

Школа Анализа Данных
Яндекса

Курс «Анализ изображений и видео, ч.2»

Лекции №10 «Бинокулярное плотное стерео»

Антон Конушин

Заведующий лабораторией компьютерной графики и мультимедиа
ВМК МГУ

21 апреля 2017 года

Благодарность



Основа презентации была подготовлена Глебом Кривовязем для спецсеминара «Компьютерное зрение» на ВМК МГУ

При подготовке были использованы материалы из курса «*Stereo Vision*» от Michael Bleyer (Vienna University of Technology):

http://www.ims.tuwien.ac.at/teaching_detail.php?ims_id=188.HQK

План



- **Введение**
- Локальные методы
- Глобальные методы
- Использование сегментации



Параллакс



- Параллакс - видимое смещение объекта в зависимости от точки обзора
- Чем объект ближе, тем смещение больше

Стереопсис



«**Стереопсис** (англ. stereopsis) - сенсорный процесс, возникающий при бинокулярном зрении как психофизическая реакция на сетчаточную горизонтальную диспаратность. В результате С. субъект переживает специфическое ощущение глубины. <...> ». (Психологическая энциклопедия)



Левое изображение



Правое изображение

Имитация 3D



Так мы обычно смотрим на экран (2D):

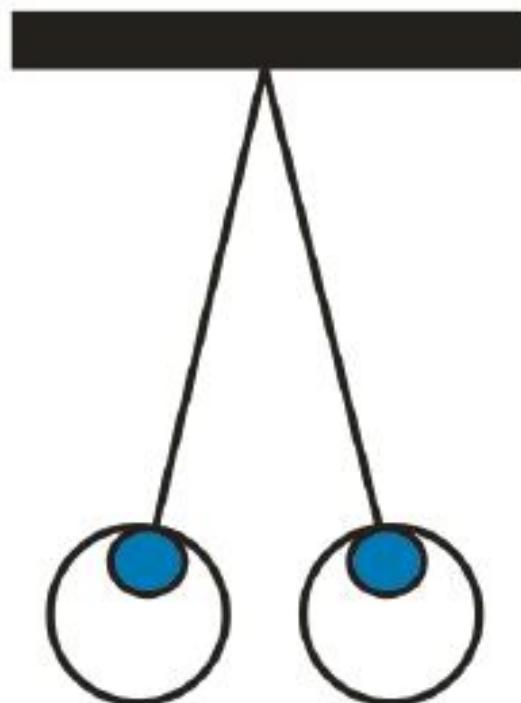
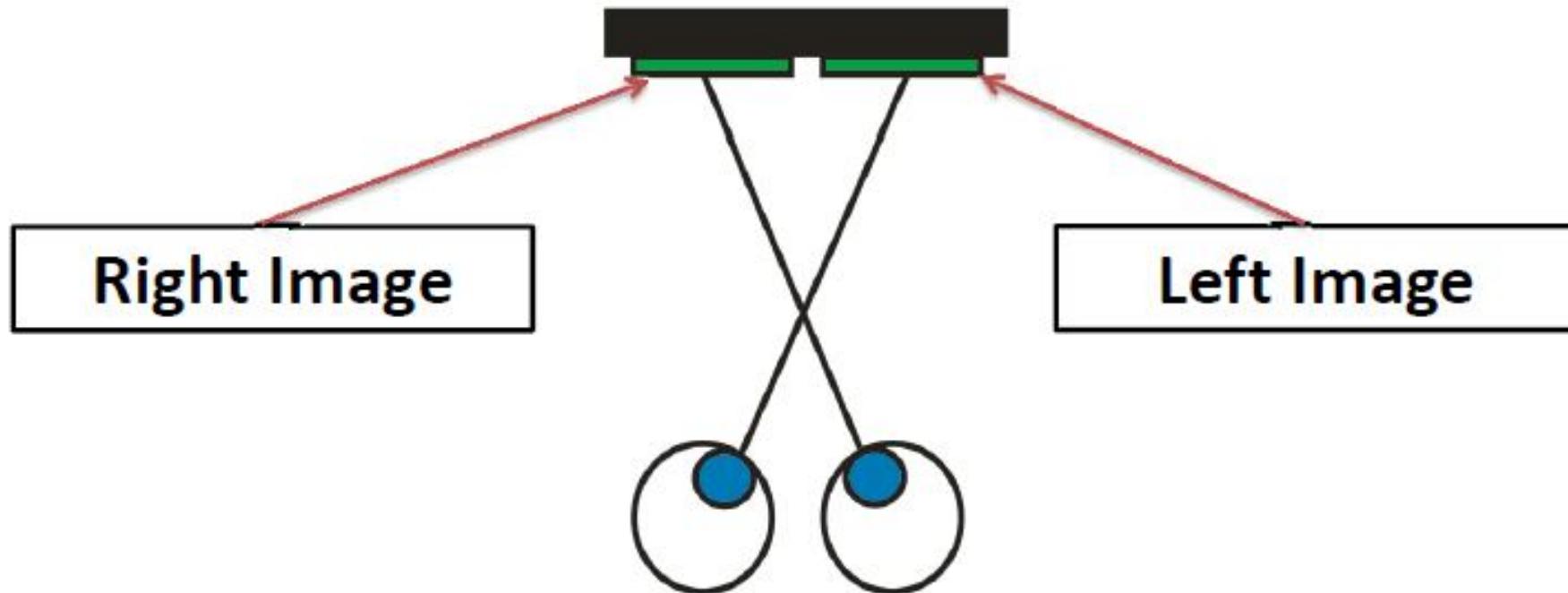


Рисунок: M. Bleyer

Имитация 3D



А так можно увидеть 3D-картинку:



Имитация 3D. Пример



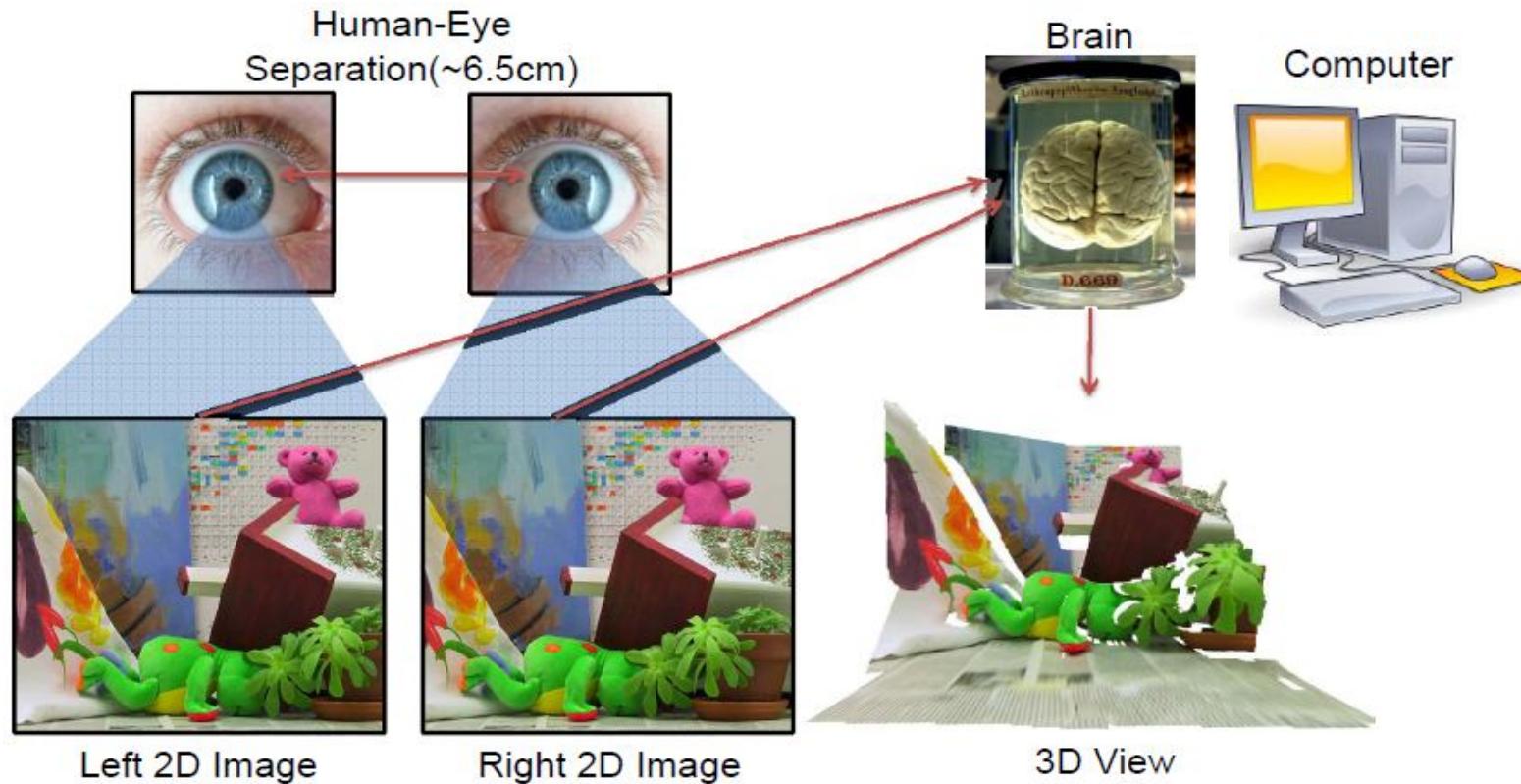
3D-картинка появится посередине!



Для правого глаза

Для левого глаза

Плотное стерео



- Задача плотного стерео: восстановить в 3D все видимые на изображениях точки
 - Если 2 изображения – то бинокулярное стерео
 - Если >2 – многовидовое (на следующей лекции)
- Калибровку камер будем считать известной



Ректификация изображений



Вычисление соответствующих точек



Восстановление 3D путем триангуляции

Общая схема



Ректификация изображений

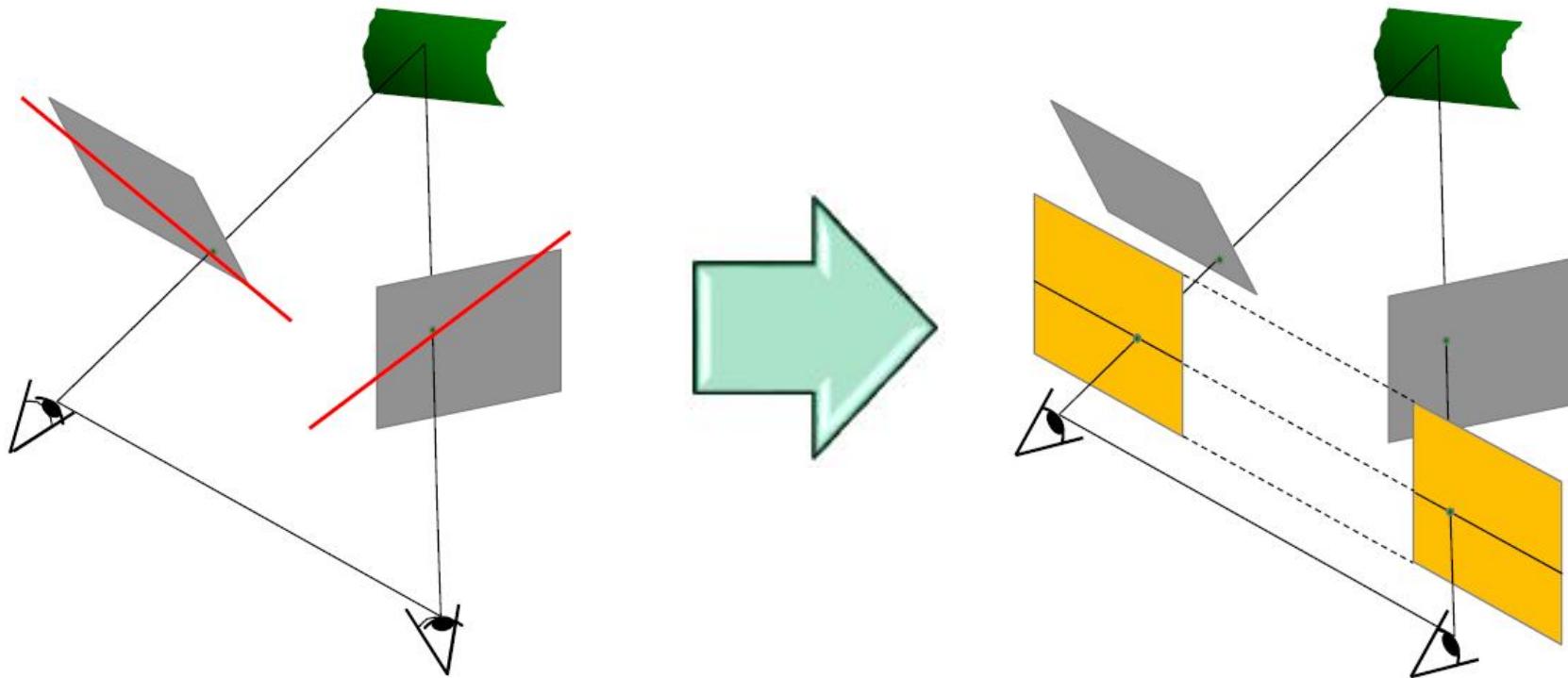


Вычисление соответствующих точек



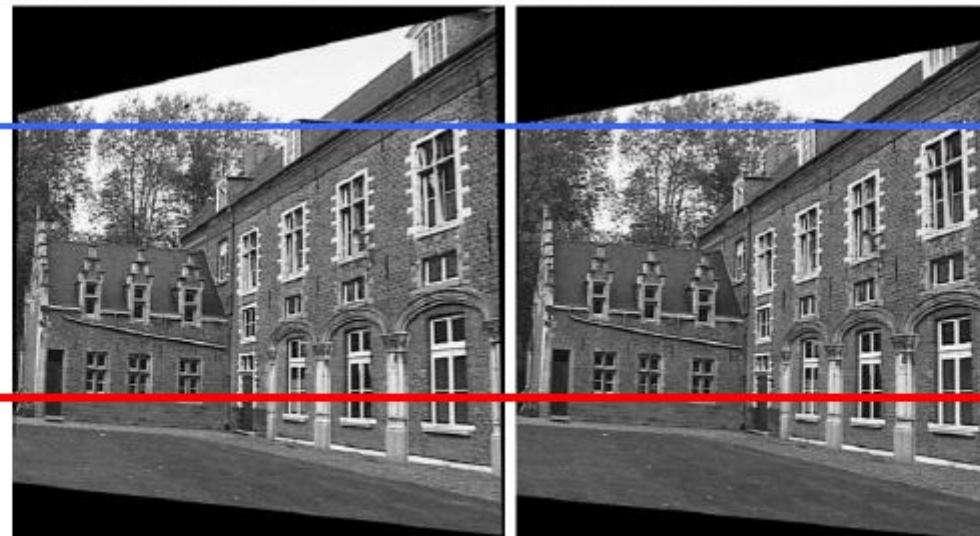
Восстановление 3D путем триангуляции

Ректификация



- «Ректификация» – преобразование стереопары в изображения, в которых соответствующие эпиполярные линии лежат на одной и той же горизонтальной строке
- Первый способ: проецирование на общую плоскость с помощью гомографии

