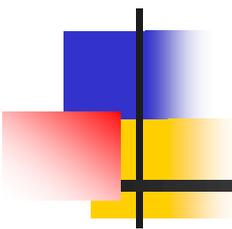
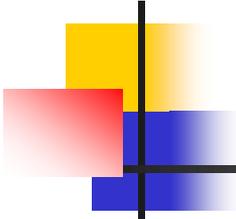


Методы повышения визуального качества восстановленного фона



Михаил Ерофеев

Video Group
CS MSU Graphics & Media Lab



Содержание

- **Введение**
- Гибридное восстановление фона
- **Бесшовный монтаж**
 - Laplacian Pyramid
 - Optimal seam
 - Poisson Image Editing
 - Деформация структуры
- Заключение

Введение

Постановка задачи

Для данного видео и маски неизвестных точек восстановить, заполнить эти точки таким образом, чтобы человек не был способен обнаружить следов редактирования



Исходное видео



Восстановленный фон

Введение

Восстановление фона для стерео



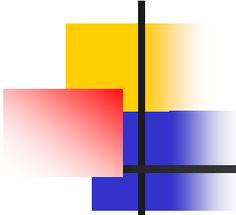
Исходный кадр



Левый построенный ракурс



Правый построенный ракурс



Содержание

- Введение
- **Гибридное восстановление фона**
- **Бесшовный монтаж**
 - Laplacian Pyramid
 - Optimal seam
 - Poisson Image Editing
 - Деформация структуры
- Заключение

Восстановление фона

Пространственно-темпоральный метод

Краткая схема метода:

1. Оценить движение с помощью ME
2. Интерполировать поле ME для неизвестных областей
3. Провести сегментацию объект\фон
4. Заполнить неизвестные области согласно приоритетам

Детали метода

Интерполяция векторов ME

$$V(x, y) = \frac{1}{2} ((1 - x_n)V_L + x_nV_R + (1 - y_n)V_T + y_nV_B)$$

$$x_n = \frac{x - x_L}{x_R - x_L} \quad y_n = \frac{y - y_L}{y - y_L}$$

$V(x, y)$ – вектор движения точки с координатами (x, y)

L, R, T, B – блоки расположенные слева, справа, СВЕРХУ и СНИЗУ от интерполируемой точки

Детали метода

Сегментация объект\фон

Используется простой метод сегментации – сравнение модуля векторов ME с порогом

Цель сегментации:

- Восстанавливать неизвестные области переднего плана перед восстановлением фона
- Не допустить использования фрагментов переднего плана для восстановления фона

Детали метода

Приоритеты блоков

■ Неизвестные блоки восстанавливаются в порядке убывания их приоритетов

$$P(p) = C(p)D(p)$$

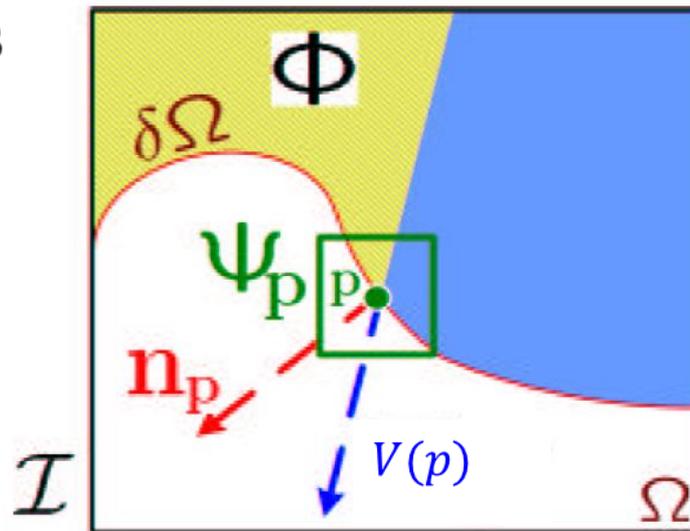
$$C(p) = \frac{\sum_{q \in \Psi_p \cap (I - \Omega)} C(q)}{|\Psi_p|}$$

$$D(p) = \frac{|V(p) \cdot n_p|}{\alpha}$$

I – исходное изображение

Ω – неизвестная область

Ψ_p – блок с центром в точке p



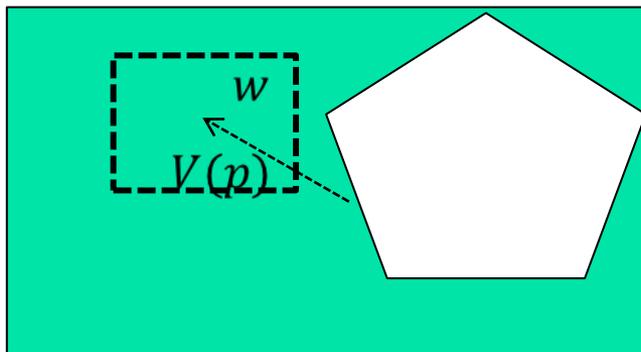
$$C(q) = \begin{cases} 1, & \text{пиксель известен} \\ 0, & \text{иначе} \end{cases}$$

n_p – нормаль к границе неизвестной области

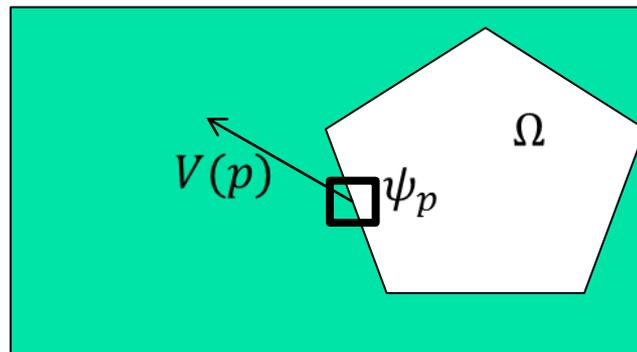
Детали метода

Выбор блоков кандидатов

В окне w (31×31 пиксель) предыдущего кадра сдвинутом на вектор $V(p)$ относительно точки p производится поиск N наиболее похожих блоков по метрике SSD по компонентам (R, G, B, V_x, V_y)



Предыдущий кадр



Текущий кадр

Детали метода

Восстановление блока

Блок Ψ_p восстанавливается на основе блоков кандидатов путем:

