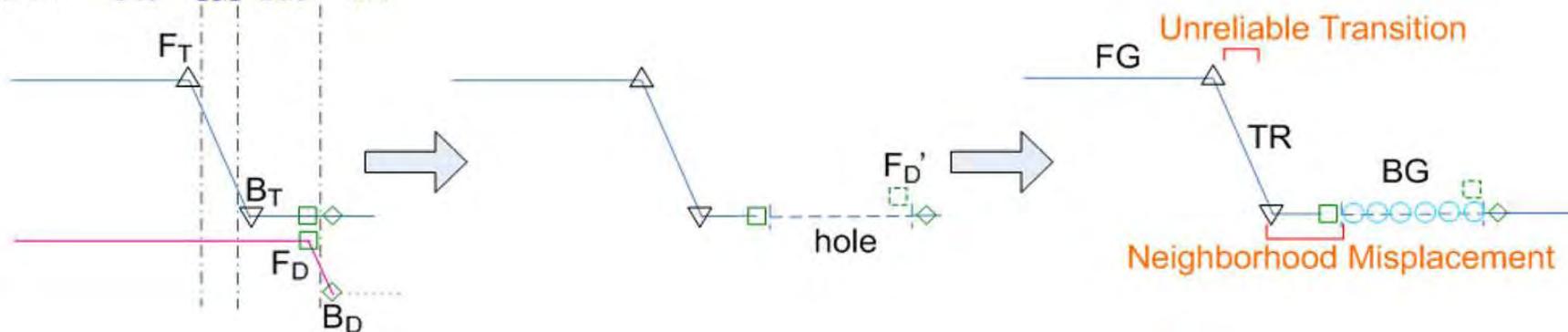
Разрыв размытой границы (2)



Возможные проблемы

Эффект «стеклянного колпака» (1)

Case 3: *FA TM BM BA*



FG — передний план

BG — задний план

TR — область перехода (TRansition area)

F_D , B_D , F_T , B_T — границы объектов на исходном изображении и на карте глубины

FA, FM, TM, BA, BM — foreground/transition/background alignment/misalignment

Возможные проблемы

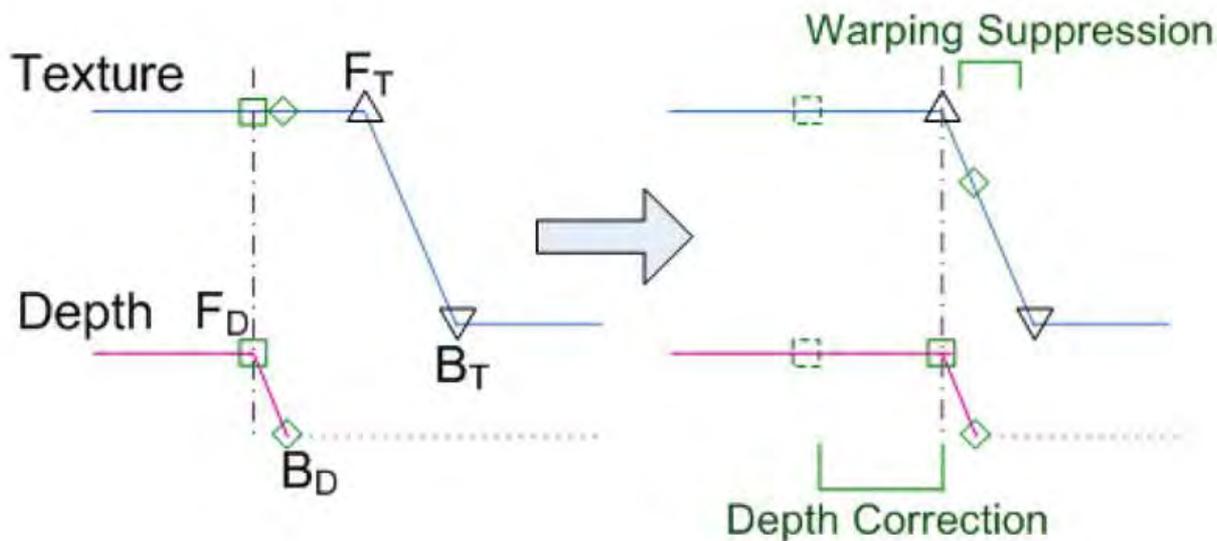
Эффект «стеклянного колпака» (2)



