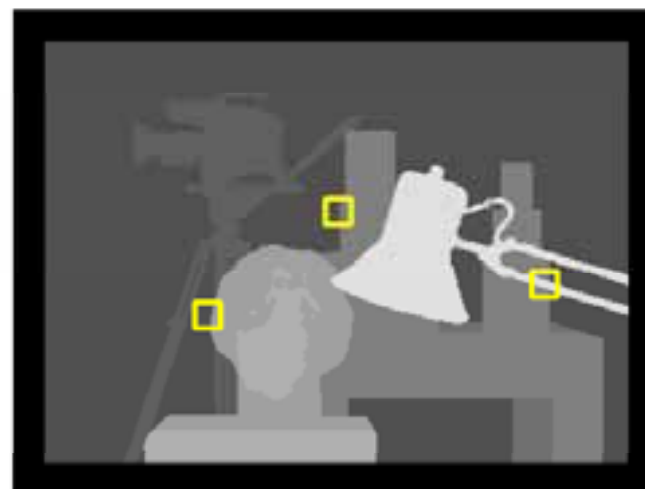
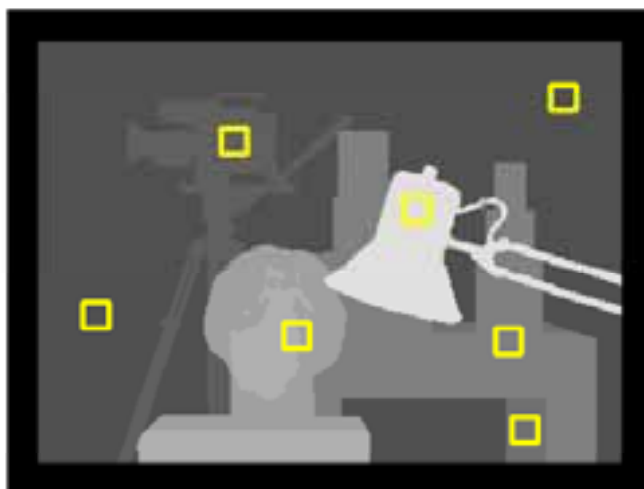




# Биноккулярное плотное стерео

---





# Благодарность

---

Презентация была подготовлена Глебом Кривовязем для спецсеминара «Компьютерное зрение»

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



# План

---

- **Введение**
- Формулировка задачи
- Трудности и предположения
- Локальные методы
- Глобальные методы
- Использование сегментации
- Резюме



# 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



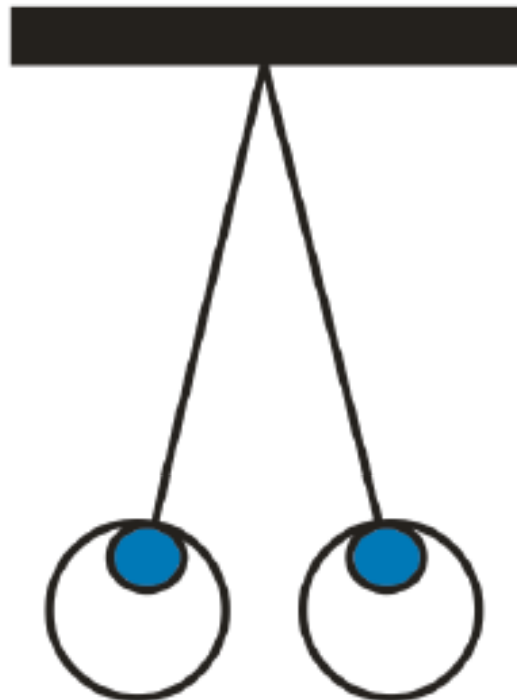
Правое изображение



# Имитация 3D

---

Так мы обычно смотрим на экран (2D):