- Выбора кандидата с наименьшим значением SSD
- Построением линейной комбинации кандидатов минимизирующей SSD

Выбор стратегии восстановления происходит автоматически с целью оптимизировать значение SSD

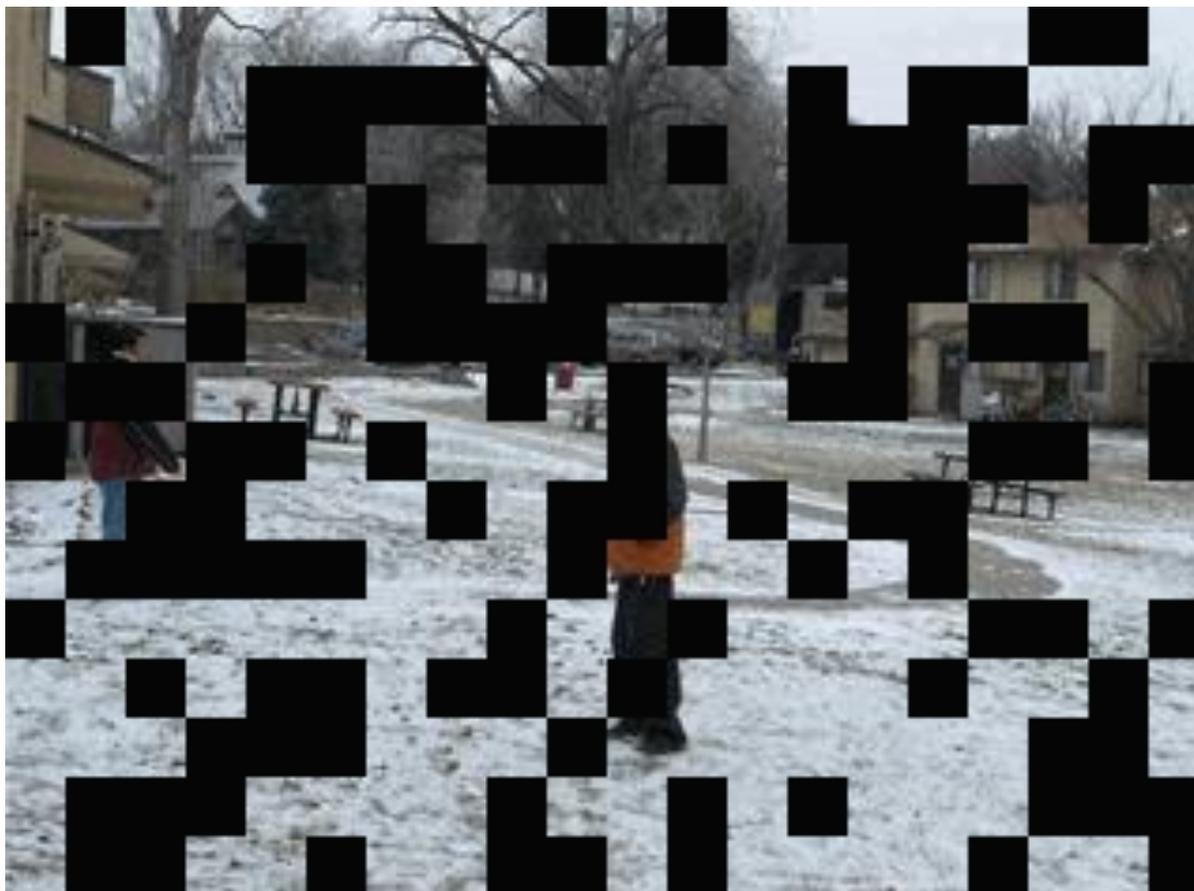
Предложенный метод

Ключевые моменты

- Адаптация зарекомендовавшего себя для изображений приоритета восстановления для видео
- Поиск блоков кандидатов с помощью комбинации ME и поиска ближайших блоков внутри окна
- «Умное» смешение блоков с наименьшим SSD

Предложенный метод

Результаты (1)



Поврежденный кадр (PSNR=9,01 дБ)

Предложенный метод

Результаты (1)



Восстановление на основе только векторов ME (PSNR=28,26 дБ)

Предложенный метод

Результаты (1)



Восстановление с помощью предложенного метода (PSNR=32,23 дБ)

Предложенный метод

Результаты (2)



Исходная последовательность

Предложенный метод

Результаты (2)



Поврежденная последовательность (PSNR=14,22 дБ)

Предложенный метод

Результаты (2)



Восстановление на основе только векторов ME (PSNR=31,23 дБ)

Предложенный метод

Результаты (2)



Восстановление на основе только поиска внутри окна (PSNR=23,41 дБ)

Предложенный метод

Результаты (2)



Восстановление с помощью предложенного метода (PSNR=34,96 дБ)

Предложенный метод

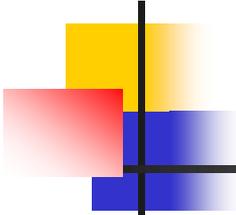
Анализ

Достоинства:

- Субъективное и объективно более высокое качество по сравнению с восстановлением только на основе МЕ
- Уменьшение заметности шва с помощью поиска кандидатов внутри окна и «умного» смешения

Недостатки:

- Ненадежная сегментация объект\фон
- Тенденция к распространению ошибок сегментации по всем кадрам результата



Содержание

- Введение
- Гибридное восстановление фона
- **Бесшовный монтаж**
 - Laplacian Pyramid
 - Optimal seam
 - Poisson Image Editing
 - Деформация структуры
- Заключение

Бесшовный монтаж

Введение

Классические применения:

- Склейка панорамы
- Фотомонтаж

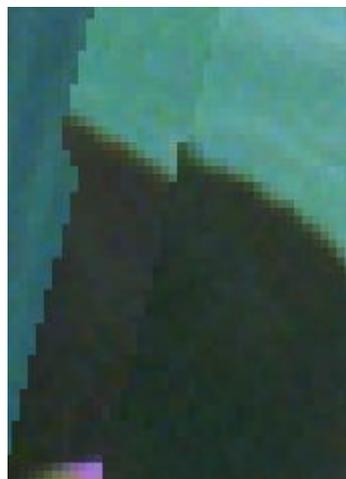
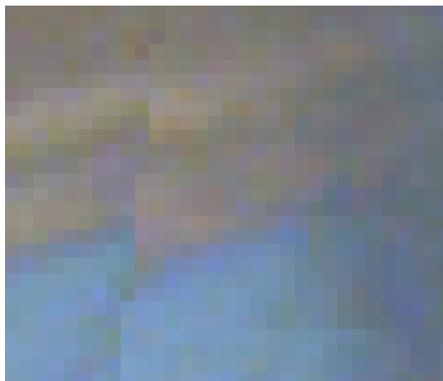