Ректификация через гомографию



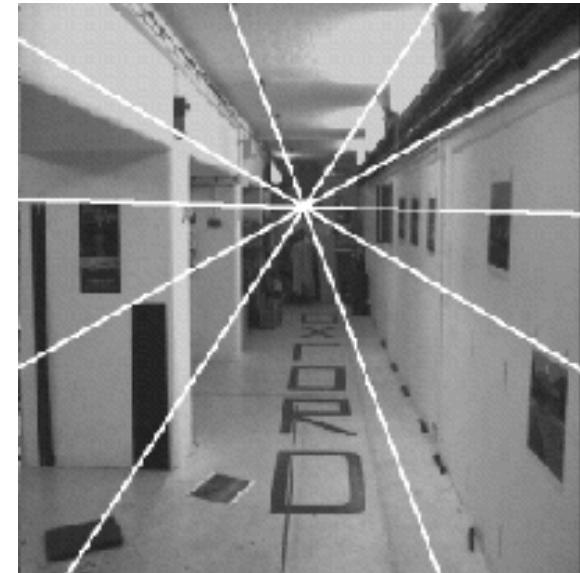
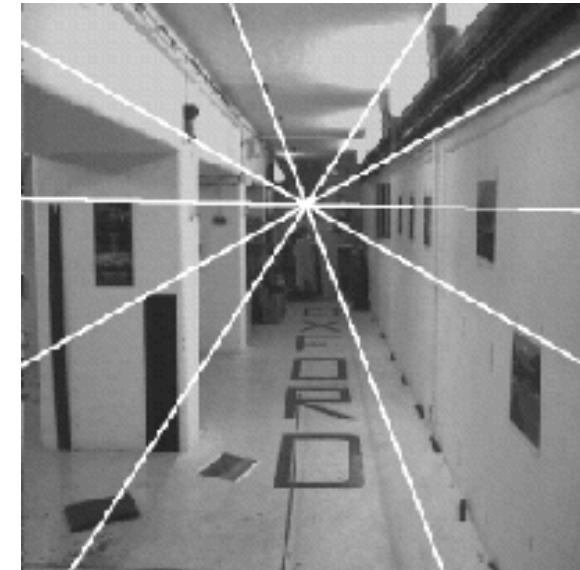
Недостатки:

- Не применим, когда камера движется вперёд или назад
- Сильные искажения в некоторых случаях

Радиальная разёртка



- Polar rectification
 - Индексируем эпиполярные линии углом поворота относительно эпиполи
 - Копируем соответствующие пары эпиполярных линии последовательно в соответствующие горизонтали ректифицированных изображений
- Работает в тех случаях, когда ошибается метод проецирования на общую плоскость



Пример



Проектирование

Полярная
ректификация

Пример





Ректификация изображений

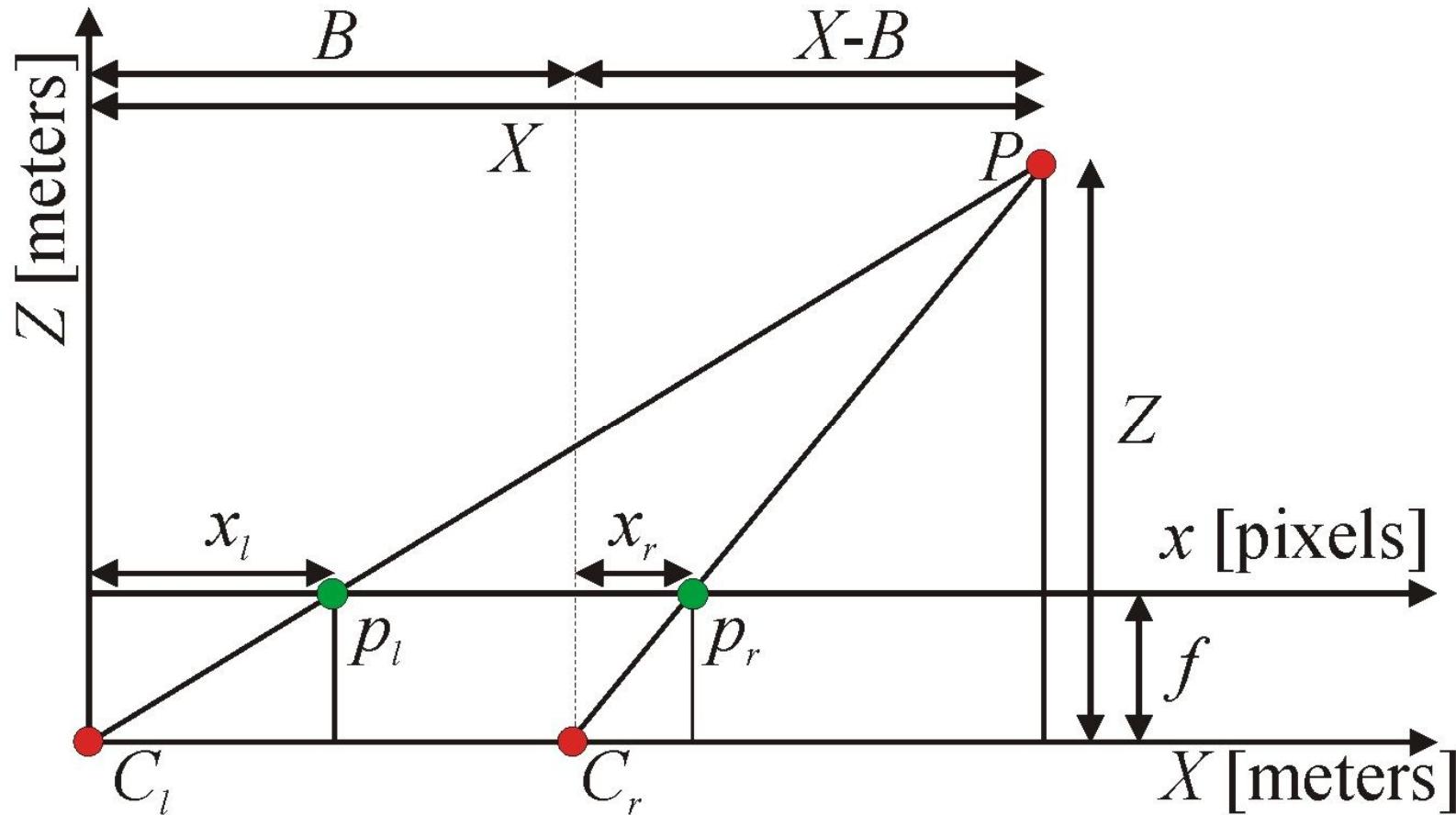


Вычисление соответствующих точек



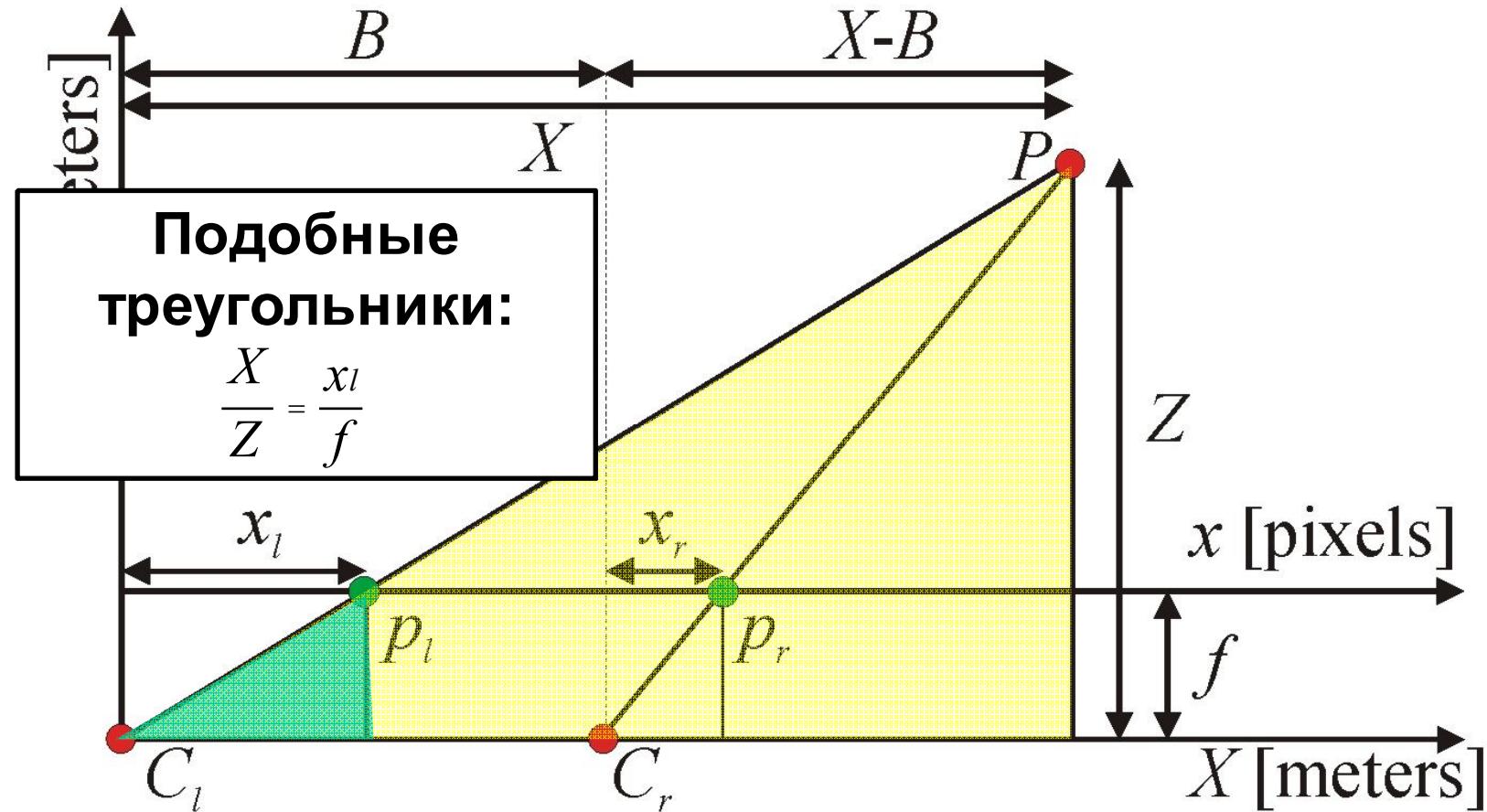
Восстановление 3D путем триангуляции

Триангуляция

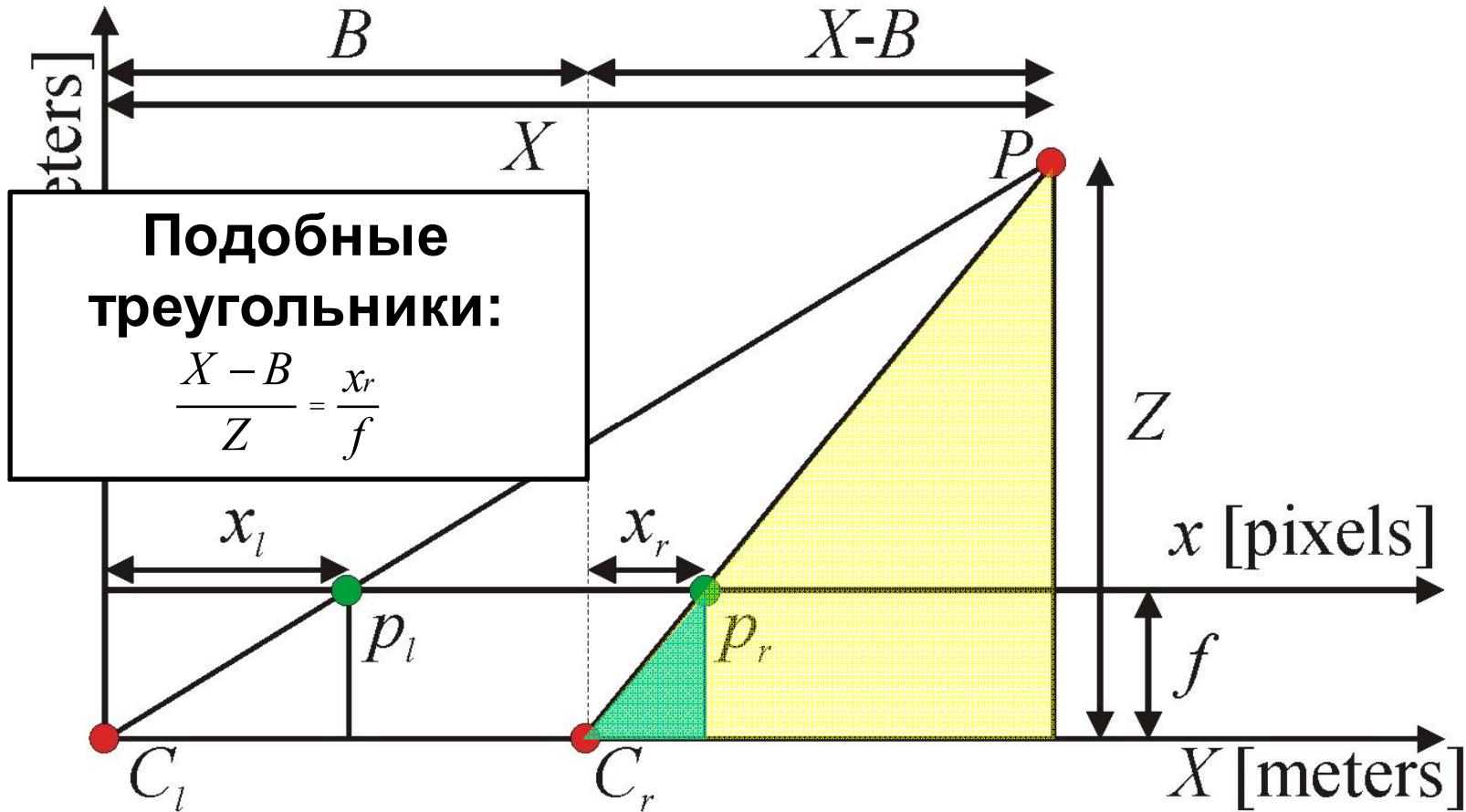


x_l, x_r - смещения относительно принципиальной точки

Триангуляция



Триангуляция





Триангуляция

Из подобия треугольников:

$$\frac{X}{Z} = \frac{x_l}{f} \quad \frac{X - B}{Z} = \frac{x_r}{f}$$

Исключаем X и выражаем Z :

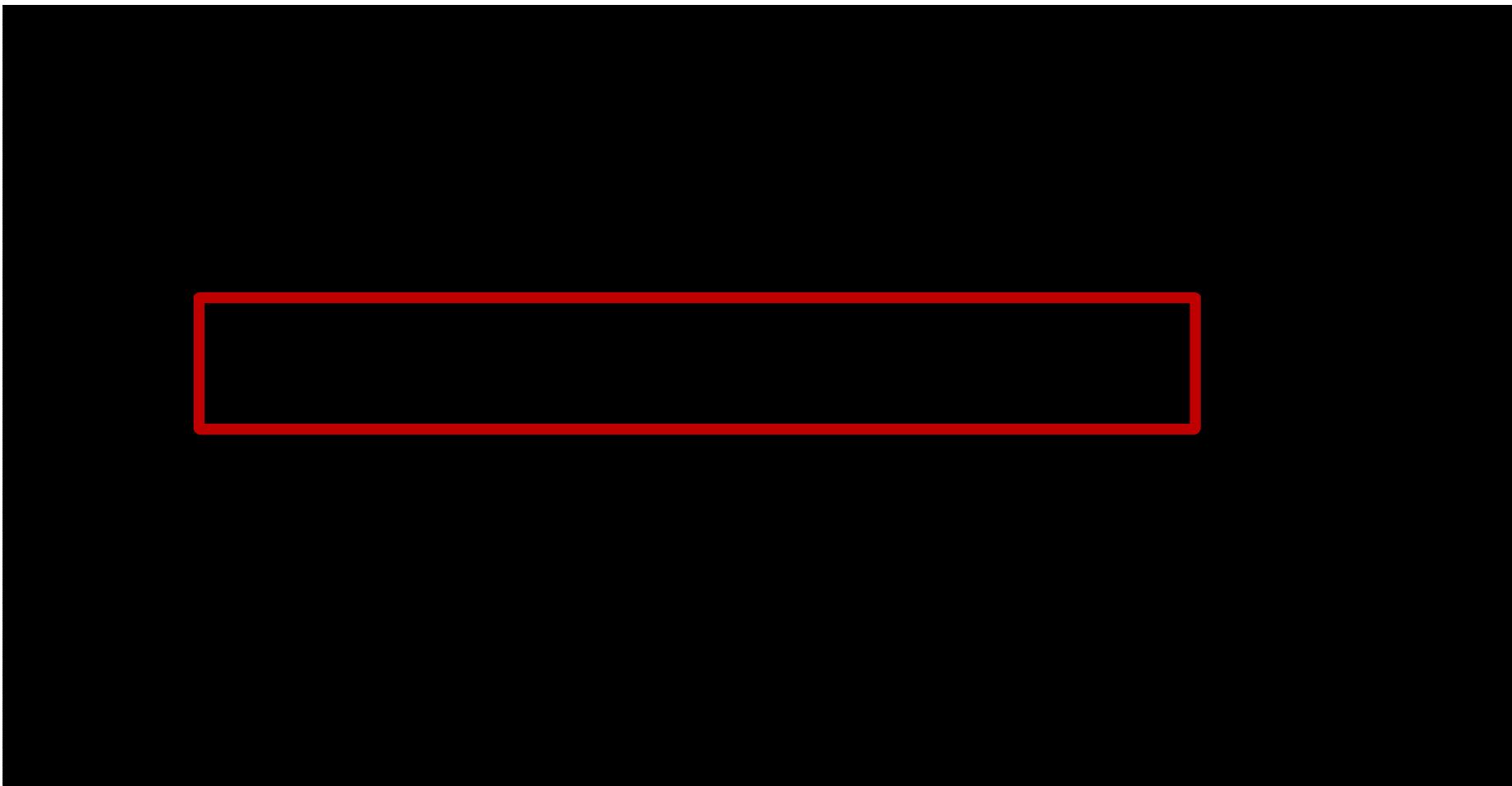
$$Z = \frac{B \cdot f}{x_l - x_r} = \frac{B \cdot f}{d} \quad \text{, где } d = x_l - x_r \text{ - диспаритет}$$