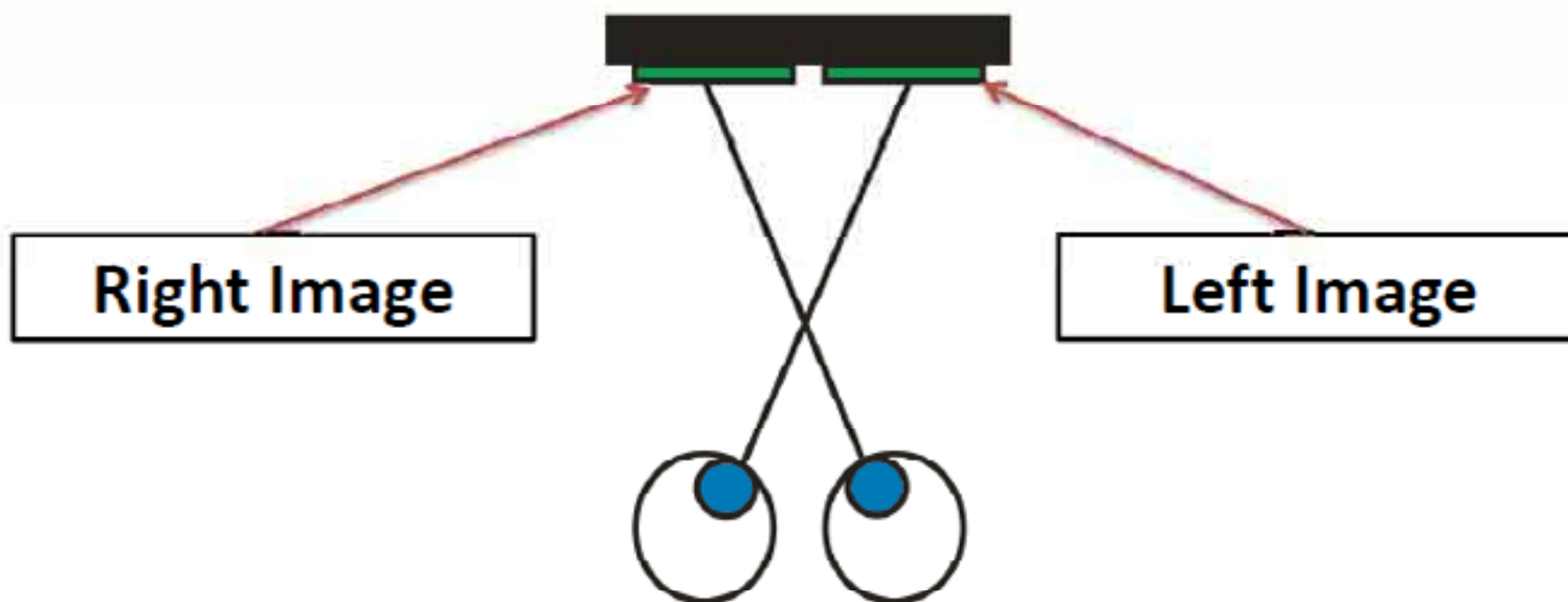




# Имитация 3D

---

А так можно увидеть 3D-картинку:





# Имитация 3D. Пример

---

3D-картинка появится посередине!