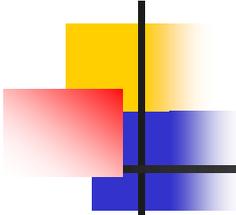
Бесшовный монтаж

Применимость к восстановлению фона

Все известные автору доклада методы восстановления фона разрабатывавшиеся в видео группе сталкивались с проблемой швов. Идея: использовать методы бесшовного монтажа для маскировки шва

Примеры швов





Содержание

- Введение
- Гибридное восстановление фона
- Бесшовный монтаж
 - **Laplacian Pyramid**
 - Optimal seam
 - Poisson Image Editing
 - Деформация структуры
- Заключение

Laplacian Pyramid

Идея метода

Простого смешения изображений в области перекрытия не достаточно, чтобы обеспечить плавный переход и избежать двоения границ. Авторы предлагают независимо смешивать различные частотные диапазоны.

Laplacian Pyramid

Шаги алгоритма

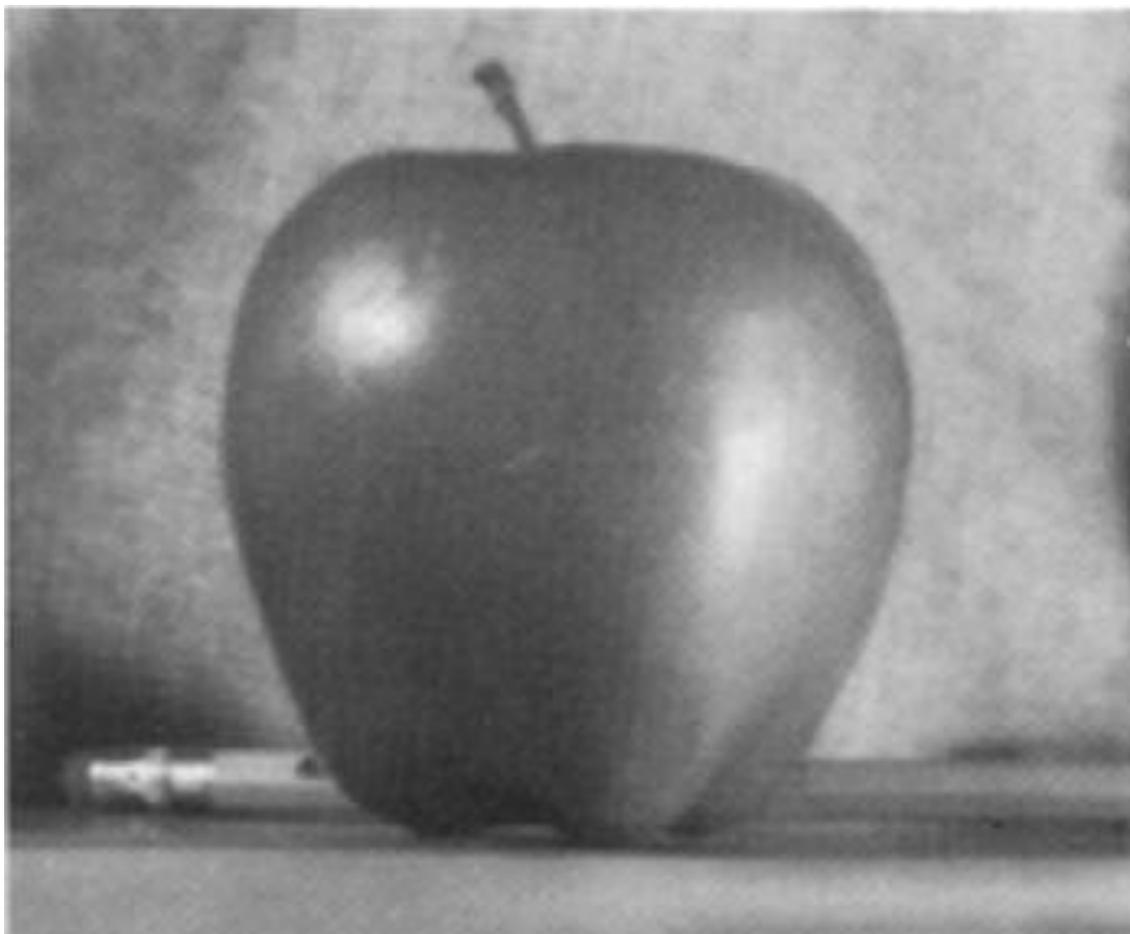
1. Построить пирамиду от высоких до низких частот для смешиваемых изображений
2. Провести независимое смешивание уровней пирамиды, применяя более плавный переход для низких частот и более резкий для высоких
3. Сложить уровни полученной пирамиды