Итоговые координаты 3D-точки:

$$X = \frac{Z \cdot x_l}{f} \quad Y = \frac{Z \cdot y_l}{f} \quad Z = \frac{B \cdot f}{d}$$

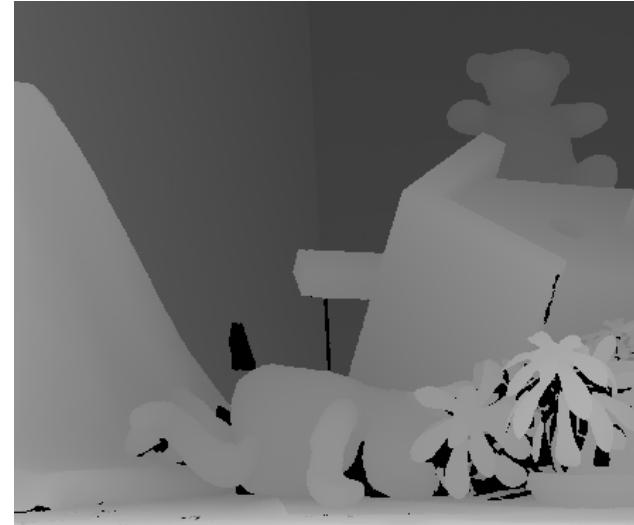
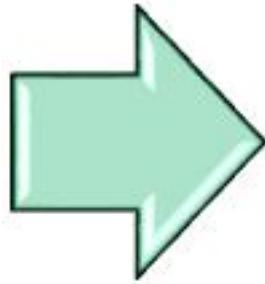
x_l, x_r - смещения относительно принципиальной точки

Общая схема



Вычисление соответствующих точек сводится к задаче
вычисления диспаритета

Вычисление карты диспаритета



Карта диспаритета

- Ректификацию и триангуляцию можно считать чисто техническими задачами.
- Основной интерес представляет вычисление карты диспаритета

Сравнение алгоритмов



Ресурс Middlebury: <http://vision.middlebury.edu/stereo/>

- Стандартные тестовые наборы
- Рейтинг алгоритмов

Error Threshold = 1		Sort by nonocc			Sort by all			Sort by disc						
Algorithm	Avg.	Tsukuba			Venus			Teddy			Cones			Average percent of bad pixels (explanation)
		ground truth	nonocc	all	disc	ground truth	nonocc	all	disc	ground truth	nonocc	all	disc	
AdaptingBP [17]	4.7	1.11 ₉	1.37 ₅	5.79 ₁₁	0.10 ₁	0.21 ₃	1.44 ₂	4.22 ₄	7.06 ₅	11.8 ₅	2.48 ₂	7.92 ₆	7.32 ₃	4.23
CoopRegion [41]	4.8	0.87 ₁	1.16 ₁	4.61 ₁	0.11 ₂	0.21 ₂	1.54 ₄	5.16 ₁₁	8.31 ₇	13.0 ₈	2.79 ₇	7.18 ₃	8.01 ₁₀	4.41
DoubleBP [35]	6.6	0.88 ₃	1.29 ₂	4.76 ₃	0.13 ₅	0.45 ₁₃	1.87 ₉	3.53 ₃	8.30 ₆	9.63 ₂	2.90 ₉	8.78 ₁₇	7.79 ₇	4.19
OutlierConf [42]	7.5	0.88 ₂	1.43 ₇	4.74 ₂	0.18 ₁₂	0.26 ₇	2.40 ₁₅	5.01 ₇	9.12 ₁₀	12.8 ₇	2.78 ₆	8.57 ₁₃	6.99 ₂	4.60
SubPixDoubleBP [30]	10.1	1.24 ₁₆	1.76 ₁₈	5.98 ₁₂	0.12 ₄	0.46 ₁₅	1.74 ₇	3.45 ₂	8.38 ₈	10.0 ₃	2.93 ₁₁	8.73 ₁₆	7.91 ₉	4.39
SurfaceStereo [79]	11.1	1.28 ₂₁	1.65 ₁₂	6.78 ₂₄	0.19 ₁₄	0.28 ₈	2.61 ₂₁	3.12 ₁	5.10 ₁	8.65 ₁	2.89 ₈	7.95 ₈	8.26 ₁₄	4.06
WarpMat [55]	12.1	1.16 ₁₀	1.35 ₄	6.04 ₁₃	0.18 ₁₃	0.24 ₅	2.44 ₁₆	5.02 ₈	9.30 ₁₁	13.0 ₁₀	3.49 ₁₉	8.47 ₁₂	9.01 ₂₄	4.98
Indr+OvrSep [48]	16.2	1.89 _{4n}	2.22 ₂₅	7.22 ₂₂	0.11 ₂	0.22 ₄	1.34 ₁	6.51 ₁₉	9.98 ₁₃	16.4 ₂₁	2.92 ₁₆	8.00 ₉	7.90 ₈	5.30

KITTI dataset

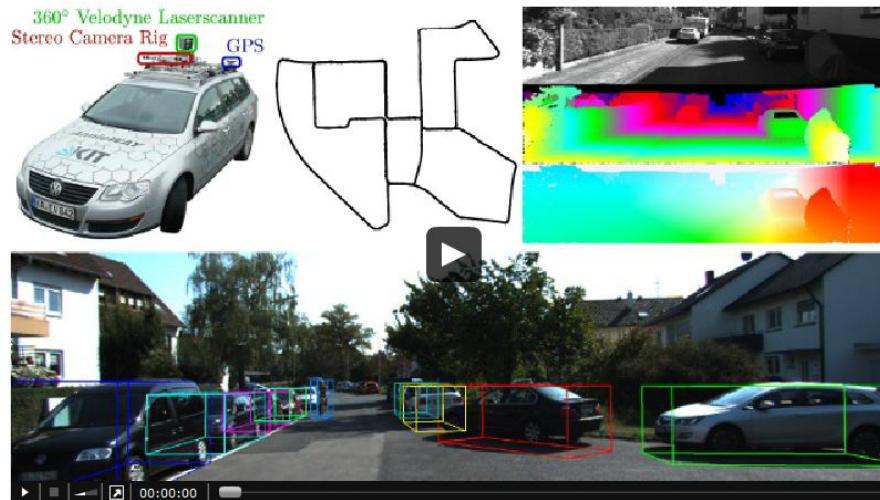


Welcome to the KITTI Vision Benchmark Suite!

We take advantage of our [autonomous driving platform Annieway](#) to develop novel challenging real-world computer vision benchmarks. Our tasks of interest are: stereo, optical flow, visual odometry, 3D object detection and 3D tracking. For this purpose, we equipped a standard station wagon with two high-resolution color and grayscale video cameras. Accurate ground truth is provided by a Velodyne laser scanner and a GPS localization system. Our datasets are captured by driving around the mid-size city of [Karlsruhe](#), in rural areas and on highways. Up to 15 cars and 30 pedestrians are visible per image. Besides providing all data in raw format, we extract benchmarks for each task. For each of our benchmarks, we also provide an evaluation metric and this evaluation website. Preliminary experiments show that methods ranking high on established benchmarks such as [Middlebury](#) perform below average when being moved outside the laboratory to the real world. Our goal is to reduce this bias and complement existing benchmarks by providing real-world benchmarks with novel difficulties to the community.



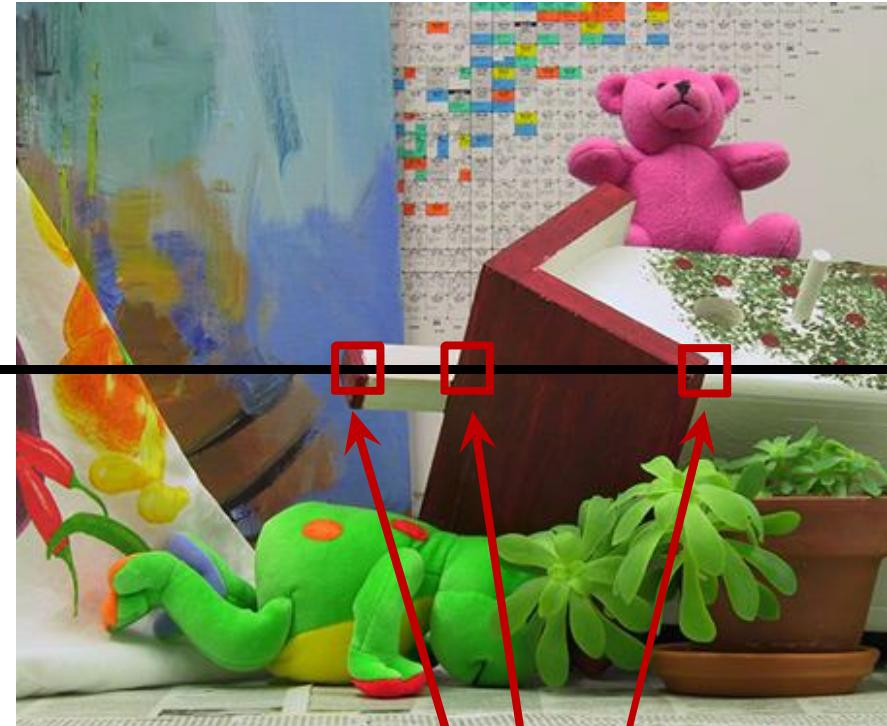
To get started, grab a cup of your favorite beverage and watch our video trailer (5 minutes):



This video: [in high-resolution \(720 MB\) at youtube](#)

<http://www.cvlibs.net/datasets/kitti/index.php>

Вычисление карты диспаритета



Какой диспаритет у этой
точке?



Основные предположения



Эпиполярное ограничение:
«Соответствующие точки лежат на одной строке»



- Самое надежное и обоснованное из всех предположений!
- Зависит от точности калибровки

Основные предположения



Соответствие цветов:
«Цвета соответствующих точек совпадают»

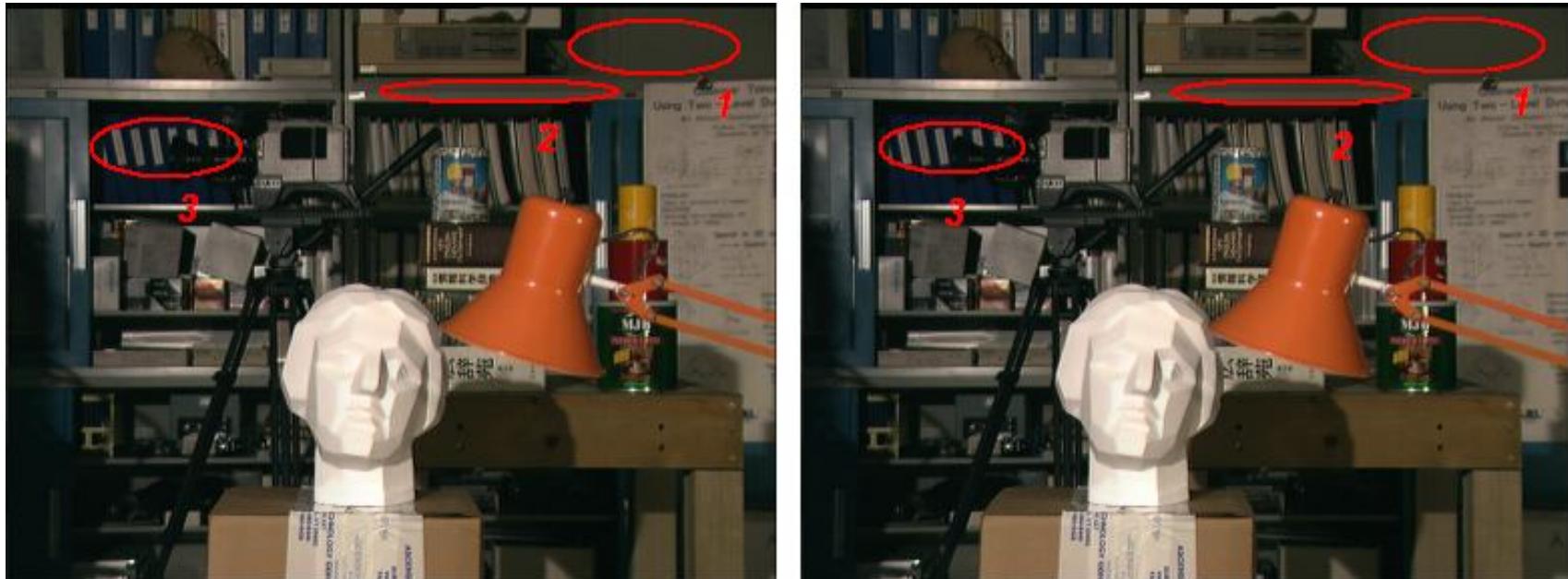


Может нарушаться (шум камеры, изменение освещенности, погрешности сэмплирования и т.д.)



Основные трудности

«Плохо» текстурированные области



1 – Однородная область

2 – Текстура неизменна в горизонтальном направлении

3 – Повторяющаяся текстура

Основные трудности



«Плохо» текстурированные области



Отсутствие текстуры
порождает ошибки

Пример: M. Bleyer

Основные трудности



Перекрытия

Некоторые пиксели одного изображения могут быть не видны (перекрыты) на другом.