Для правого глаза

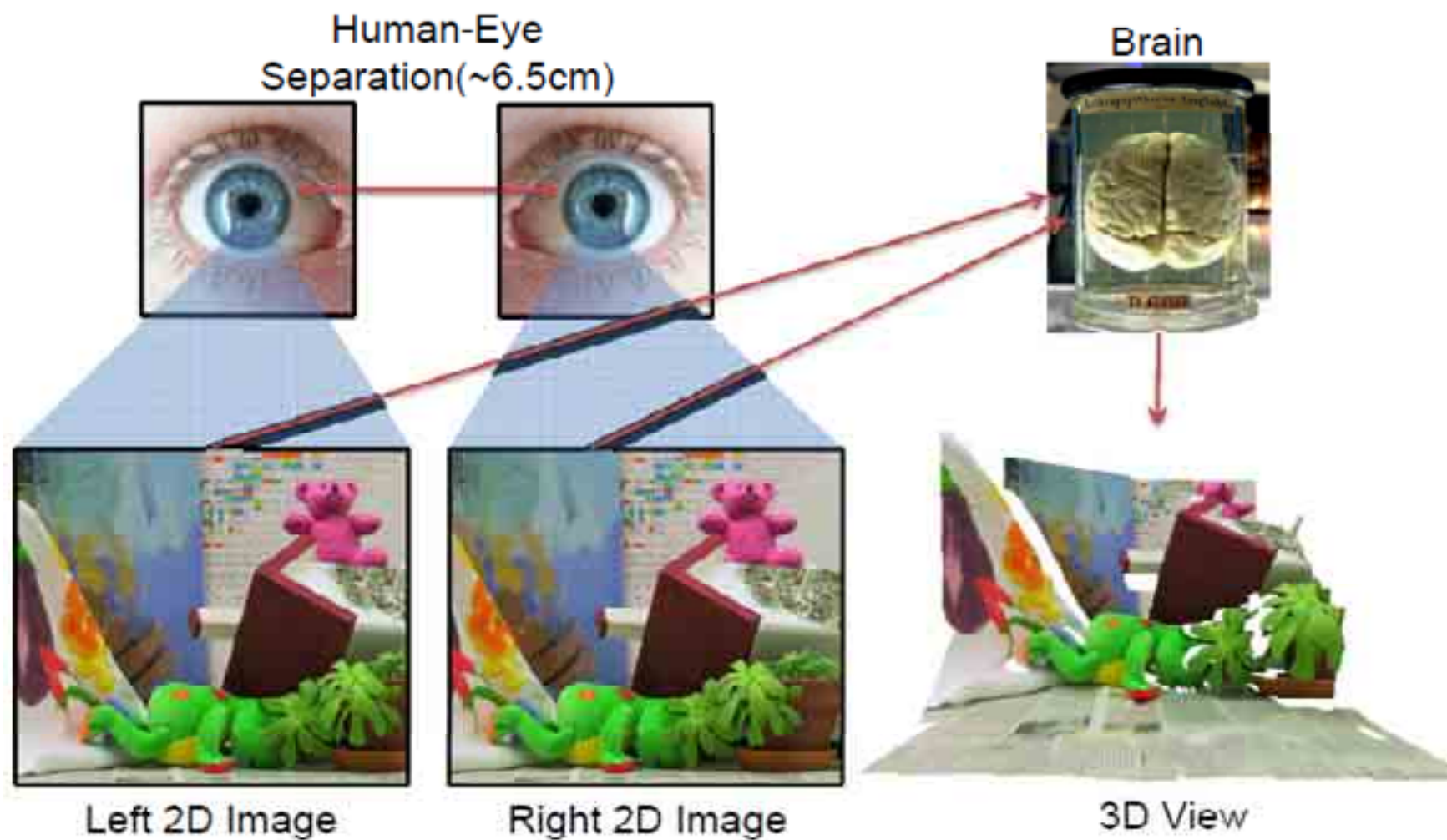


Для левого глаза





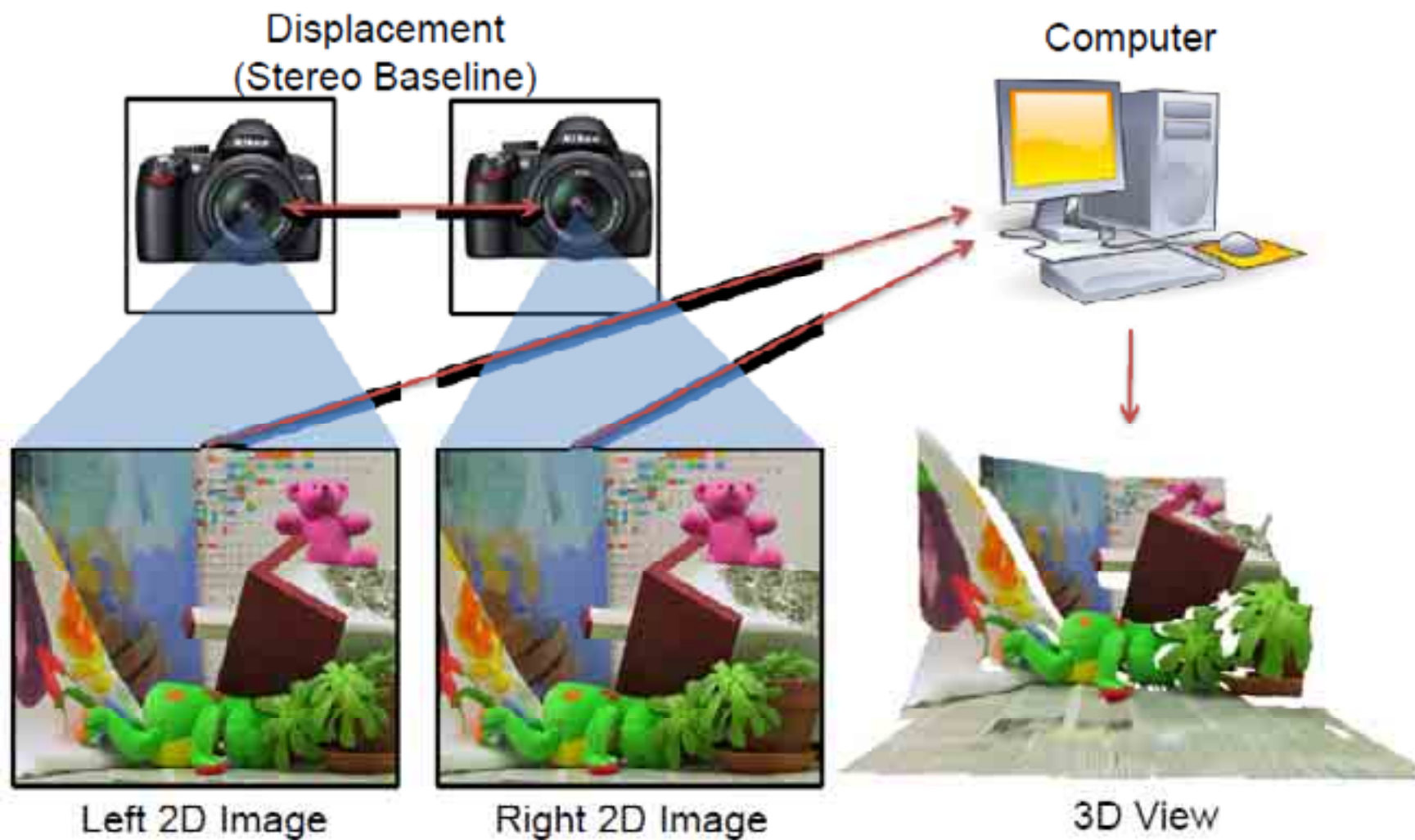
# Зрение человека







# «Зрение» компьютера





# План

---

- Введение
- **Формулировка задачи**
- Трудности и предположения
- Локальные методы
- Глобальные методы
- Использование сегментации
- Резюме