Пример пирамиды разрешений

Laplacian Pyramid

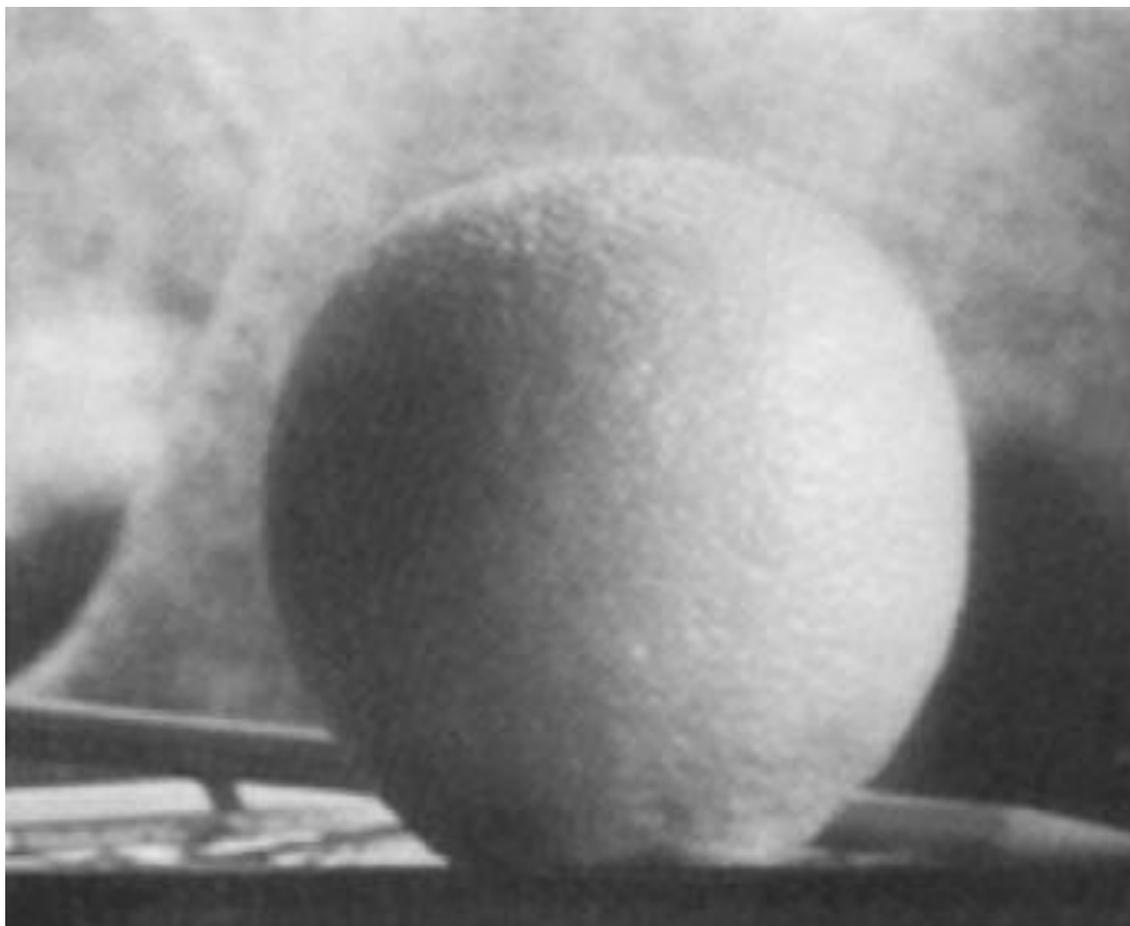
Результат работы



«Яблоко»

Laplacian Pyramid

Результат работы

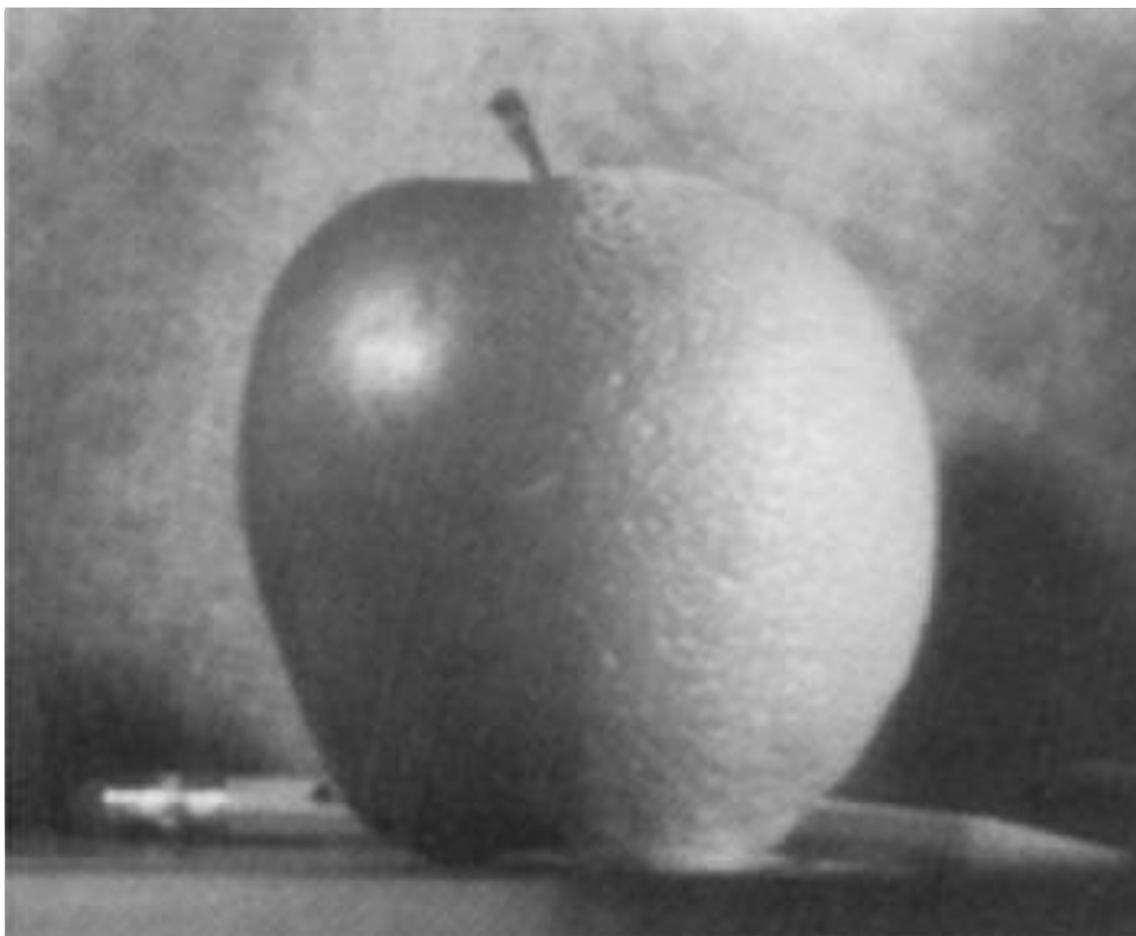


«Апельсин»

E. H. Adelson, et al, "Pyramid methods in image processing," RCA Engineer

Laplacian Pyramid

Результат работы



«ЯБЛОКОСИН»

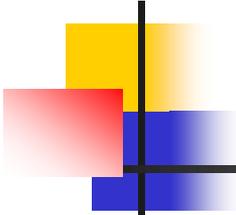
E. H. Adelson, et al, "Pyramid methods in image processing," RCA Engineer