«Невидимые» области называют областями перекрытия

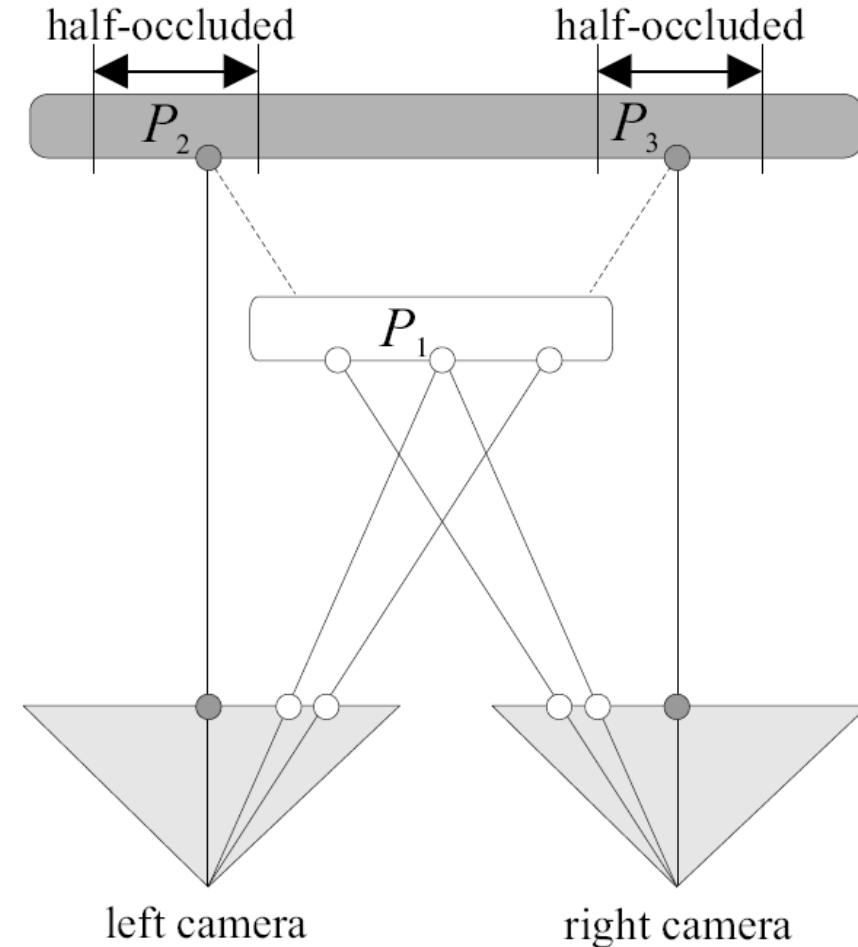
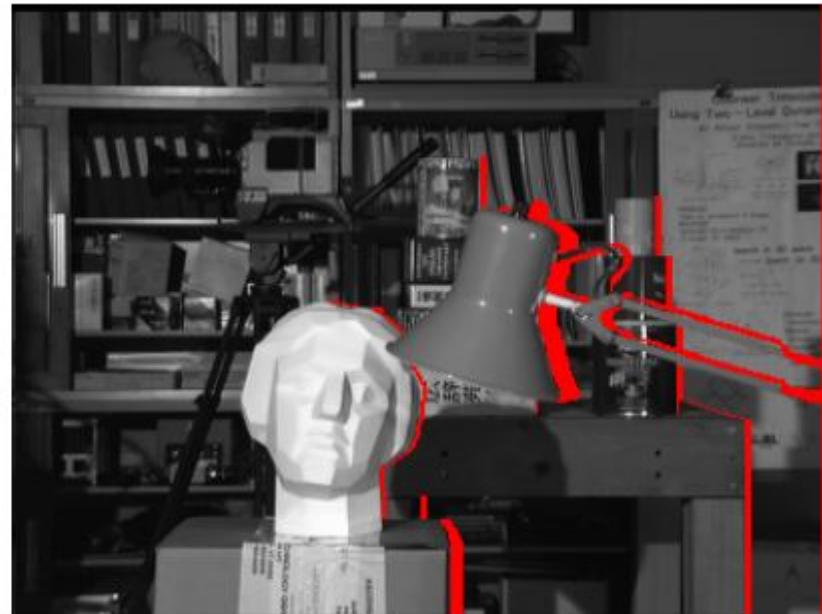


Рисунок: M. Bleyer

Основные трудности



Перекрытия



Пример областей перекрытия (выделены красным)

Рисунок: M. Bleyer

Основные предположения



Только лишь эпиполярного ограничения и предположения о соответствии цветов недостаточно, т.к. задача остается некорректно поставленной:

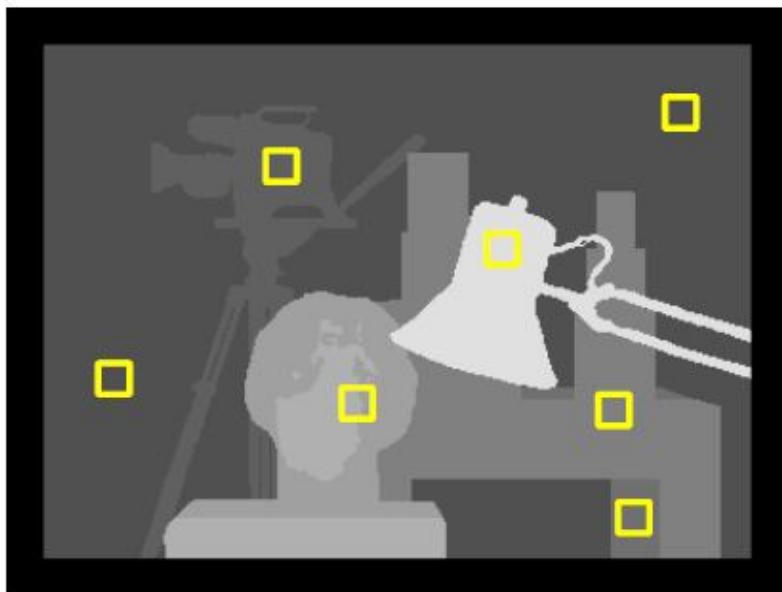
- Решение может не существовать (перекрытия)
- Решение может быть не уникально (однородные области)
- Небольшие изменения входных данных существенно влияют на результат (шум)

Нужны дополнительные предположения!

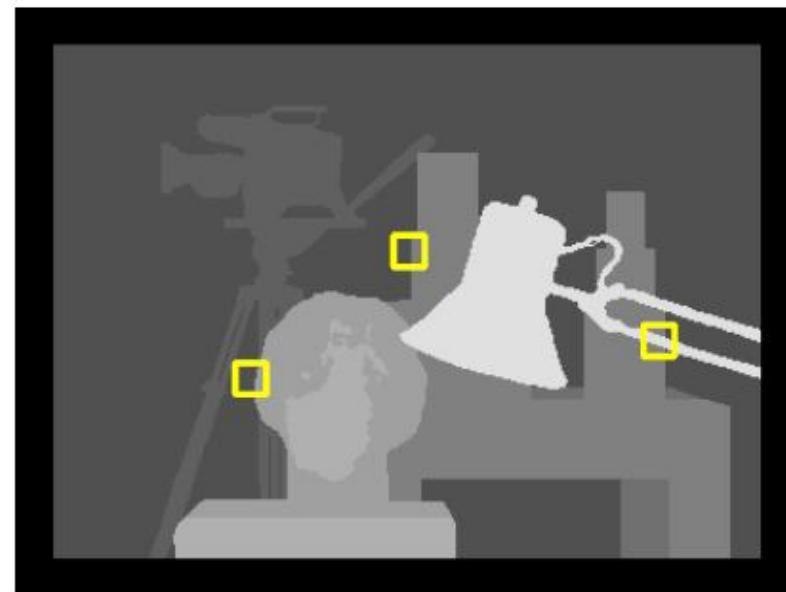
Основные предположения



Гладкость: «Искомая карта диспаритета является кусочно-гладкой»



Гладкие области



Разрывы на краях объектов

Основные предположения



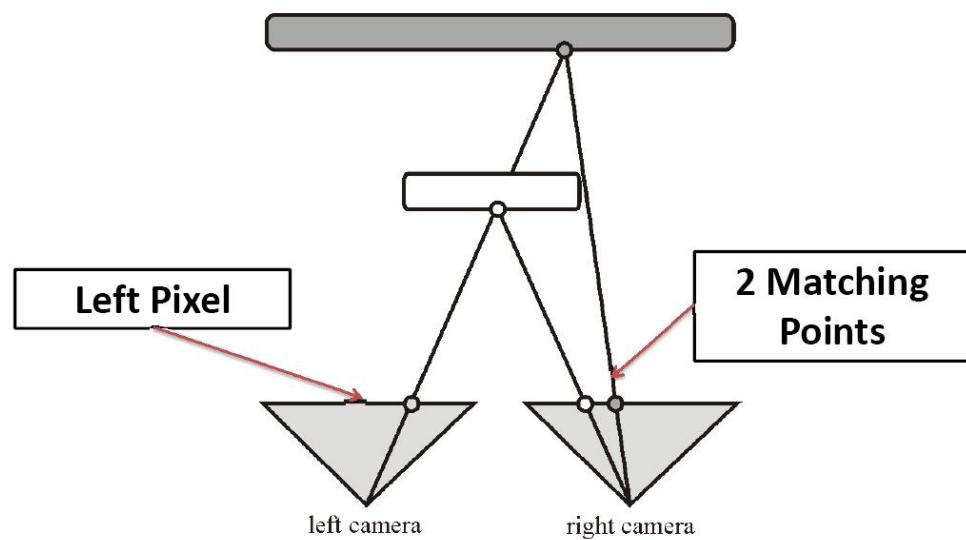
Гладкость: «Искомая карта диспаритета является кусочно-гладкой»

- Нарушается при наличии в сцене большого числа тонких объектов (ветки деревьев, волосы и т.п.)

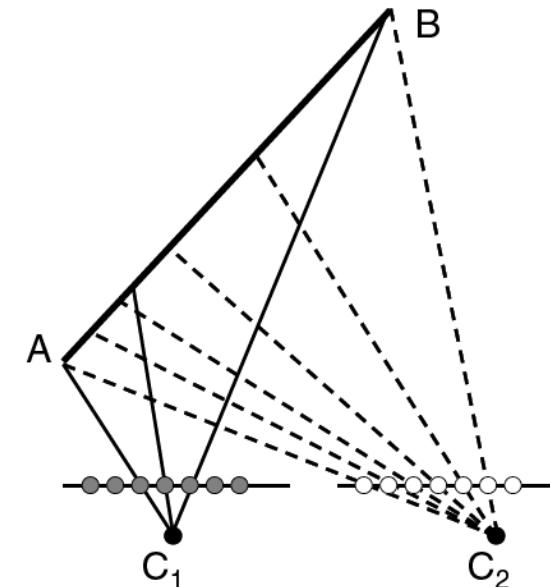
Основные предположения



- **Уникальность:** «Каждая точка одного изображения имеет не более одного соответствия на другом»
- Нарушается при наличии в сцене прозрачных объектов и горизонтально наклоненных плоскостей



Прозрачный объект

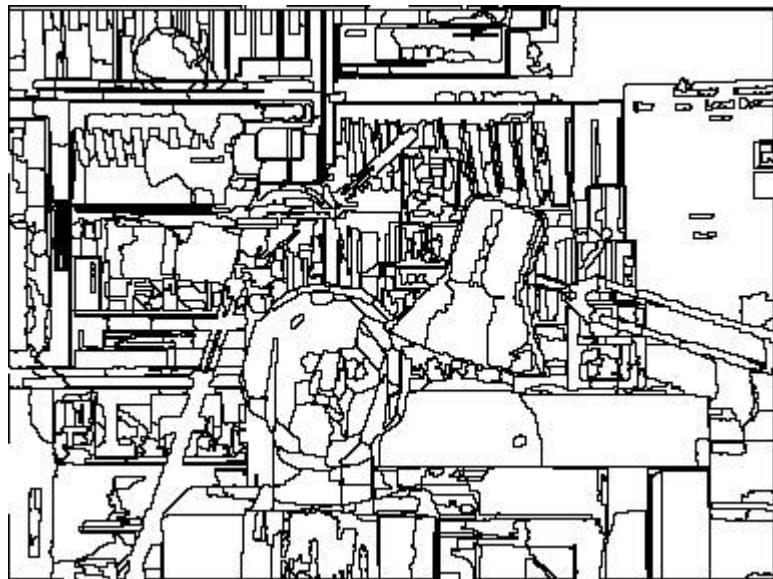


Горизонтально наклоненная
плоскость

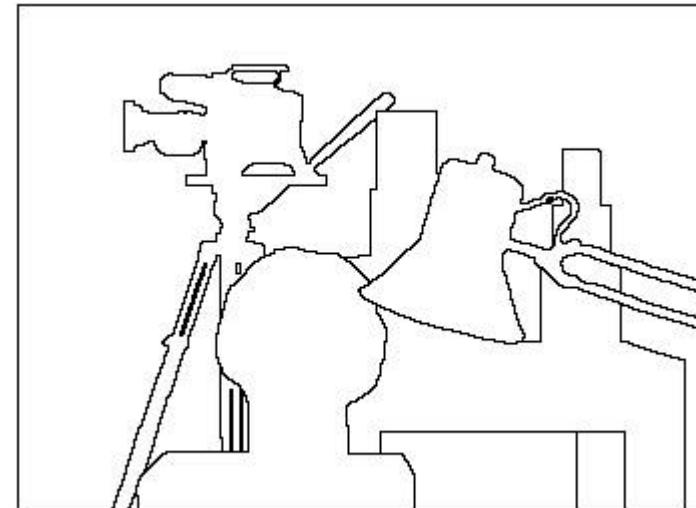
Основные предположения



Предположение о сегментации: «Области разрыва диспаритета совпадают с краями на изображении»



Границы цветовых сегментов



Границы диспаритета

Может нарушаться (ничто не гарантирует, что граница объекта не попадет внутрь цветового сегмента)

План



- Введение
- **Локальные методы**
- Глобальные методы
- Использование сегментации

Локальные методы

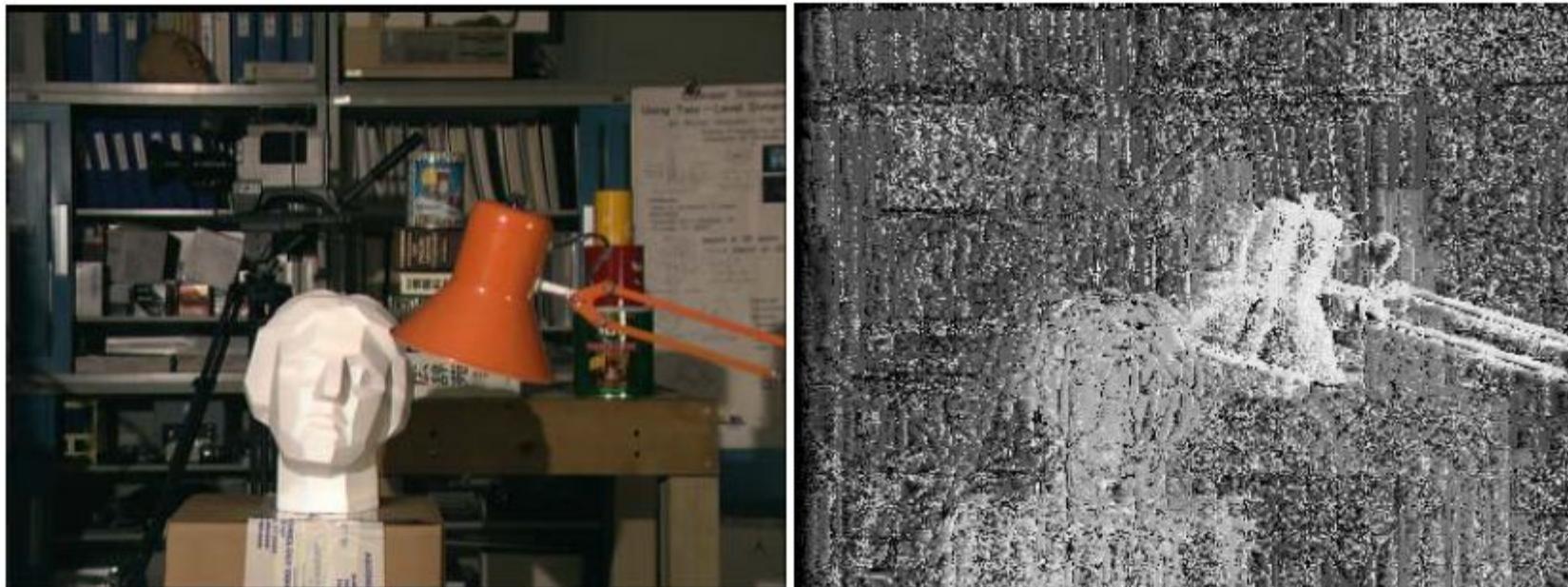


- Локальные, т.к. диспаритет в каждой точке зависит только от ее локальной окрестности
- Используют стратегию WTA (Winner-Take-All), т.е. в каждой точке «побеждает» соответствие с наименьшей «стоимостью»

Наивный алгоритм



- Стоимость соответствия – разность интенсивностей пикселей
- Результат – слишком много шума



Пример работы наивного алгоритма

Пример: M. Bleyer

Наивный алгоритм



Проблема: Слишком велика
неоднозначность сопоставления

Решение: Учитывать окрестность

Учет окрестности



- Стоимость соответствия – SAD, SSD, NCC по окну вокруг пикселя

$$d_p = \arg \min_{0 \leq d \leq d_{\max}} \sum_{q \in W_p} c(q, q - d)$$

где:

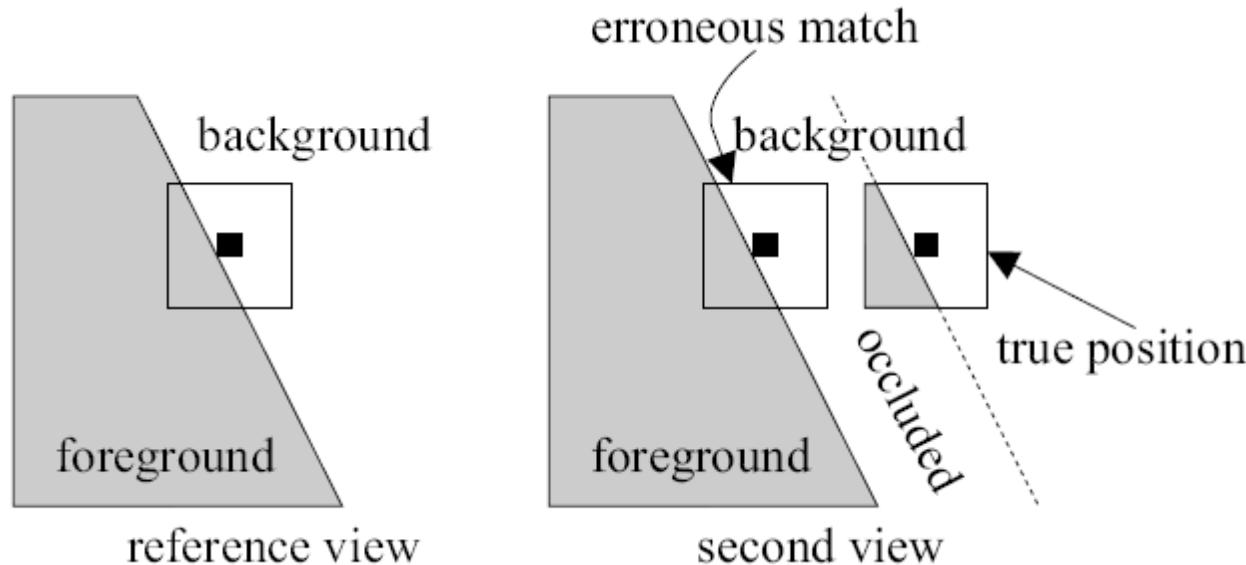
- d_p – искомый диспаритет в пикселе p ,
 $c(p, q)$ – функция стоимости,
 W_p – окно вокруг пикселя p ,
 d_{\max} – максимально возможный диспаритет

- Возможна очень эффективная реализация
 - Метод «скользящего окна»

Проблема выбора размера окна



- Маленькие окна – недостаточно текстуры
- Большие окна – эффект «раздувания» объектов переднего плана (foreground fattening) при однотонных фонах



Эффект «foreground fattening»

Проблема выбора размера окна



- Маленькие окна – недостаточно текстуры
- Большие окна – эффект «раздувания» объектов переднего плана (foreground fattening)



Ground truth



Окно 3x3



Окно 21x21

Проблема на границах объектов



- Сопоставление окон неявным образом реализует предположение о гладкости
 - Все пиксели внутри окна полагаются имеющими одинаковый диспаритет, что, вообще говоря, неверно
- Выход – исключить границы из рассмотрения с помощью адаптивных окон
 - Окна с адаптивной структурой
 - Окна с адаптивными весами

Окна с адаптивной структурой



Пробуются 9 окон, по-разному расположенных относительно рассматриваемого пикселя

Выбирается окно, дающее наименьшую стоимость

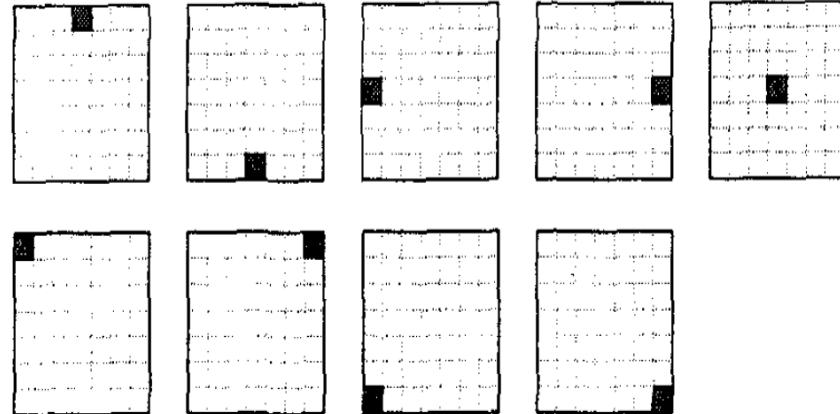


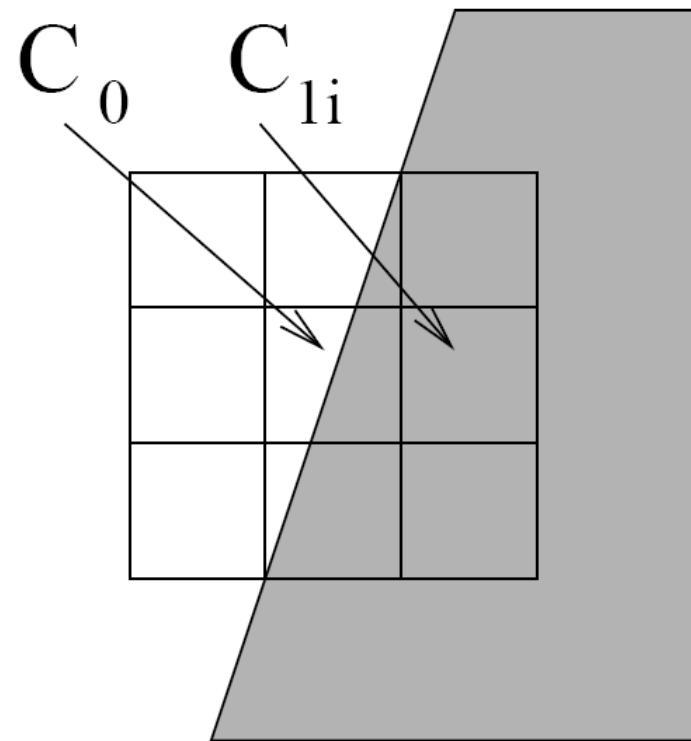
Figure 1. The nine asymmetric correlation windows. The pixel for which disparity is computed is highlighted.

Окна с адаптивной структурой



Окно делится на 9 подобластей

При сопоставлении учитываются лишь стоимости по 5 «лучшим» подобластям



H. Hirschmeuller, P. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *IJCV*, 2002.

Окна с адаптивными весами



- Каждая точка внутри окна берется с определенным весом, отражающим вероятность того, что ее диспаритет и искомый диспаритет рассматриваемого пикселя совпадают:

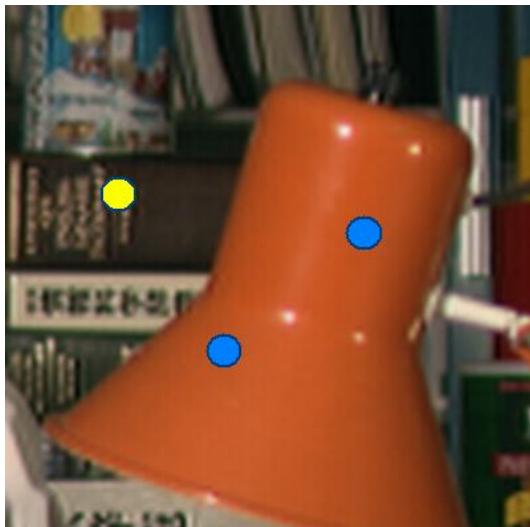
$$d_p = \arg \min_{0 \leq d \leq d_{\max}} \sum_{q \in W_p} w(p, q) \cdot c(q, q - d)$$

- Главный вопрос – как вычислить вес $w(p, q)$?

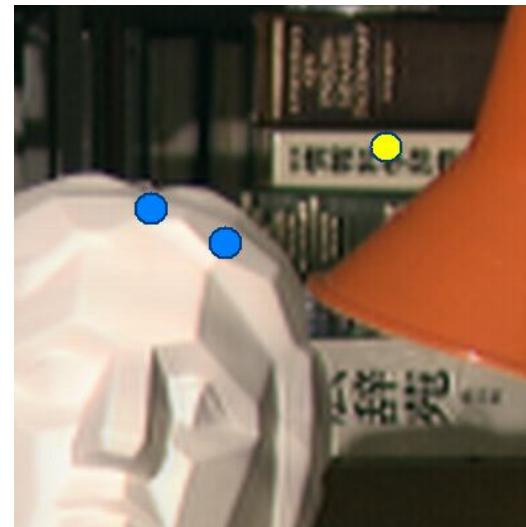
Вычисление веса



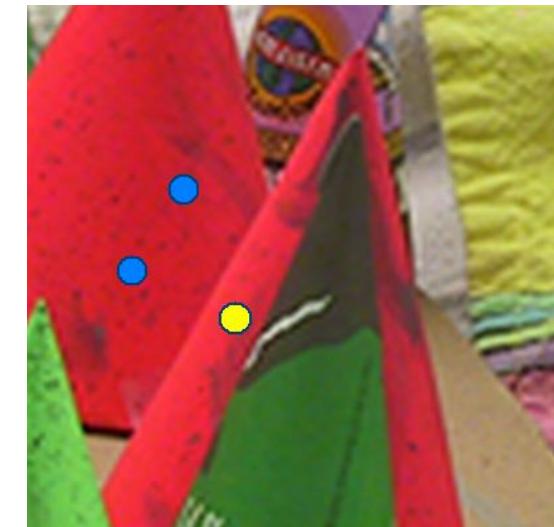
- Эвристические подсказки
 - Цвет
 - Расстояние
 - Границы



Цвет

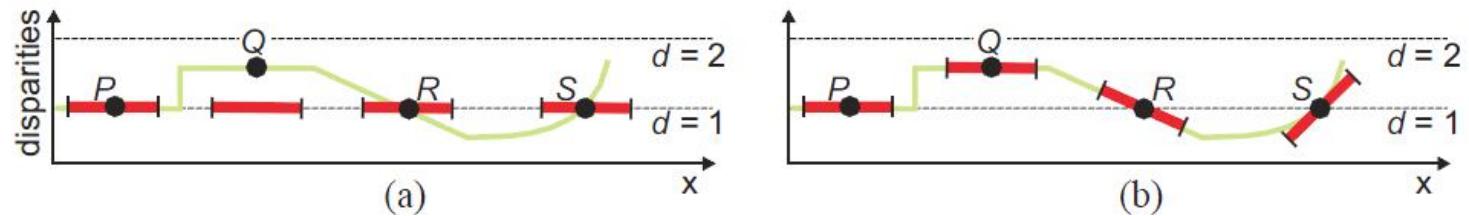


Расстояние



Граница

PatchMatch



- Вычисление качества сопоставления через небольшие фрагменты поверхности
- Точка пикселя может лежать на одной из набора поверхностей в сцене
- Рандомизированная схема PatchMatch

M. Bleyer, C. Rhemann, C. Rother: "PatchMatch Stereo - Stereo Matching with Slanted Support Windows"; BMVC 2011

Схема PatchMatch



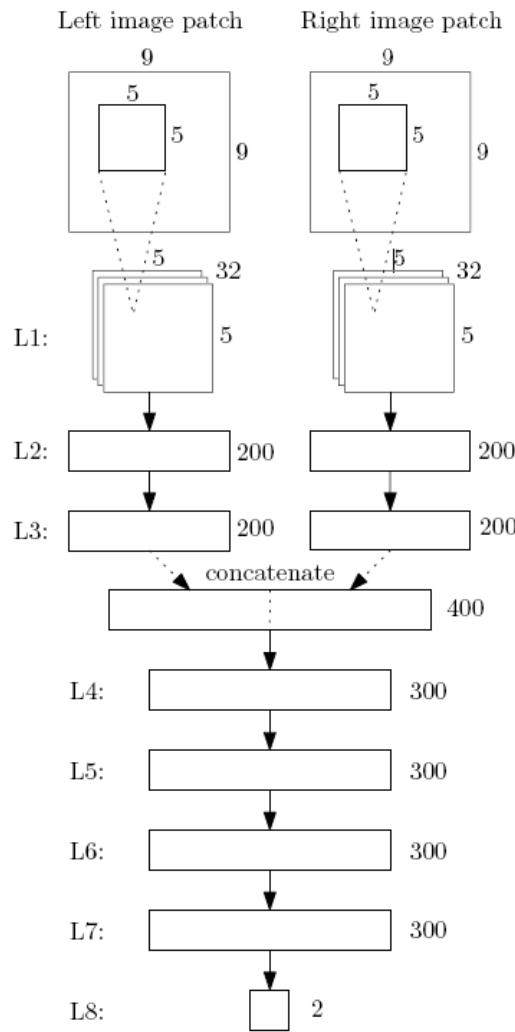
1. Инициализация алгоритма
 - Применяем какой-то метод стерео
 - Выделяем в карте множество плоскостей
2. Инициализация фрагментов
 - Каждому пикселью назначаем случайную плоскость из списка
3. Распространение гипотез
 - Сканируем изображение сверху вниз, справо налево
 - Для каждого пикселя проверяем все гипотезы соседей и оставляем наилучшую
4. Уточнение фрагментов
 - Уточняем параметры плоскостей для каждого фрагмента
 - Повторяем 3-4 до сходимости

Адаптивные окна. Резюме



- Достоинство: гибкость
- Недостаток: скорость
- Развитие локальных методов на сегодняшний день идет именно в направлении адаптивных окон

Обучение с помощью CNN



- Собираем обучающую выборку $\langle \mathcal{P}_{9 \times 9}^L(p), \mathcal{P}_{9 \times 9}^R(q) \rangle$
 - 1 положительный и 1 отрицательный пример на 1 пиксель с известным диспаритетом

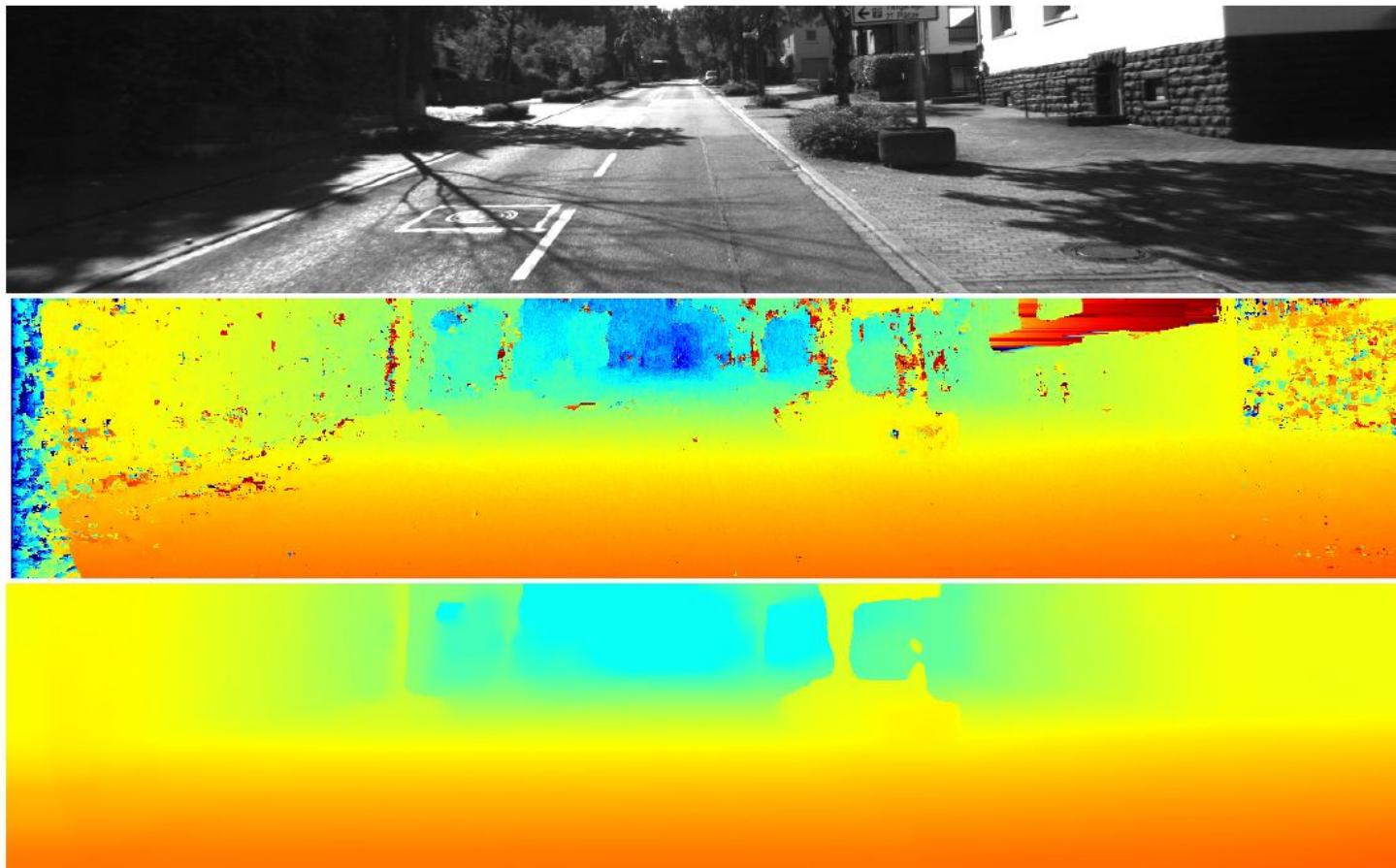
$$\mathbf{q} = (x - d + o_{\text{neg}}, y) \in \{-N_{\text{hi}}, \dots, -N_{\text{lo}}, N_{\text{lo}}, \dots, N_{\text{hi}}\}$$