Варианты решения

Алгоритм удаления артефактов (1/3)

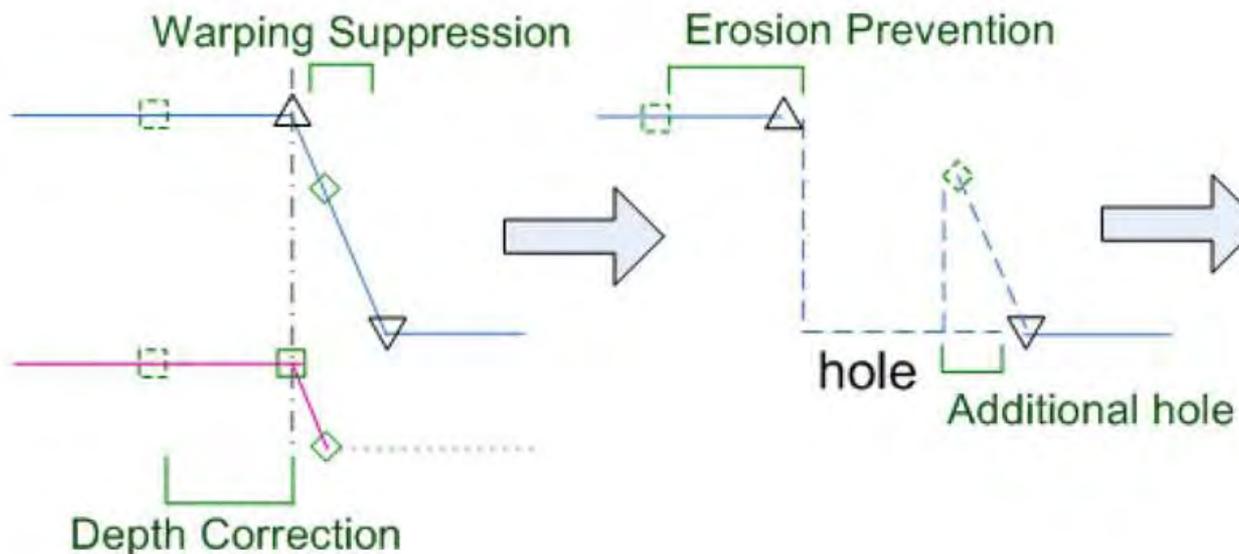
1. Выравнивание границы объекта на исходном изображении и карте глубины



Варианты решения

Алгоритм удаления артефактов (2/3)

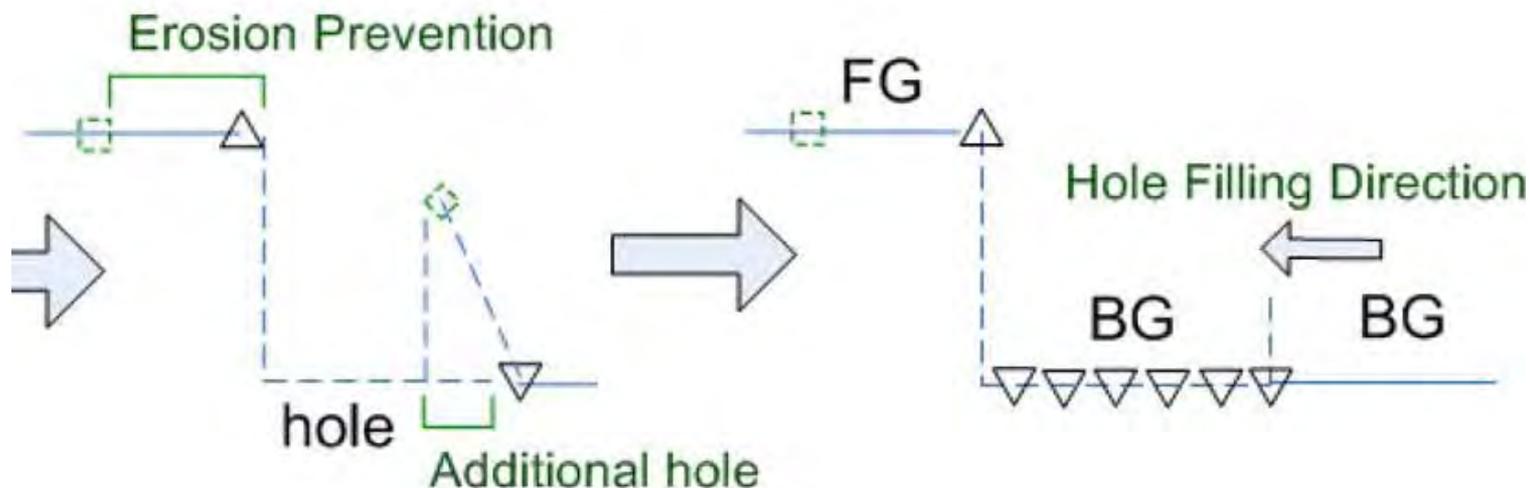
- Удаление некорректно сдвинутых пикселей объекта