Laplacian Pyramid

Авторы метода



Авторы метода: Bergen, Anderson, Adelson, Burt



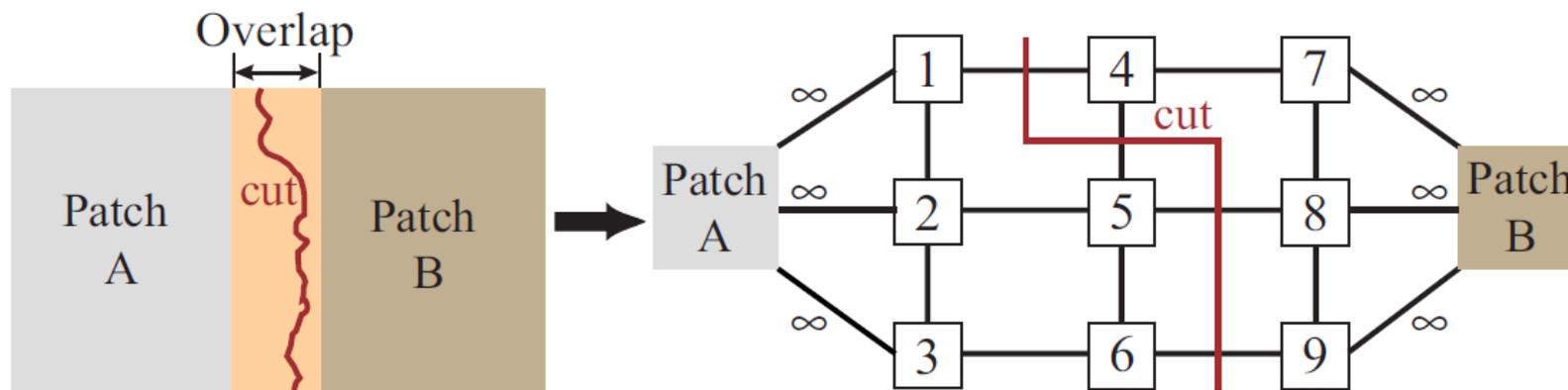
Содержание

- Введение
- Гибридное восстановление фона
- **Бесшовный монтаж**
 - Laplacian Pyramid
 - **Optimal seam**
 - Poisson Image Editing
 - Деформация структуры
- Заключение

Graphcut для монтажа

Идея метода

Применить метод Graphcut для поиска оптимального шва между монтируемыми изображениями. Определить веса ребер как значение SSD перекрывающихся пикселей



Graphcut для монтажа

Результаты (1)



Исходное изображение

Результат монтажа

Graphcut для монтажа

Результаты (2)



Исходное изображение



Результат монтажа



Исходные изображения



Результат монтажа

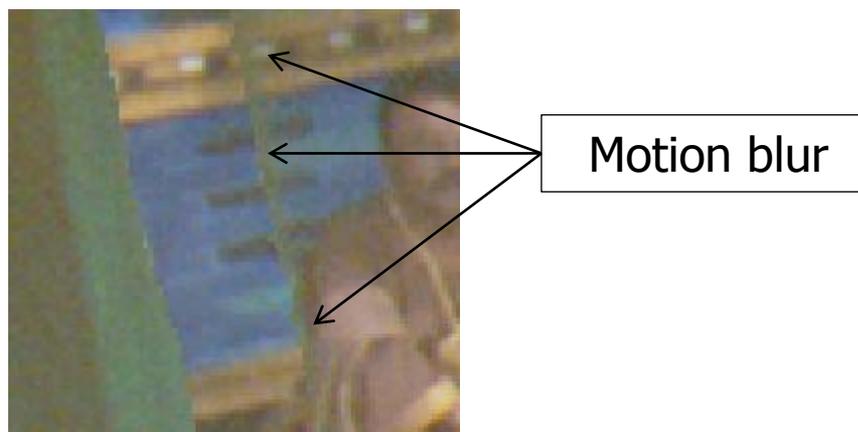


Выбранный шов

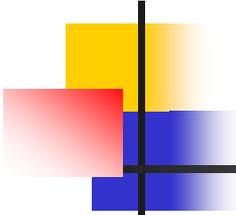
Graphcut для монтажа

Применимость к восстановлению фона

Есть надежда, что оптимальный шов может отрезать остатки motion blur на фоне



Пример остатков motion blur
на фоне



Содержание

- Введение
- Гибридное восстановление фона
- **Бесшовный монтаж**
 - Laplacian Pyramid
 - Optimal seam
 - **Poisson Image Editing**
 - Деформация структуры
- Заключение

Poisson Image Editing

Основная идея

Для того чтобы вставить изображение g в область Ω изображения f^* , решим следующую задачу:

$$\min_f \iint_{\Omega} |\nabla f - \mathbf{v}|^2 \quad \text{с граничным условием} \quad f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

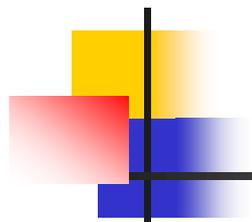
f – искомый результат монтажа

Задача может быть переформулирована:

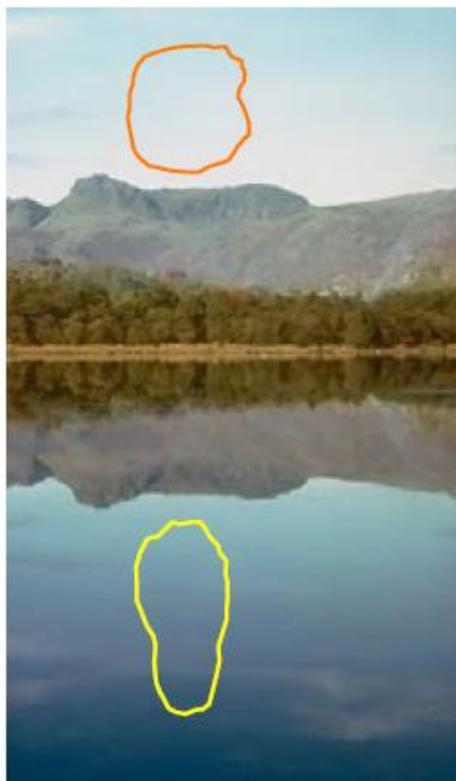
$$\Delta f = \operatorname{div} \mathbf{v} \text{ over } \Omega, \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