- Положительные примеры:

$$\mathbf{q} = (x - d + o_{\text{pos}}, y) \in \{-P_{\text{hi}}, \dots, P_{\text{hi}}\}$$

1

Пример результата



- Ошибка 14.7% для winner-take-all стратегии
- Ошибка 2.61% при использовании метрики внутри semi-global стратегии (идею рассмотрим дальше)

Борьба с перекрытиями

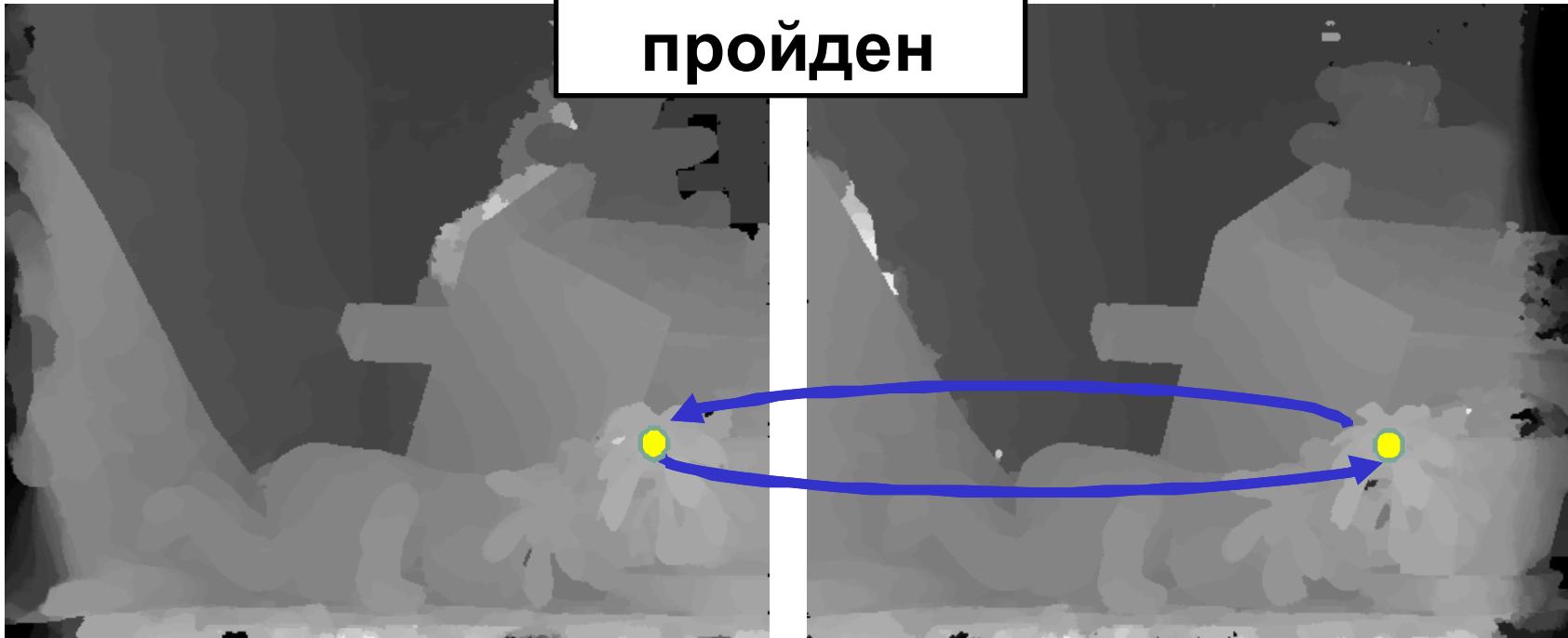


- Распространенное средство – кросс-проверка (cross-validation, consistency check)
- Идея кросс-проверки:
 - Возьмем поочередно каждое из изображений за базовое и проверим на согласованность две карты диспаритета
 - Рассогласованные пиксели отнесем к области перекрытия

Кросс-проверка



Тест
пройден



Карта диспаритета (левое
изображение)

Карта диспаритета (правое
изображение)

Кросс-проверка



Тест не
пройден



Карта диспаритета (левое
изображение)



Карта диспаритета (правое
изображение)



Кросс-проверка

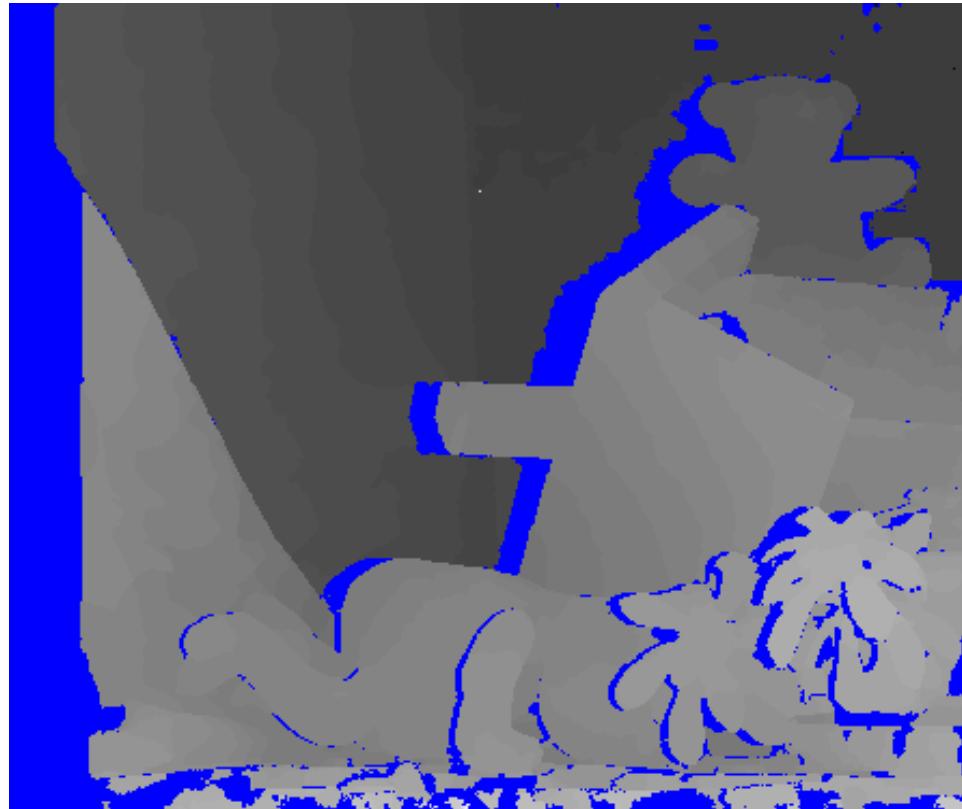
- Кросс-проверка реализует предположение об уникальности:



Диспаритет одного из двух пикселей слева будет обнулен в результате кросс-проверки

- В результате кросс-проверки все пиксели имеют либо ровно одно, либо ни одного соответствия

Борьба с перекрытиями



Пиксели, не прошедшие кросс-проверку
(выделены синим)

Пример: M. Bleyer

Борьба с перекрытиями



«Пустые» области необходимо
заполнить

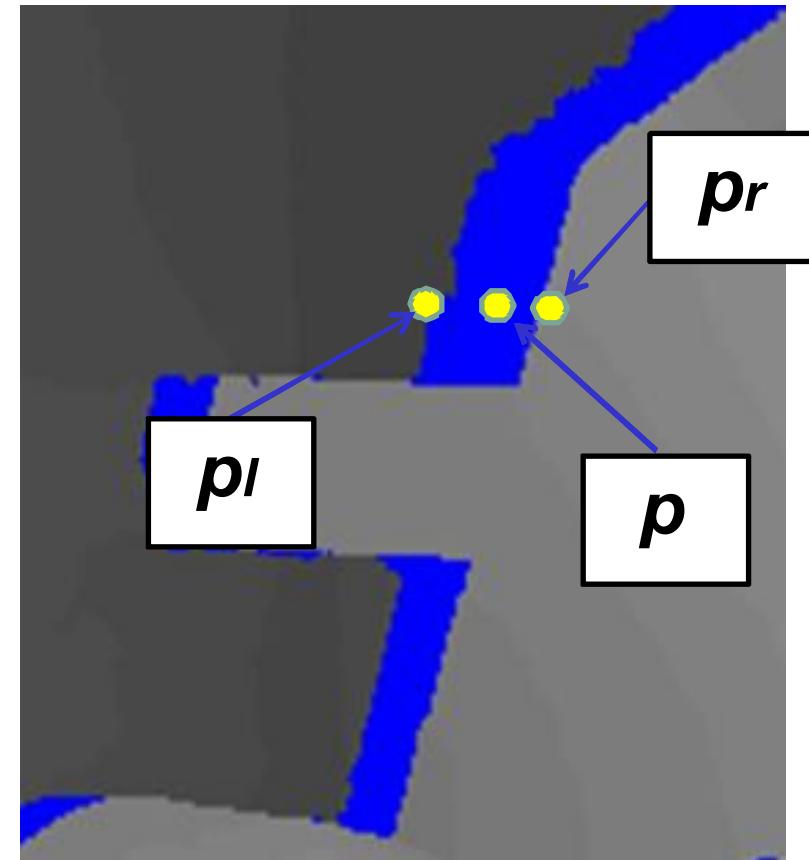
Заполнение областей перекрытия



Найдем ближайшие к p «значащие» пиксели слева и справа вдоль строки

Присвоим p наименьший из двух диспаритетов, т.к. перекрытия обычно располагаются в области фона:

$$d_p = \min(d_{p_l}, d_{p_r})$$



Заполнение областей перекрытия



После заполнения карту диспаритета можно сгладить, например, медианным фильтром



До сглаживания



После сглаживания

План



- Введение
- Локальные методы
- **Глобальные методы**
- Использование сегментации

Глобальные методы



- Глобальные, т.к. диспаритет в каждой точке вычисляется при помощи некоторой глобальной процедуры оптимизации, т.е. зависит не только от локальной окрестности
- Как правило, формулируются в терминах разметки графа и минимизации энергии

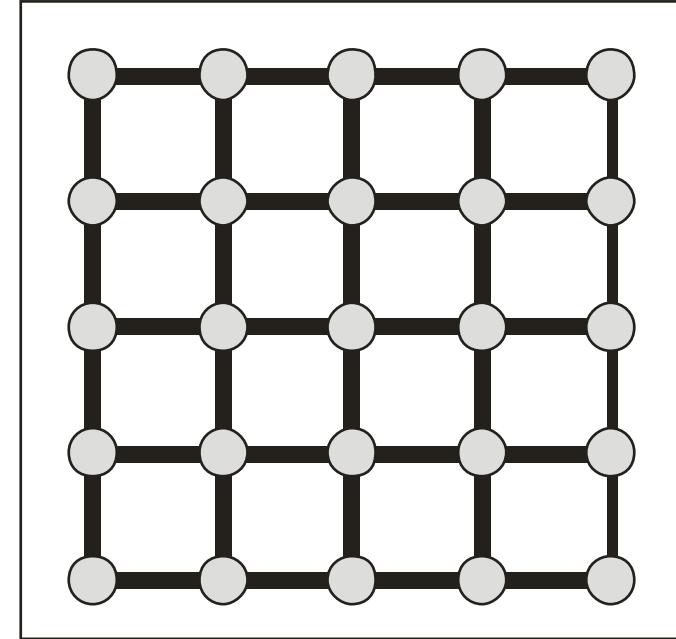
Глобальные методы



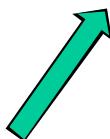
Необходимо найти разметку D , минимизирующую функцию энергии $E(D)$.

Метки – значения диспаритета.

Граф – решетка, узлы – пиксели.



$$E(D) = E_{data}(D) + E_{smooth}(D)$$



Соответствие цветов



Гладкость

Унарный потенциал



$$E_{data}(D) = \sum_{p \in I} m(p, p - d_p)$$

где:

$m(p, q)$ – мера цветового сходства между пикселием p одного изображения и пикселием q другого

Например:

$$m(p, q) = |I_p - I_q|$$

$$m(p, q) = \min(|I_p - I_q|, k)$$

Парный потенциал



$$E_{smooth}(D) = \sum_{\langle p, q \rangle \in N} s(d_p, d_q)$$

где:

N – система соседства (множество всех пар пикселей-соседей),

$s(d_p, d_q)$ – функция штрафа за нарушение гладкости

Например:

$$s(d_p, d_q) = |d_p - d_q| \cdot P \quad \text{или} \quad s(d_p, d_q) = \begin{cases} 0, & d_p = d_q \\ P, & d_p \neq d_q \end{cases}$$

Линейная модель



Модель Поттса

Проблема сохранения границ



Предположим, необходимо восстановить диспаритет в области границы между объектами, как показано на рисунке

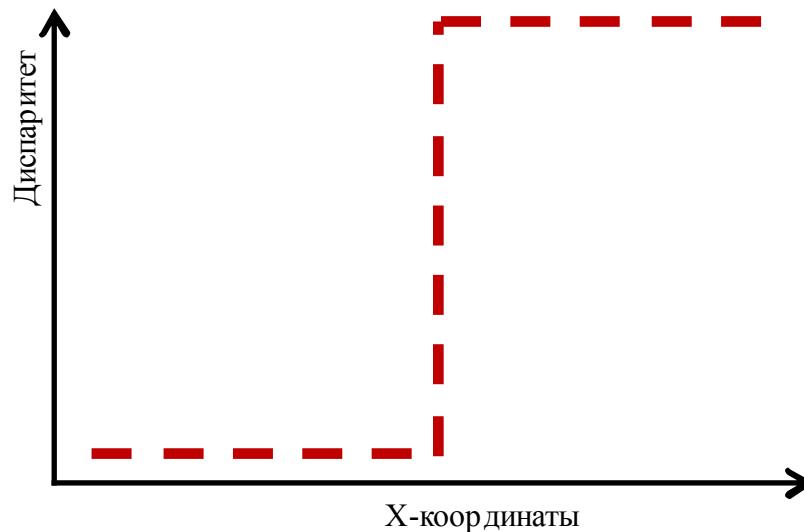


Рисунок: M. Bleyer

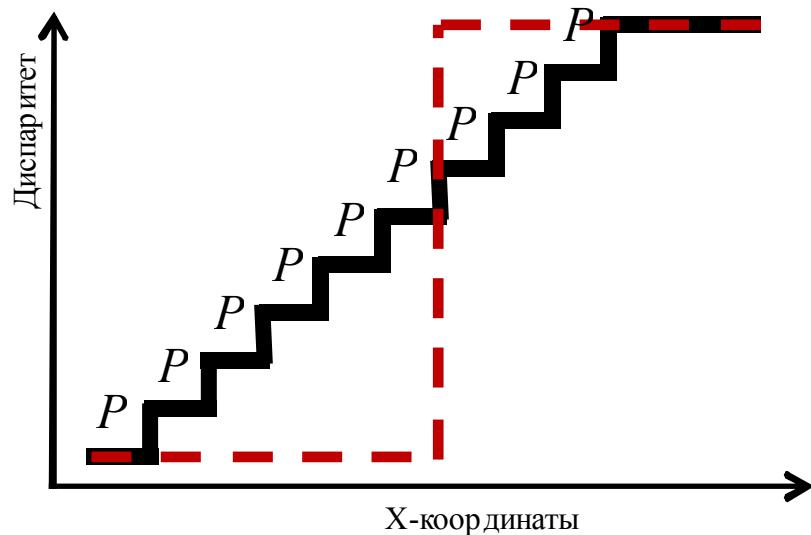
Проблема сохранения границ



Случай линейной модели:

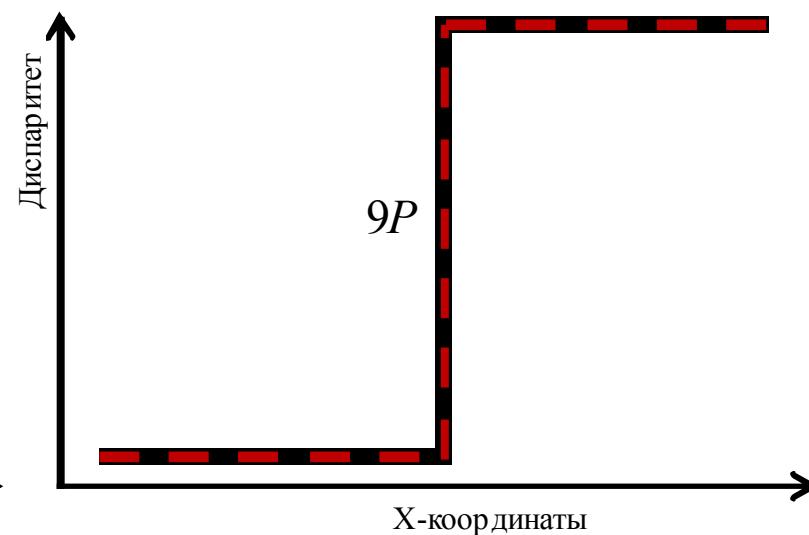
$$s(d_p, d_q) = |d_p - d_q| \cdot P$$

Вклад в энергию:
 $9P$



(неверное решение)

Вклад в энергию:
 $9P$



(верное решение)

Линейная модель не поощряет резких разрывов диспаритета.
Она чересчур сглаживает решение

Рисунки: M. Bleyer

Проблема сохранения границ



Случай линейной модели:

$$s(d_p, d_q) = |d_p - d_q| \cdot P$$



Пример результатов для линейной модели. Справа показан
увеличененный фрагмент

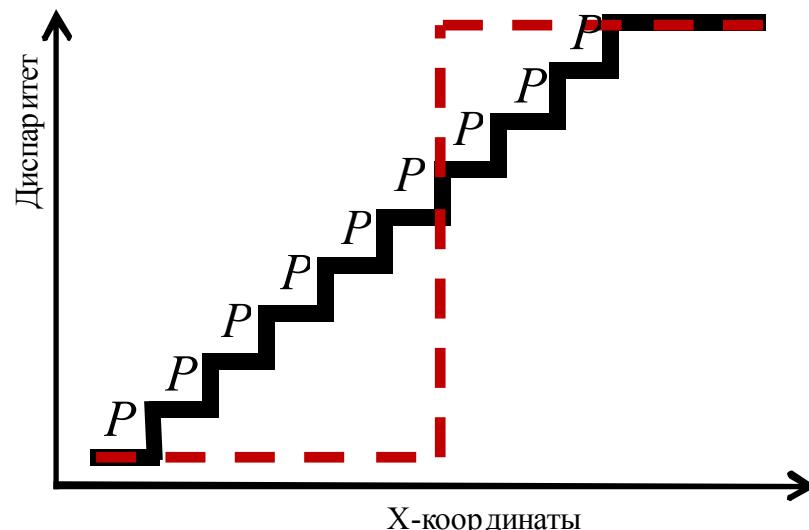
Проблема сохранения границ



Случай модели Поттса:

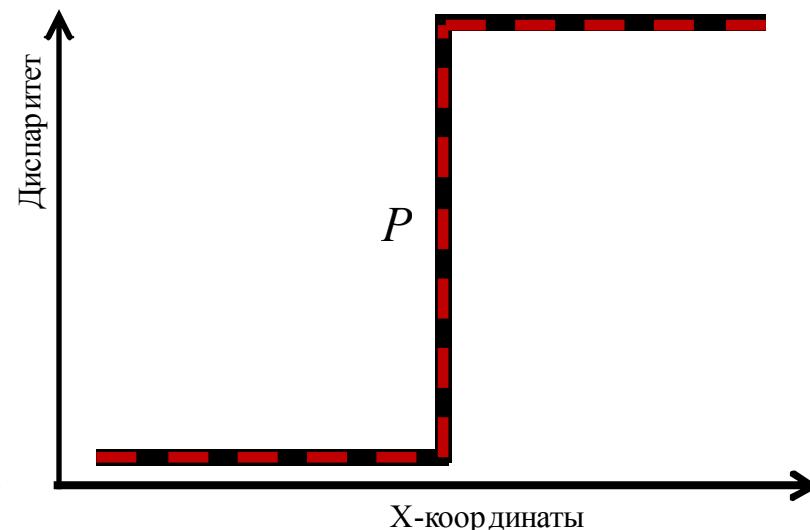
$$s(d_p, d_q) = \begin{cases} 0, & d_p = d_q \\ P, & d_p \neq d_q \end{cases}$$

Вклад в энергию: **9Р**



(неверное решение)

Вклад в энергию: Р



(верное решение)

- Модель Поттса не препятствует резким разрывам диспаритета.
 - Такие модели называют сохраняющими разрывы (discontinuity preserving) – как и в оптическом потоке

Рисунки: M. Bleyer

Проблема сохранения границ



Случай модели Поттса:

$$s(d_p, d_q) = \begin{cases} 0, & d_p = d_q \\ P, & d_p \neq d_q \end{cases}$$



Пример результатов для модели Поттса. Справа показан
увеличененный фрагмент

Другие модели



- Модифицированная модель Поттса:

$$s(d_p, d_q) = \begin{cases} 0, & d_p = d_q \\ P_1, & |d_p - d_q| = 1 \\ P_2, & |d_p - d_q| > 1 \end{cases}$$

- Усеченная линейная модель:

$$s(d_p, d_q) = \min(|d_p - d_q|, k) \cdot P$$

- Модель 2-го порядка [Woodford, 2008]:

$$s(d_p, d_q, d_r) = \min(|d_p - 2d_q + d_r|, k) \cdot P$$

Минимизация энергии



- Плотное стерео – задача многоклассовой разметки
- Эффективное решение на графе общего вида существует лишь для выпуклых относительно $|d_p - d_q|$ парных потенциалов [Ishikava, 2003]
- Но необходимо использовать модели, сохраняющие границы, а они невыпуклы относительно $|d_p - d_q|$
- Задача становится NP-полной
- Необходимы приближённые алгоритмы
 - Fusion move, Loopy belief propagation, TRW

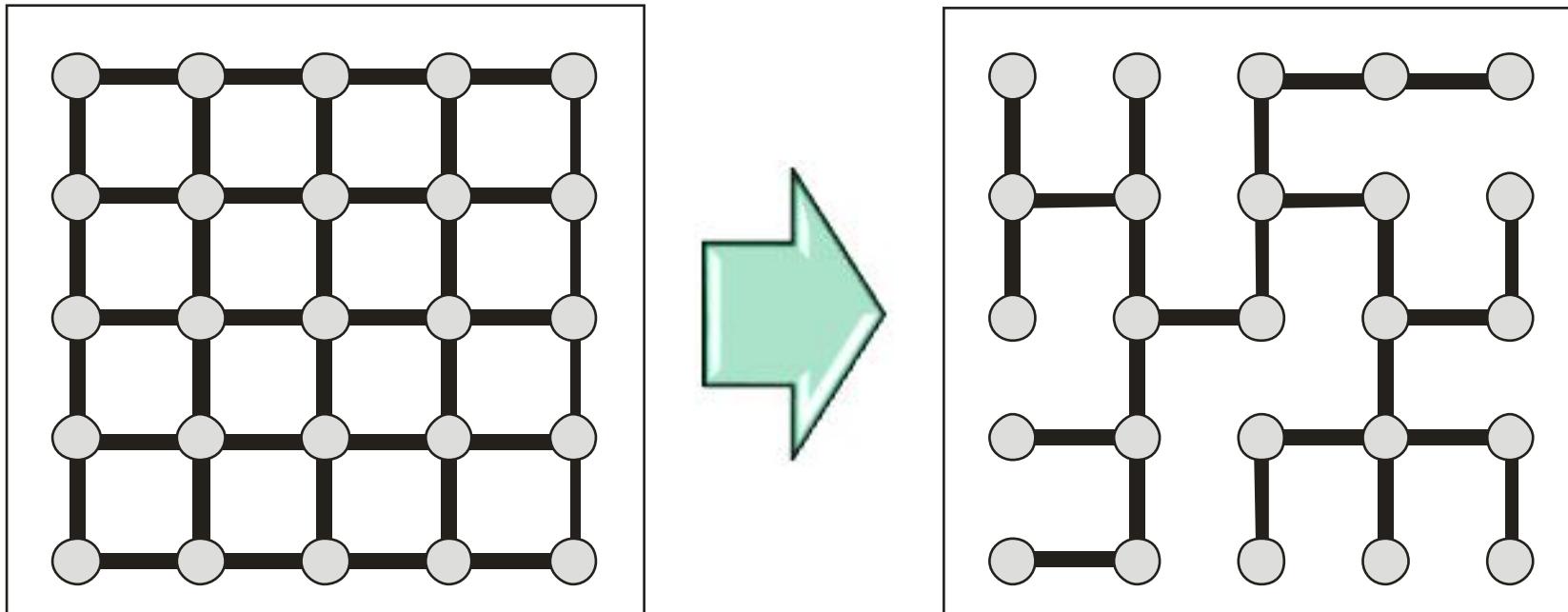
Минимизация энергии



графе общего вида существует

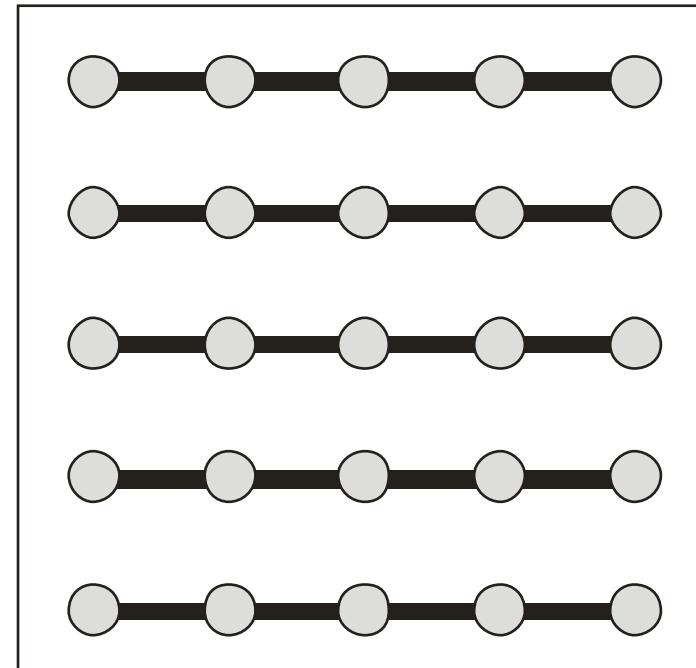
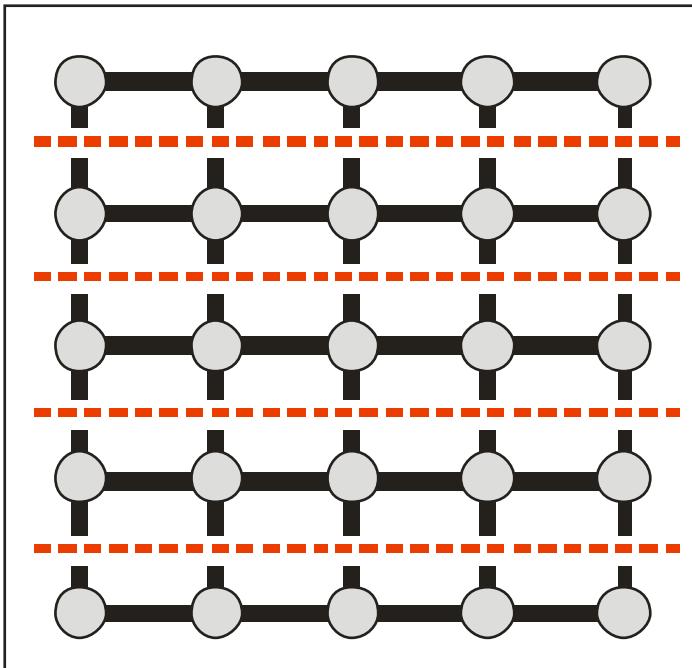
Альтернативный подход
заключается в уходе от графов
общего вида

Переход к деревьям



- Отсутствие циклов позволяет использовать метод динамического программирования
 - Глобальный минимум, произвольная энергия, высокая скорость работы
- Главный вопрос – какие ребра убирать?

Алгоритм Scanline Optimization



- Удаляются все вертикальные ребра
- Так поступали в первых подобных алгоритмах

Алгоритм Scanline Optimization



Очевидная проблема – рассогласованность строк между собой
(horizontal streaking)

Пример: M. Bleyer

Алгоритм на основе MST



Идея – не форсировать гладкость между пикселями сильно разного цвета

Каждому ребру (паре пикселей p и q) присваивается вес:

$$w(p, q) = |I(p) - I(q)|$$

Строится минимальное покрывающее дерево (Minimum Spanning Tree, MST)