# Смежные задачи

Внутренняя калибровка	Внешняя калибровка	3D-точки	2D-соответствия	Задача
?	?	?	✓	Структура из движения
?	?	✓	✓	Калибровка
✓	?	✓	✓	PnP (внешняя калибровка)
✓	✓	?	✓	Триангуляция
✓	✓	?	?	Стерео



# Смежные задачи

Внутренняя калибровка	Внешняя калибровка	3D-точки	2D-соответствия	Задача
?	?	?	✓	Структура из движения
?	?	✓	✓	Калибровка
✓	?	✓	✓	PnP (внешняя калибровка)
✓	✓	?	✓	Триангуляция
✓	✓	?	?	Стерео



# Смежные задачи

Внутренняя коробочка	Внешняя коробочка	3D-точки	2D-соответствия	Задача
<b>Задача <u>плотного</u> стерео: восстановить в 3D <u>все</u> видимые на изображениях точки</b>				
				Стерео



# Виды стерео

---

- Binocular stereo (два изображения)
- Multi-view stereo (много изображений)
- Photometric stereo





# Виды стерео

---

- Binocular stereo (два изображения)
- Multi-view stereo (много изображений)
- Photometric stereo



# Общая схема

---

Ректификация изображений



Вычисление соответствующих точек



Восстановление 3D путем триангуляции



# Общая схема

---

Ректификация изображений



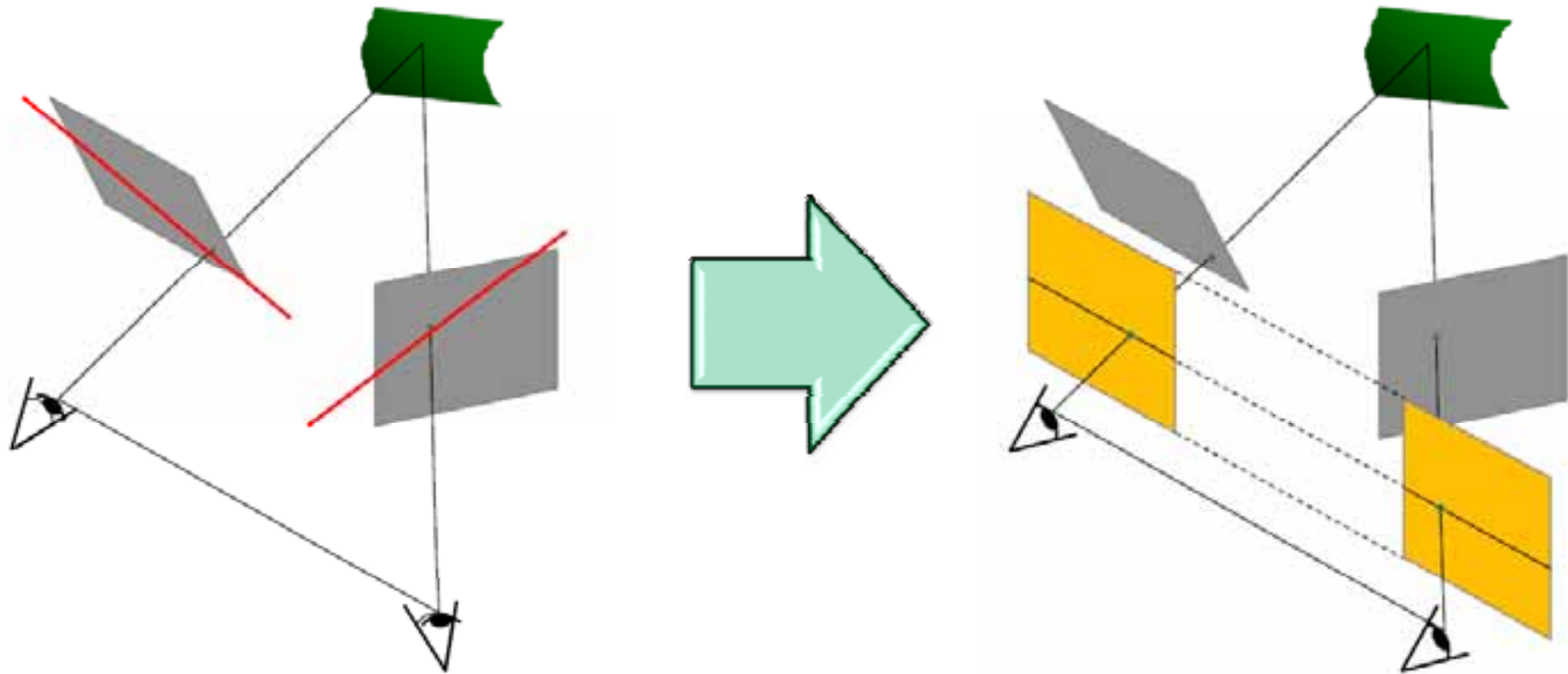
Вычисление соответствующих точек



Восстановление 3D путем триангуляции



# Ректификация

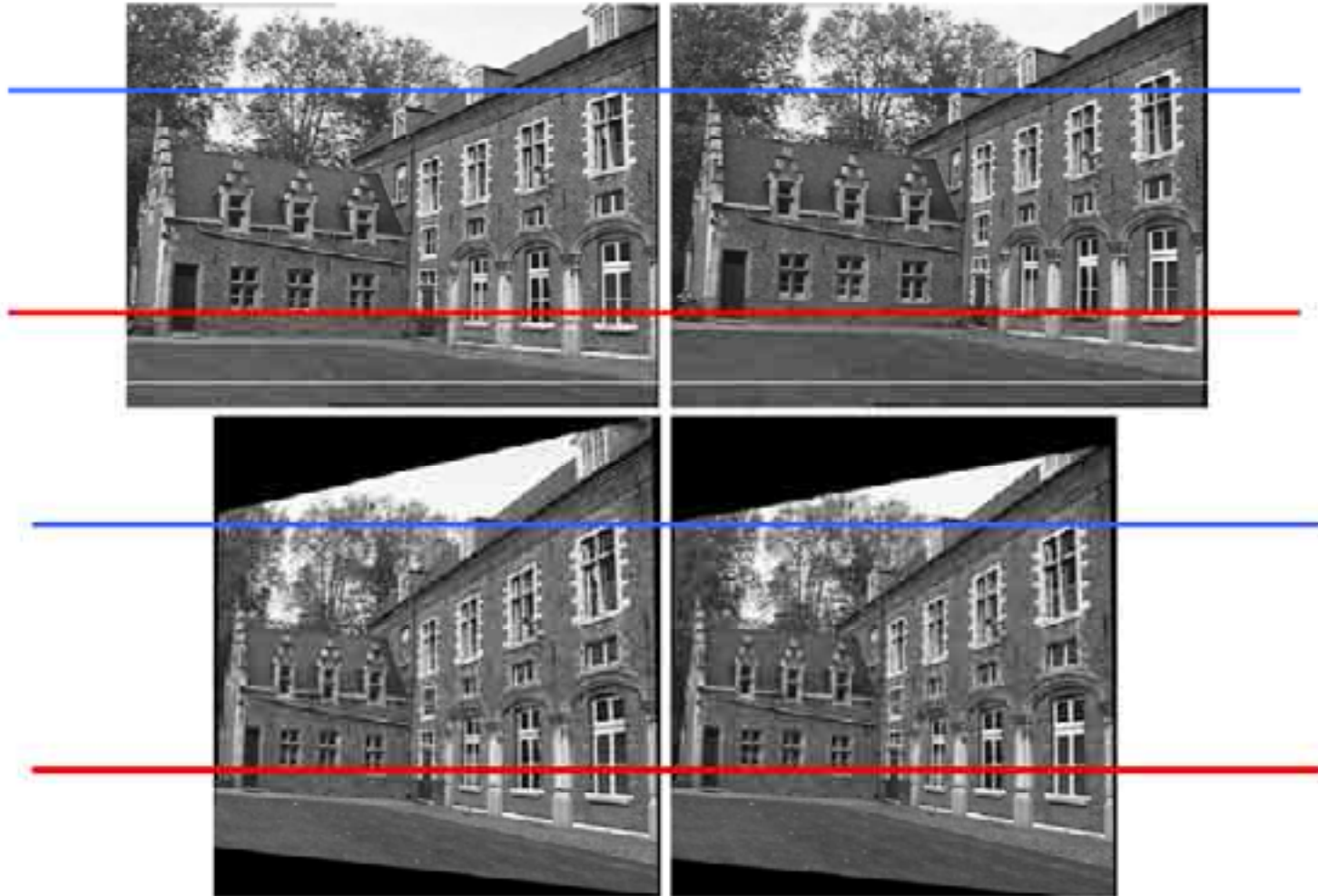


- «Ректификация» – преобразование стереопары в изображения, в которых соответствующие эпиполярные линии лежат на одной и той же горизонтальной строке
- Пример: проецирование на общую плоскость с помощью гомографии



# Ректификация. Пример

---



Недостатки:

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



# Радиальная развёртка

---

- Polar rectification
- Индексируем эпполярные линии углом поворота относительно эпполи
- Копируем соответствующие пары эпполярных линии последовательно в соответствующие горизонталы ректифицированных изображений
- Работает в тех случаях, когда ошибается метод проецирования на общую плоскость