Poisson Image Editing

Результаты (1)



Исходные изображения



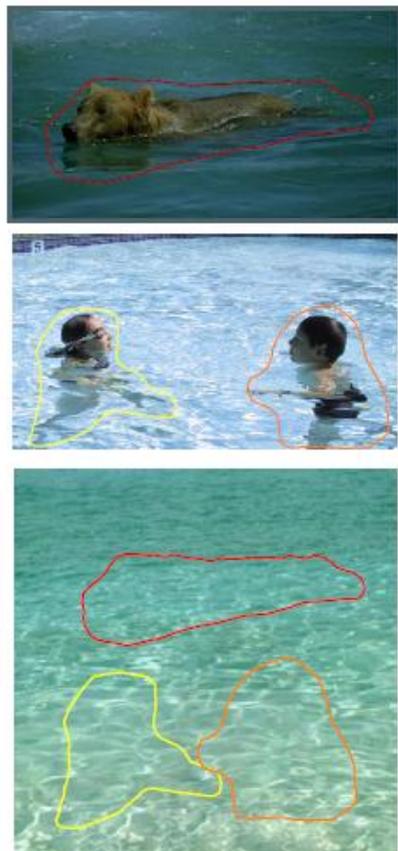
Копирование



Предложенный метод

Poisson Image Editing

Результаты (2)



Исходные изображения



Копирование



Предложенный метод

Poisson Image Editing

Результаты (3)



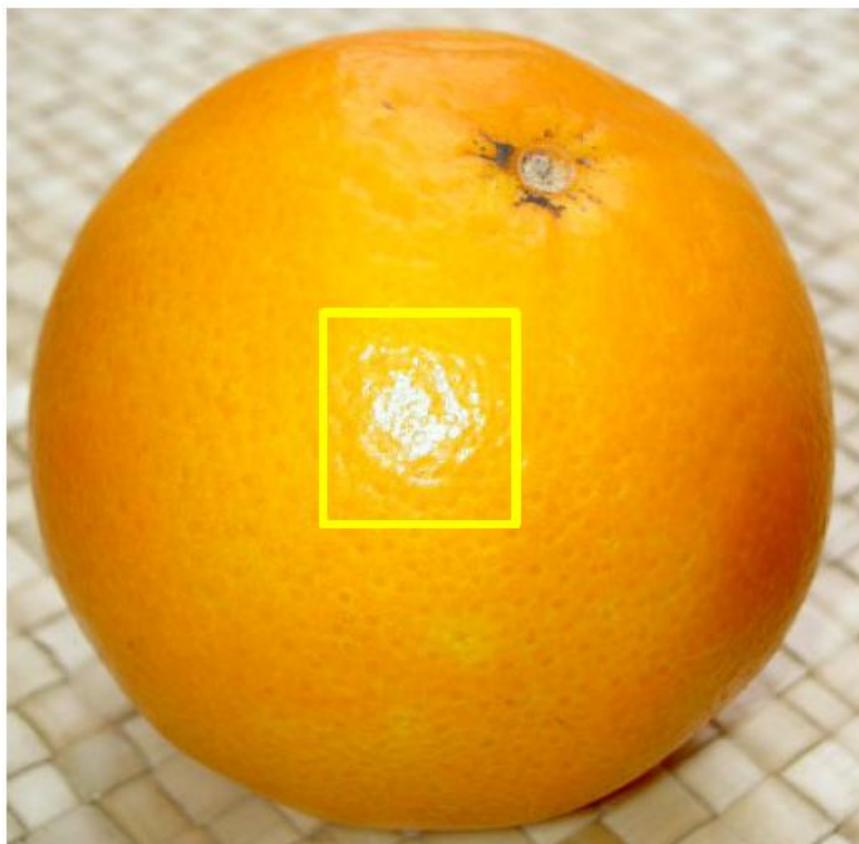
Исходное изображение



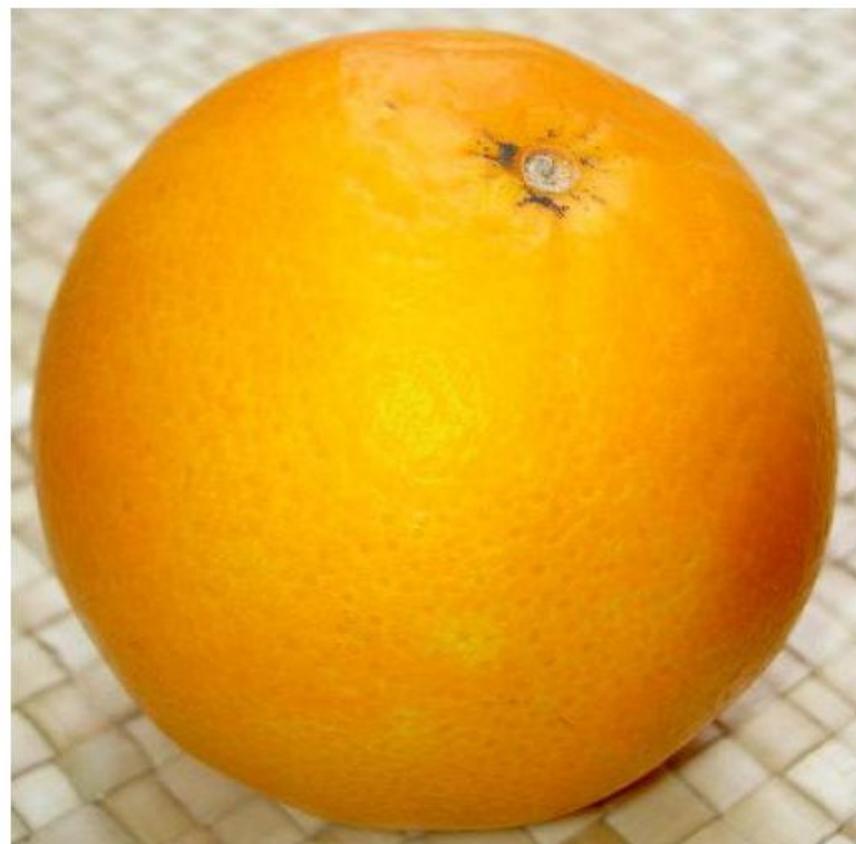
Локальное изменение контраста

Poisson Image Editing

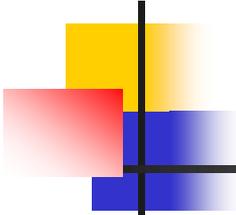
Результаты (3)