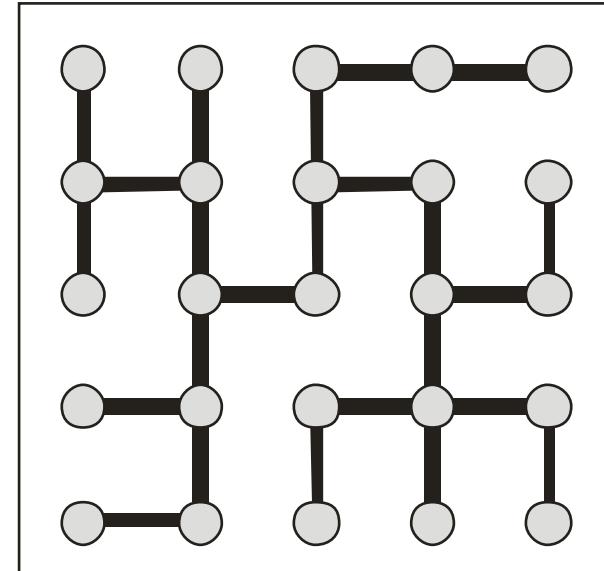


Рисунок: M. Bleyer

Алгоритм на основе MST



Лучше, чем scanline optimization, но некоторая рассогласованность остается

Алгоритм Semi-Global Matching



В каждом пикселе строится свое дерево

Оптимизация производится вдоль лучей, исходящих из пикселя

Подход не совсем глобальный, но и не локальный

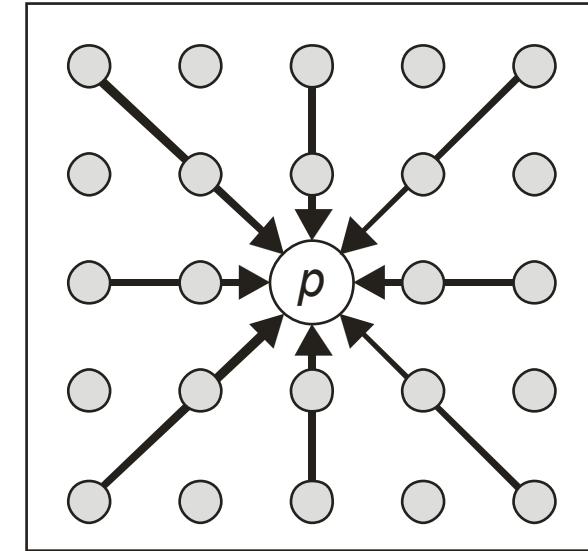


Рисунок: M. Bleyer

Алгоритм Semi-Global Matching



- Рассогласованности нет, но есть «изолированные» пиксели

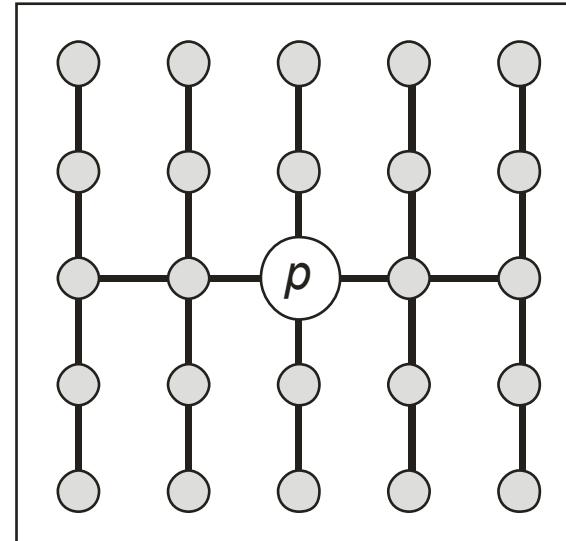
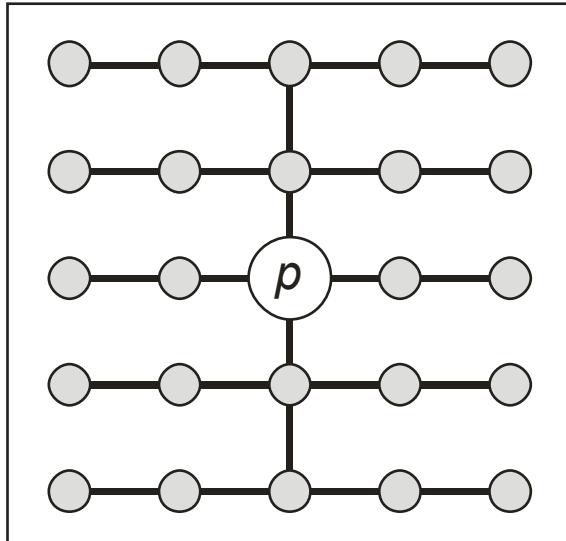
Алгоритм Semi-Global Matching



Хорошо работает на аэрофотоснимках

Примеры: <http://www.robotic.de/Heiko.Hirschmueller/>

Алгоритм Simple Tree



Рисунки: M. Bleyer

В каждом пикселе строятся два дерева, совместно покрывающих все изображение. Алгоритм глобальный и лишен недостатка SGM.

M. Bleyer, M. Gelautz. Simple but Effective Tree Structures for Dynamic Programming-based Stereo Matching. VISAPP, 2008.

Алгоритм Simple Tree



Пример работы

Борьба с перекрытиями



- Модифицируем унарный потенциал:

$$E_{data}(D) = \sum_{p \in I} m(p, d_p)$$



$$E_{data}(D) = \sum_{p \in I} (m(p, d_p)(1 - O(p)) + P_{occ} \cdot O(p))$$

где:

$O(p)$ – функция, возвращающая 1, если пиксель принадлежит области перекрытия, иначе 0

P_{occ} – штраф за перекрытие

Борьба с перекрытиями

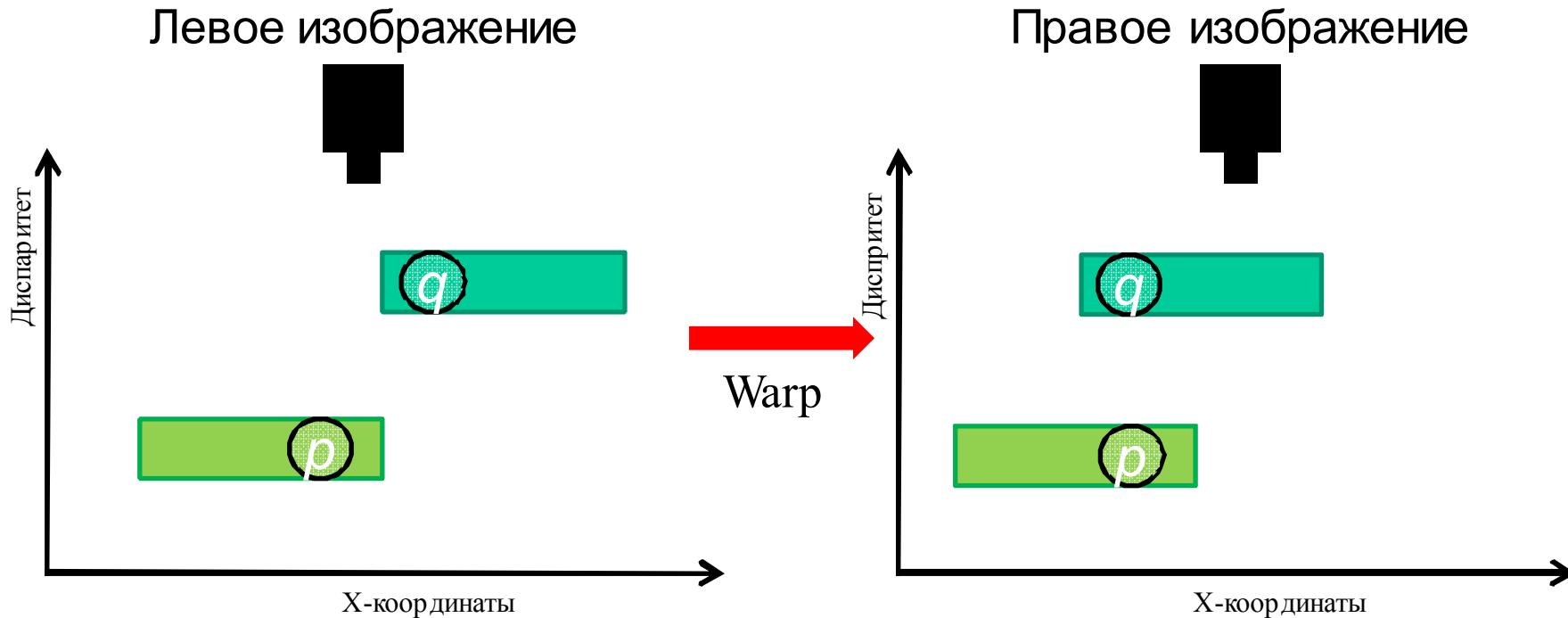


Как задать функцию $O(p)$?

Ограничение видимости



Из двух пикселей p и q виден тот, который ближе к камере. Пиксель p на левом изображении принадлежит области перекрытия.





- Унарный потенциал:

$$E_{data}(D) = \sum_{p \in I} (m(p, d_p)(1 - O(p)) + P_{occ} \cdot O(p))$$

- Зададим $O(p)$ при помощи ограничения видимости (visibility constraint):

$$O(p) = \begin{cases} 1, & \text{если } \exists q \in I : p - d_p = q - d_q \text{ и } d_p < d_q \\ 0, & \text{иначе} \end{cases}$$

Глобальные методы. Резюме



- Некоторое время были наилучшими по качеству методами
- Методы, основанные на динамическом программировании, - отличный компромисс между качеством и скоростью
- Узкое место глобальных методов – функционалы энергии, а не алгоритмы оптимизации

План



- Введение
- Локальные методы
- Глобальные методы
- **Использование сегментации**

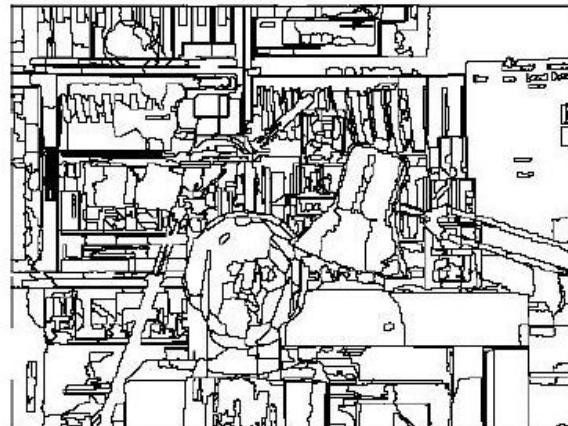
Использование сегментации



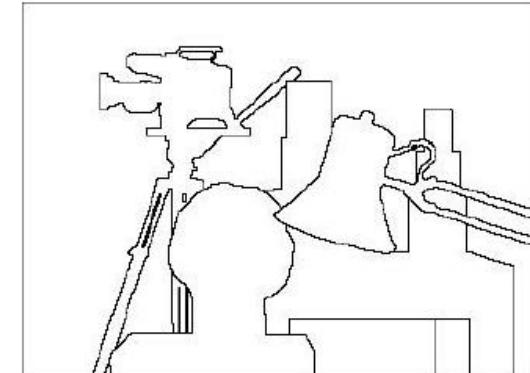
- Используется предположение о сегментации («области разрыва диспаритета совпадают с краями на изображении»)
- Для его реализации применяют пересегментацию (т.е. сегменты достаточно мелкие, «с запасом»)



Исходное изображение



Результат сегментации



Границы объектов

- Происходит переход из пиксельного пространства в пространство сегментов. Гладкость внутри сегментов форсируется
- Этот подход сейчас показывает наилучшие результаты
 - Возможно, переобучение на Middlebury

Примеры: M. Bleyer

Базовый алгоритм с сегментацией



- Пересегментация
- Инициализация решения
 - Любой локальный алгоритм на пикселях
- Апроксимация сегментов гладкими поверхностями
 - Модель: плоскость, B-сплайн
 - Средство: RANSAC, голосование и т.д.
- Уточнение разметки сегментов
 - Iterated Conditional Modes (ICM), Cooperative Optimization и др.

Использование сегментации



- Преимущества
 - Надежность в областях со слабой текстурой
 - Снижение размерности задачи (оптимизация на уровне сегментов)
- Недостатки
 - Нет защиты от нарушения предположения о сегментации
 - Сложность выбора модели, описывающей изменение диспаритета внутри сегмента
 - Проблему перекрытий все равно необходимо решать на пиксельном уровне



Stereo Evaluation

Rank	Method	Setting	Code	Out-Noc	Out-All	Avg-Noc	Avg-All	Density	Runtime	Environment	Compare
1	Displets		code	2.47 %	3.27 %	0.7 px	0.9 px	100.00 %	265 s	>8 cores @ 3.0 Ghz (Matlab + C/C++)	<input type="checkbox"/>
	F. Guney and A. Geiger: Displets: Resolving Stereo Ambiguities using Object Knowledge . Conference on Computer Vision and Pattern Recognition (CVPR) 2015.										
2	MC-CNN			2.61 %	3.84 %	0.8 px	1.0 px	100.00 %	100 s	Nvidia GTX Titan (CUDA, Lua/Torch7)	<input type="checkbox"/>
	J. Zbontar and Y. LeCun: Computing the Stereo Matching Cost with a Convolutional Neural Network . Conference on Computer Vision and Pattern Recognition (CVPR) 2015.										
3	PRSM		code	2.78 %	3.00 %	0.7 px	0.7 px	100.00 %	300 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
	C. Vogel, K. Schindler and S. Roth: 3D Scene Flow Estimation with a Piecewise Rigid Scene Model . ijcv 2015.										
4	SPS-StFL			2.83 %	3.64 %	0.8 px	0.9 px	100.00 %	35 s	1 core @ 3.5 Ghz (C/C++)	<input type="checkbox"/>
	K. Yamaguchi, D. McAllester and R. Urtasun: Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation . ECCV 2014.										
5	VC-SF			3.05 %	3.31 %	0.8 px	0.8 px	100.00 %	300 s	1 core @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
	C. Vogel, S. Roth and K. Schindler: View-Consistent 3D Scene Flow Estimation over Multiple Frames . Proceedings of European Conference on Computer Vision. Lecture Notes in Computer Science 2014.										
6	OSF		code	3.28 %	4.07 %	0.8 px	0.9 px	99.98 %	50 min	1 core @ 3.0 Ghz (Matlab + C/C++)	<input type="checkbox"/>
	M. Menze and A. Geiger: Object Scene Flow for Autonomous Vehicles . Conference on Computer Vision and Pattern Recognition (CVPR) 2015.										
7	CoR		code	3.30 %	4.10 %	0.8 px	0.9 px	100.00 %	6 s	6 cores @ 3.3 Ghz (Matlab + C/C++)	<input type="checkbox"/>
	A. Chakrabarti, Y. Xiong, S. Gortler and T. Zickler: Low-level Vision by Consensus in a Spatial Hierarchy of Regions . CVPR 2015.										
8	SPS-St		code	3.39 %	4.41 %	0.9 px	1.0 px	100.00 %	2 s	1 core @ 3.5 Ghz (C/C++)	<input type="checkbox"/>
	K. Yamaguchi, D. McAllester and R. Urtasun: Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation . ECCV 2014.										
9	PCBP-SS			3.40 %	4.72 %	0.8 px	1.0 px	100.00 %	5 min	4 cores @ 2.5 Ghz (Matlab + C/C++)	<input type="checkbox"/>
	K. Yamaguchi, D. McAllester and R. Urtasun: Robust Monocular Epipolar Flow Estimation . CVPR 2013.										
10	DDS-SS			3.83 %	4.59 %	0.9 px	1.0 px	100.00 %	1 min	1 core @ 2.5 Ghz (Matlab + C/C++)	<input type="checkbox"/>
	D. Wei, C. Liu and W. Freeman: A Data-driven Regularization Model for Stereo and Flow . 3DTV-Conference, 2014 International Conference on 2014.										
11	StereoSLIC			3.92 %	5.11 %	0.9 px	1.0 px	99.89 %	2.3 s	1 core @ 3.0 Ghz (C/C++)	<input type="checkbox"/>
	K. Yamaguchi, D. McAllester and R. Urtasun: Robust Monocular Epipolar Flow Estimation . CVPR 2013.										
12	PR-Sf-E			4.02 %	4.87 %	0.9 px	1.0 px	100.00 %	200 s	4 cores @ 3.0 Ghz (Matlab + C/C++)	<input type="checkbox"/>
	C. Vogel, S. Roth and K. Schindler: Piecewise Rigid Scene Flow . International Conference on Computer Vision (ICCV) 2013.										
13	PCBP			4.04 %	5.37 %	0.9 px	1.1 px	100.00 %	5 min	4 cores @ 2.5 Ghz (Matlab + C/C++)	<input type="checkbox"/>
	K. Yamaguchi, T. Hazan, D. McAllester and R. Urtasun: Continuous Markov Random Fields for Robust Stereo Estimation . ECCV 2012.										
14	CSPMS			4.13 %	5.92 %	1.2 px	1.6 px	100.00 %	6 s	4 cores @ 2.5 Ghz (C/C++)	<input type="checkbox"/>
	Anonymous submission										
15	MBM			4.35 %	5.43 %	1.0 px	1.1 px	100.00 %	0.2 s	1 core @ 3.0 Ghz (C/C++)	<input type="checkbox"/>
	Anonymous submission										
16	PR-Sceneflow			4.36 %	5.22 %	0.9 px	1.1 px	100.00 %	150 sec	4 core @ 3.0 Ghz (Matlab - C/C++)	<input type="checkbox"/>
	C. Vogel, S. Roth and K. Schindler: Piecewise Rigid Scene Flow . International Conference on Computer Vision (ICCV) 2013.										
17	CPB-CNN		code	4.40 %	5.72 %	1.0 px	1.2 px	100.00 %	6 s	4 cores @ 3.0 Ghz (Matlab + C/C++)	<input type="checkbox"/>

Резюме лекции



- Многое уже сделано, но задача не решена
- Раньше я говорил «главный компромисс: скорость либо качество». Now I'm not so sure.
- Требуется расширение тестовых выборок
- Явный тренд в сторону полу-глобальных алгоритмов, пересегментации, подгонки плоскостей и обучения метрик
- Что не вошло:
 - Учет цвета в стерео
 - Борьба с перепадами яркости
 - Совмещение задач стерео и матирования
 - Обучение парных потенциалов