Варианты решения

Алгоритм удаления артефактов (3/3)

3. Заполнение disocclusion-областей



Варианты решения

Matting

Декомпозиция исходного изображения I :

1. F – передний план
2. B – задний план
3. α – карта прозрачности



=



+

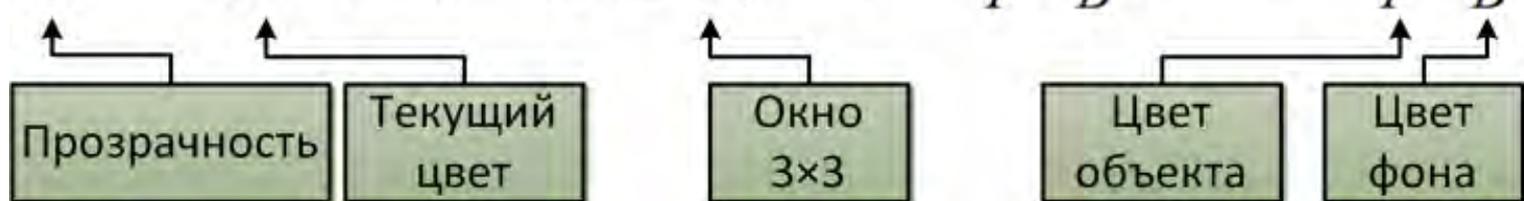


$$I = \alpha F + (1 - \alpha) B$$

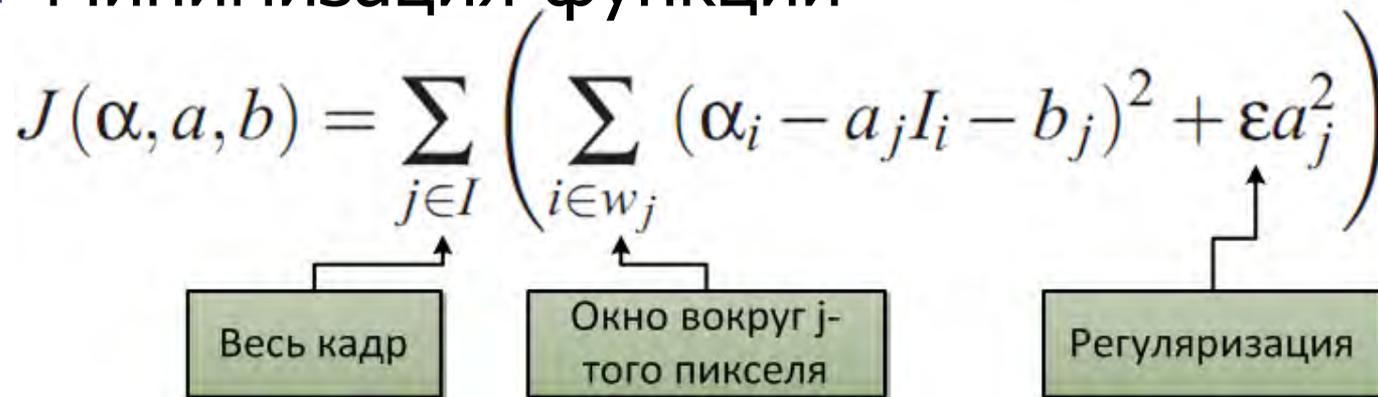
Matting

Идея

- Предположение о локальной гладкости

$$\alpha_i \approx aI_i + b, \quad \forall i \in w, \quad a = \frac{1}{F-B}, \quad b = -\frac{B}{F-B}$$