Исходное изображение



Локальное изменение контраста



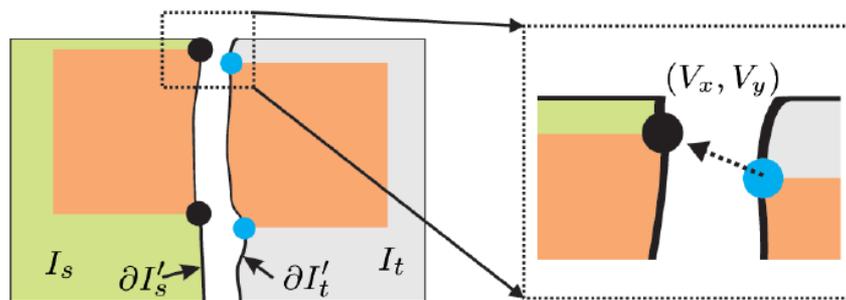
Содержание

- Введение
- Гибридное восстановление фона
- **Бесшовный монтаж**
 - Laplacian Pyramid
 - Optimal seam
 - Poisson Image Editing
 - **Деформация структуры**
- Заключение

Деформация структуры

Основная идея

1. Найти оптимальный шов между монтируемыми изображения
2. Выделить и сопоставить салиентные точки вдоль шва
3. Построить гладкую модель деформации
4. Применить деформацию к полю градиента
5. Построить результат, решив уравнение Пуассона

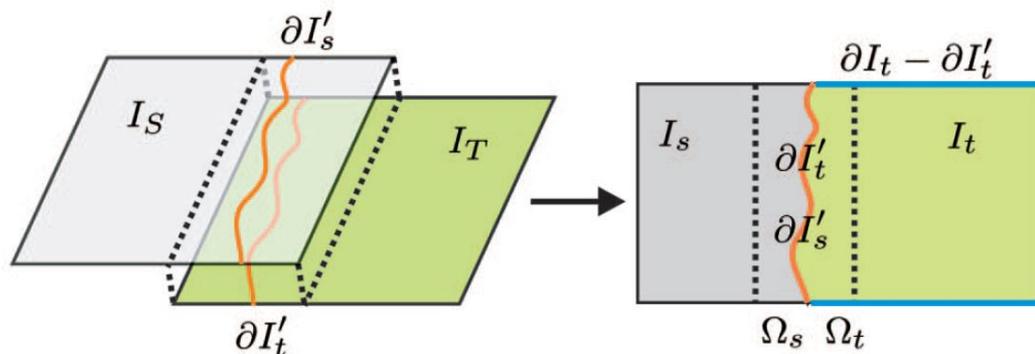


Пример сопоставления салиентных точек

Деформация структуры

Поиск оптимального шва

I_S, I_T – монтируемые изображения
 Ω – область наложения



Для решения применяется Graphcut
 Веса ребер:

$$S(p, q) = \sum_{r,g,b} ((1 - \beta)S_m + \beta S_d)$$

$$S_m(p, q) = \|\nabla I_S(p)\| + \|\nabla I_S(q)\| + \|\nabla I_T(p)\| + \|\nabla I_T(q)\|$$

$$S_d(p, q) = \|\nabla_x I_S(p) - \nabla_x I_T(p)\| + \|\nabla_x I_S(q) - \nabla_x I_T(q)\| + \\ \|\nabla_y I_S(p) - \nabla_y I_T(p)\| + \|\nabla_y I_S(q) - \nabla_y I_T(q)\|$$

Деформация структуры

Поиск салиентных точек

1. Вдоль найденного на предыдущем шаге шва для изображений I_S и I_T вычислить значение градиента
2. Бинаризовать по порогу (50) модуль градиента
3. Удалить не локальные максимумы
4. Вычислить силу ($g_k(s)$) каждой найденной точки, как проекцию градиента на касательную к шву

Деформация структуры

Сопоставление салиентных точек

Пусть для I_T найдено m , а для $I_S - n$, и $n \geq m$
Требуется найти монотонное соответствие между салиентными точками, т.е. минимизировать такую функцию

$$E' = \min \sum_{0 \leq i < m} (g_T(i) - g_S(k_i))^2$$

s.t. $0 \leq k_0 < k_1 < \dots < k_{m-1} < n$

Точкам оставшимся без пар сопоставим точку с наиболее схожей силой, не нарушающую монотонности преобразования или оставим без пары

Деформация структуры

Построение итоговой деформации

■ Имеем разряженное поле векторов деформации вдоль границы V

S'_t – область, в которой выполняется деформация ($S'_t = \Omega$ или $S'_t = I_T$ на выбор пользователя)

Построим гладкое поле деформации, решив задачу

$$\mathbf{V}^* = \arg \min_V \int_{p \in S_t} \|\nabla \mathbf{V}\|^2 dp \quad \mathbf{V}(p) = \mathbf{0} \quad \forall p \in \partial S'_t$$