# Пример

---

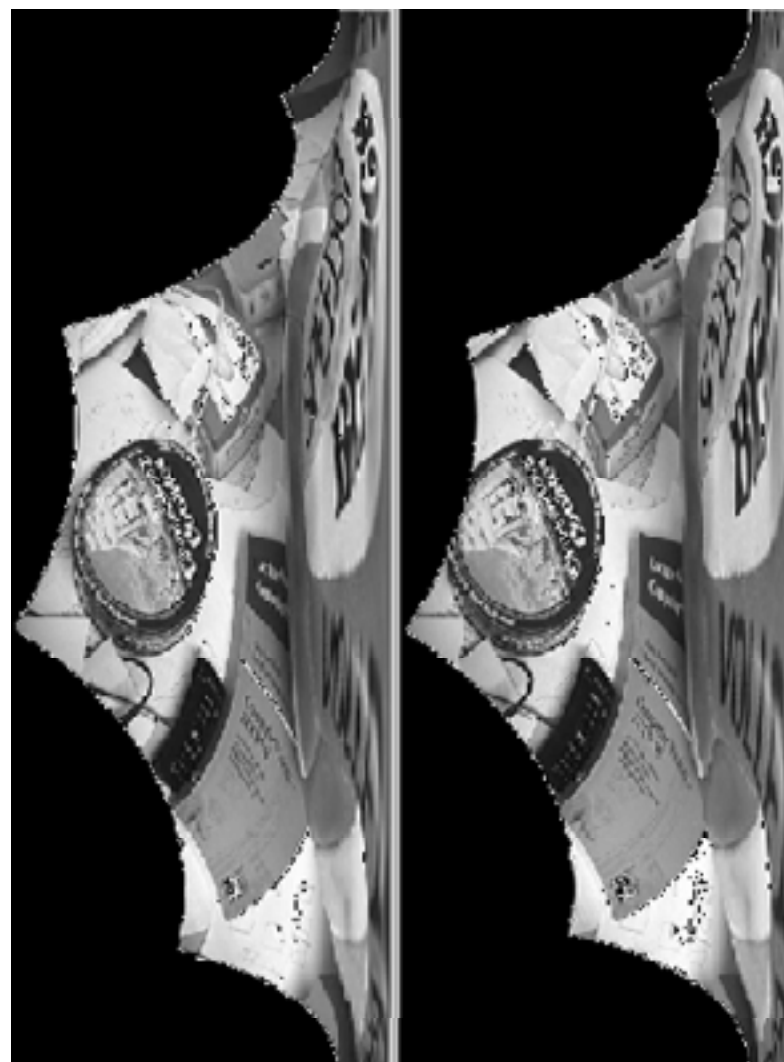


Проецирование

Полярная  
ректификация



# Пример





# Общая схема

---

Ректификация изображений



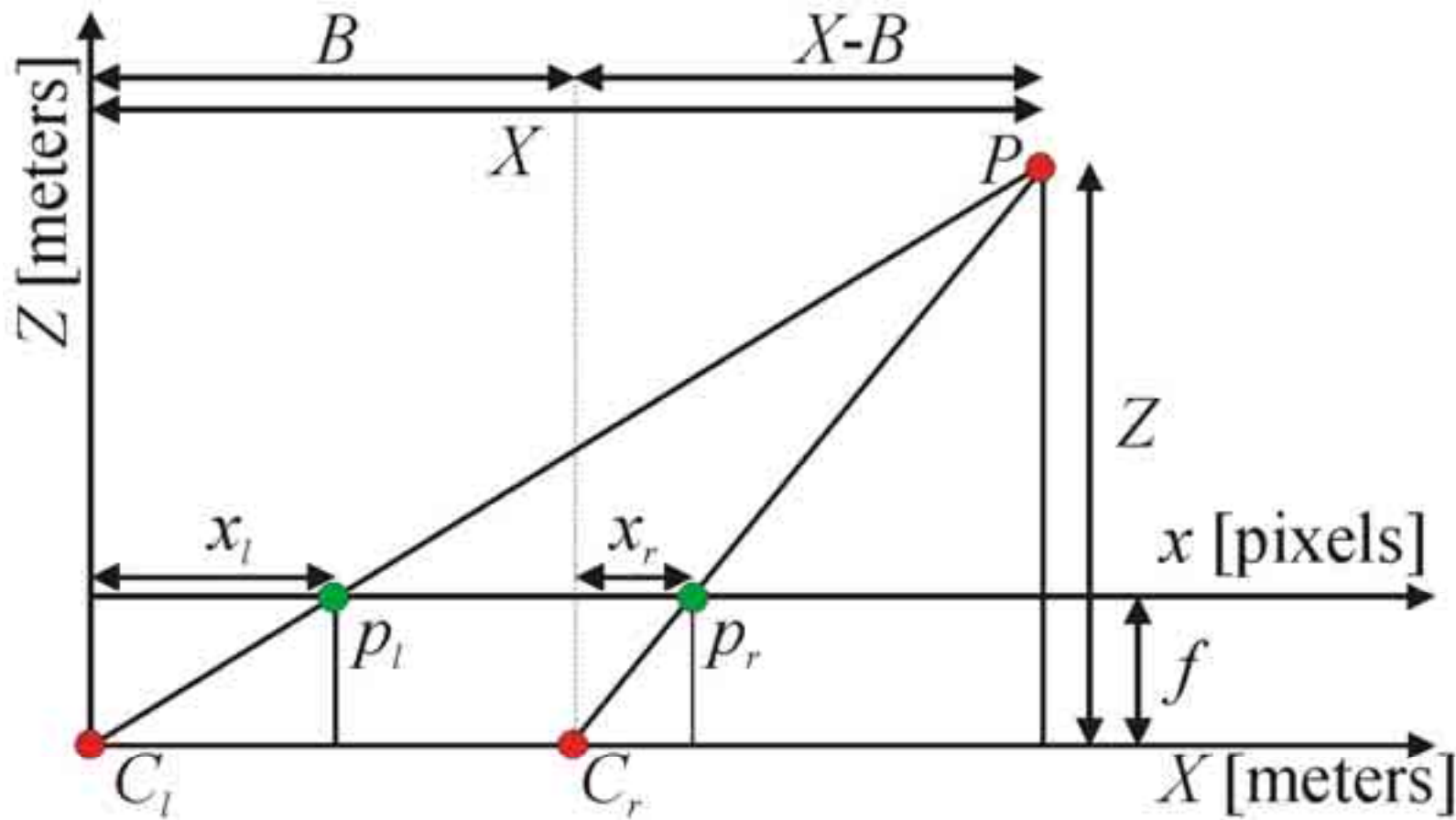
Вычисление соответствующих точек



Восстановление 3D путем триангуляции