- Минимизация функции

$$J(\alpha, a, b) = \sum_{j \in I} \left(\sum_{i \in w_j} (\alpha_i - a_j I_i - b_j)^2 + \epsilon a_j^2 \right)$$


Matting

Результаты (1)

Matting disabled



Matting enabled



Matting

Результаты (2)



Исходное изображение

Matting

Результаты (3)



Восстановленный задний план

Matting

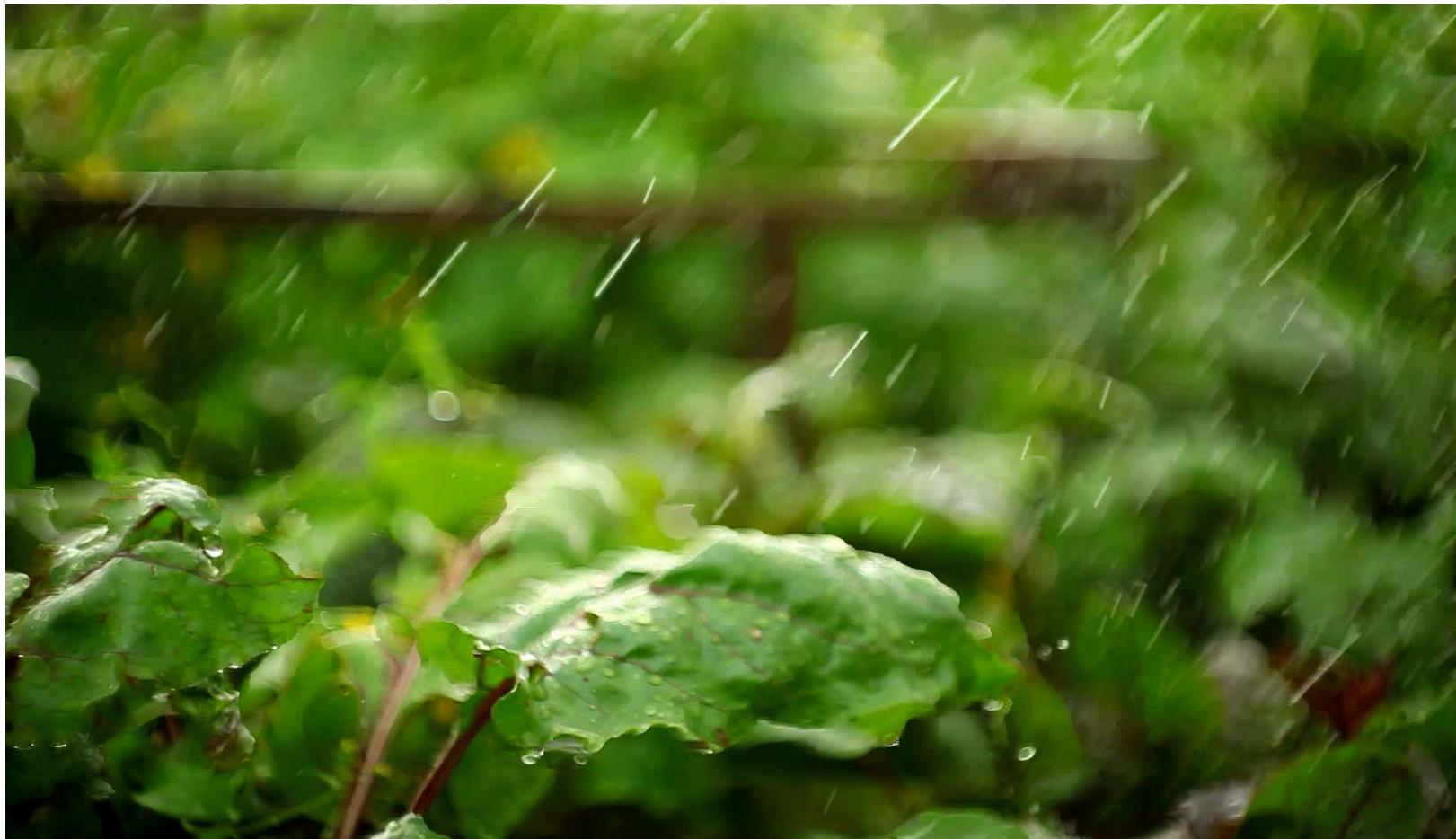
Результаты (4)



Результат в объеме

Matting

Результаты (5)



Matting

Результаты (6)

Block-based inpainting