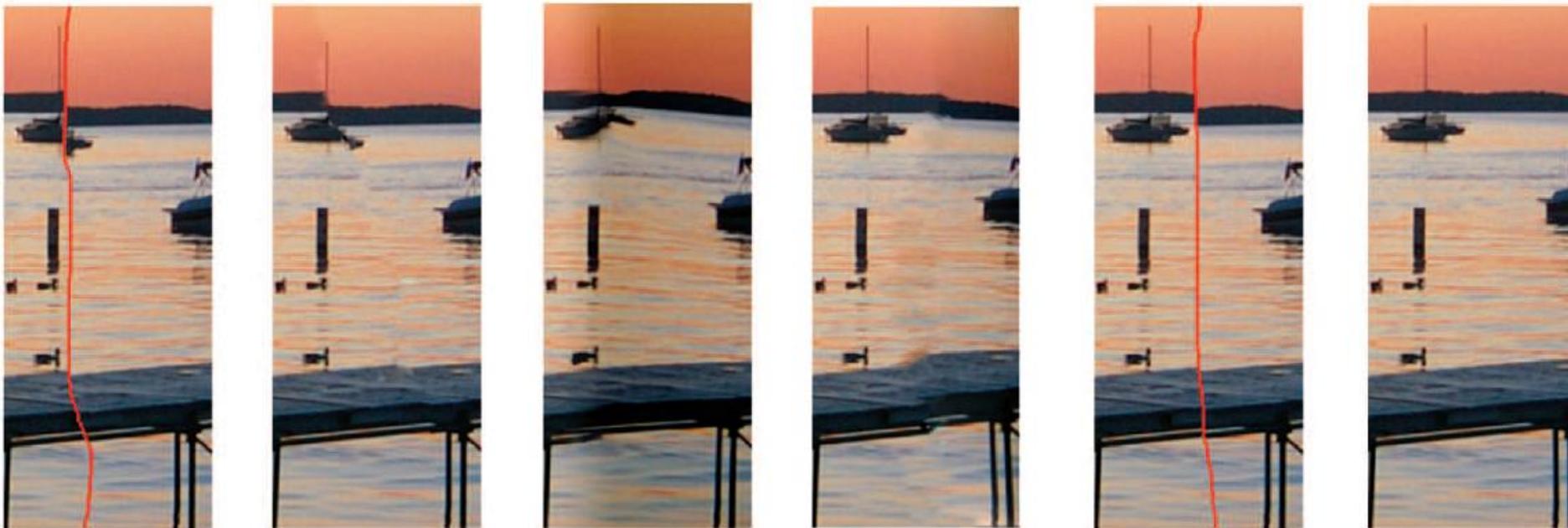
Деформация структуры

Построение итогового результата

- Деформируем поле градиентов I_T
2. Вычислим результат, решив уравнение Пуассона

Деформация структуры

Результаты (1)



Конкуренты

Предложенный метод.
Найденный шов и
конечный результат

Деформация структуры

Результаты (2)



Оптимальный шов



Простое смешение



Предложенный метод

Деформация структуры

Результаты (3)



Простое смешение



Оптимальный шов



GIST1



Деформация структуры



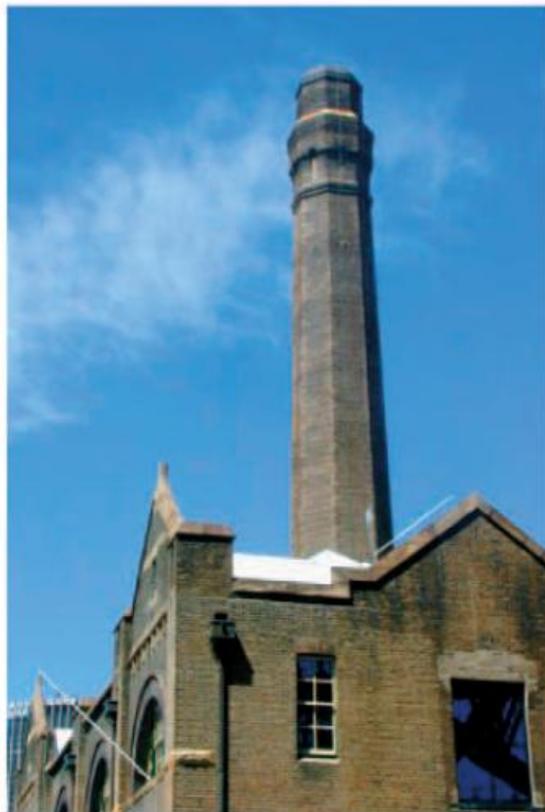
Poisson Editing



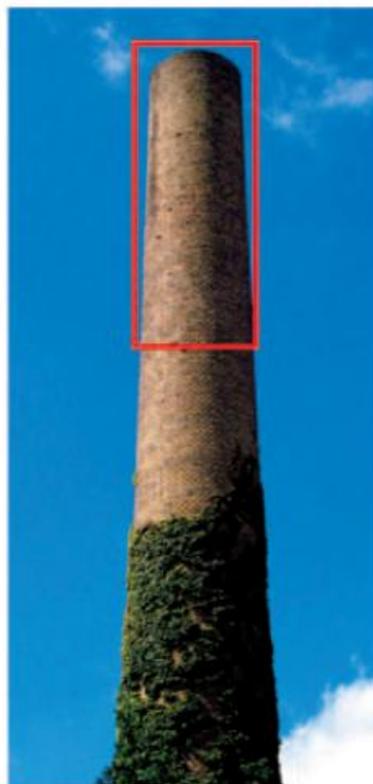
Предложенный метод

Деформация структуры

Результаты (4)



Исходные изображения



Предложенный метод

Предложенный метод

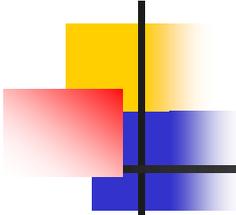
Анализ

Достоинства:

- Способен смоделировать произвольную деформацию
- Визуальное качество превосходит другие state of art методы

Недостаток:

- Неверное сопоставление салиентных точек может привести к нестабильности во времени



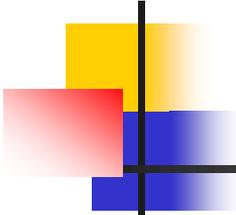
Содержание

- Введение
- Гибридное восстановление фона
- **Бесшовный монтаж**
 - Laplacian Pyramid
 - Optimal seam
 - Poisson Image Editing
 - Деформация структуры
- **Заключение**

Заключение

Направления дальнейшей работы

- Уточнение экстраполированного вектора с помощью поиска в небольшом окне
- Применение компонент (R, G, B, V_x, V_y) при поиске блока кандидата
- «Умное» смешение блоков кандидатов
- Применение Graphcut для удаления остатков motion blur
- Возможно, ряд идей из последней статьи



Литература

1. M. Ebdelli, C. Guillemot, and O. Le Meur, "Exemplar-based video inpainting with motion-compensated neighbor embedding," in *Proc. of the IEEE Intl. Conf. on Image Processing, ICIP*, 2012.
2. E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, "Pyramid methods in image processing," *RCA Engineer*, 29(6):33–41, 1984.
3. V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," in *SIGGRAPH*, 2003.
4. P. Perez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM SIGGRAPH*, 22(3):313–318, 2003.
5. J. Jia and C.-K. Tang, "Image stitching using structure deformation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):617–631, 2008.