# Триангуляция

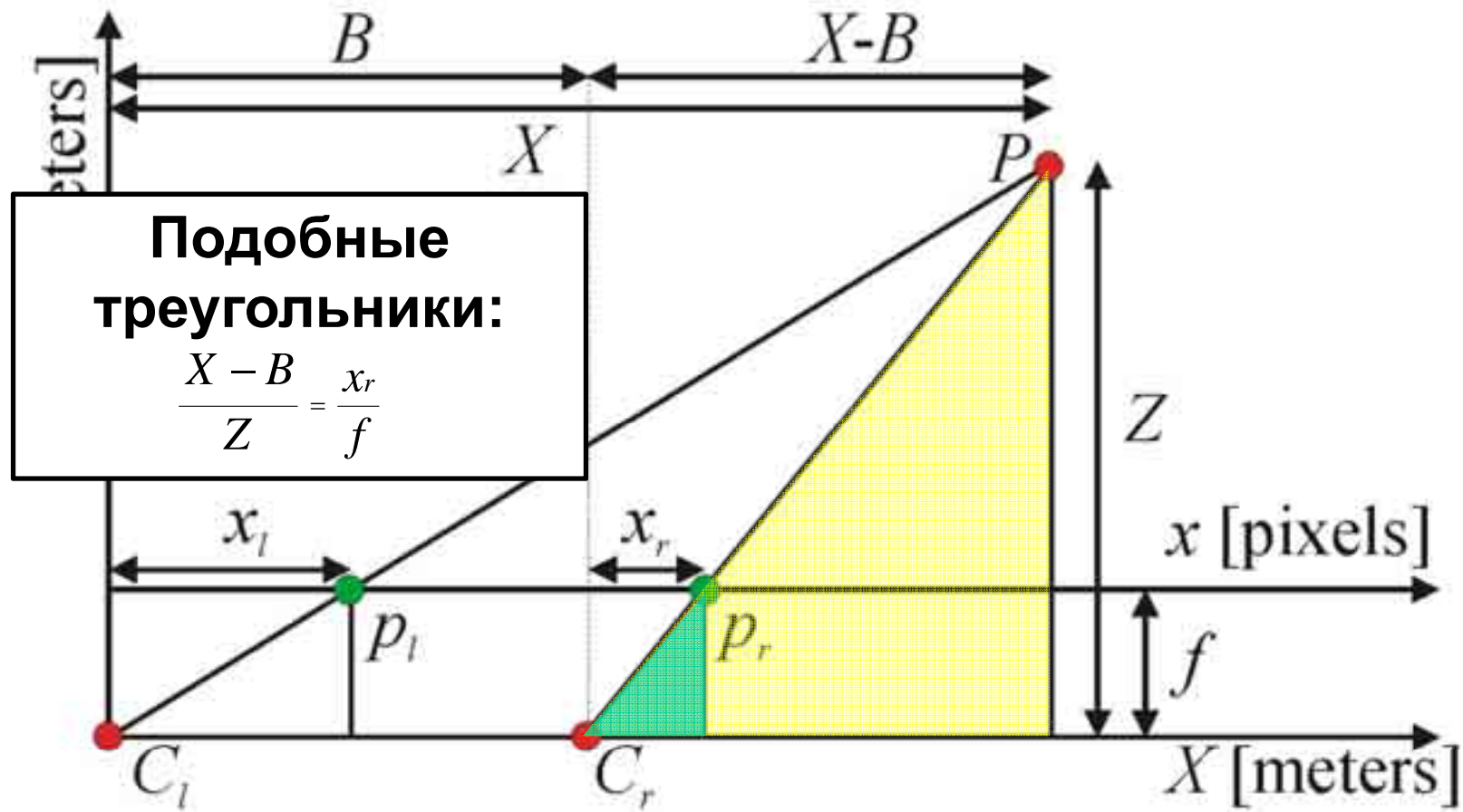


$x_l, x_r$  - смещения относительно принципиальной точки





# 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  $Z$ :

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

Итоговые координаты 3D-точки:

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

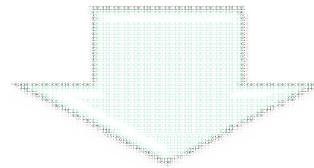
$x_l, x_r$  - смещения относительно принципиальной точки



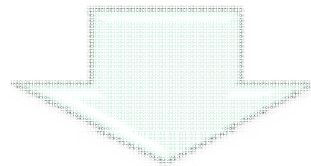
# Общая схема

---

Ректификация изображений



Вычисление карты диспаритета



Восстановление 3D путем триангуляции

Вычисление соответствующих точек сводится к задаче  
вычисления диспаритета



## Общая схема

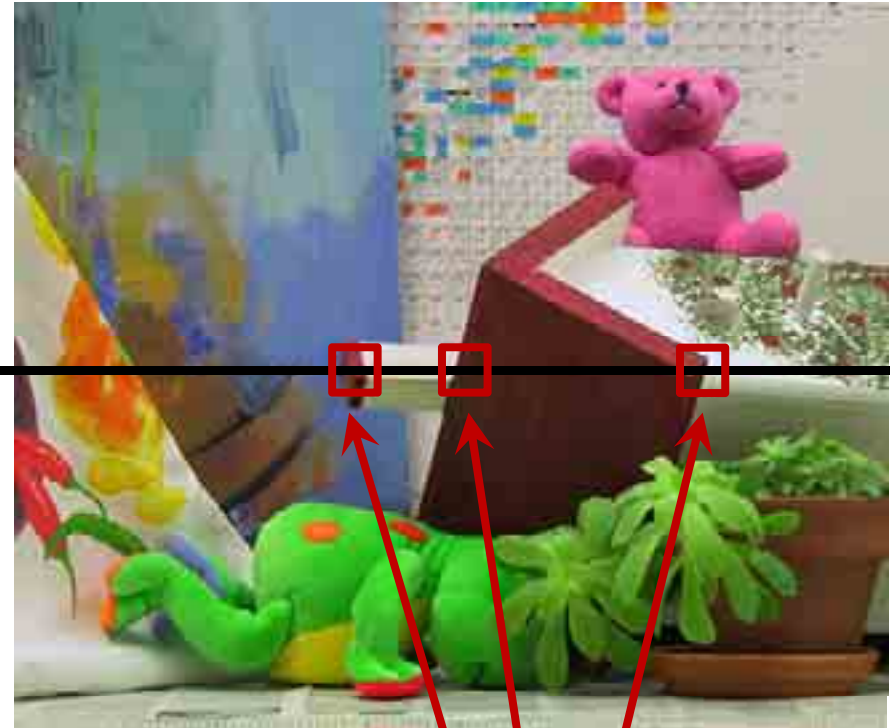
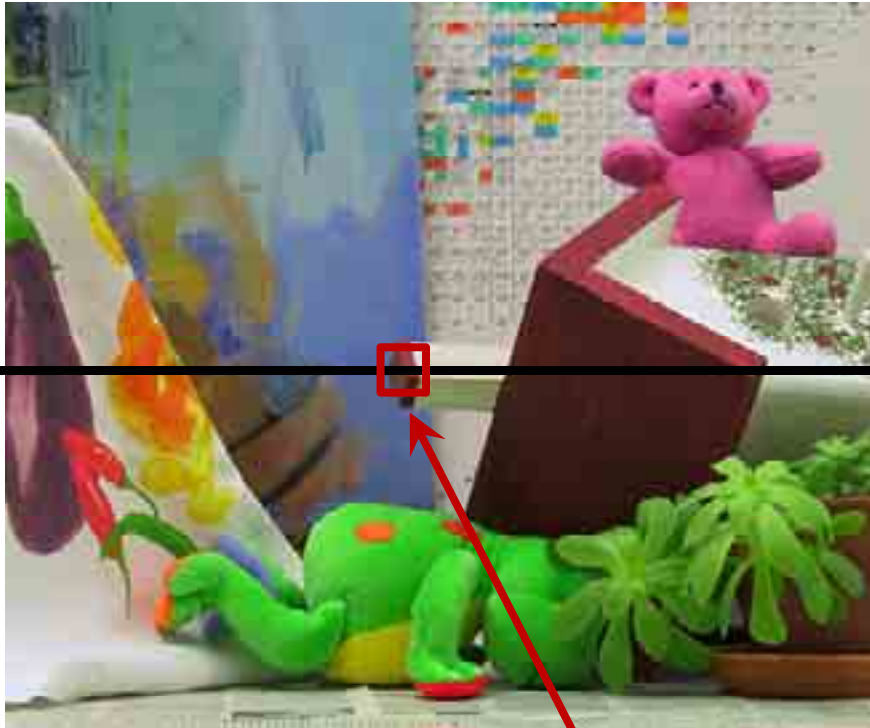
---

Ректификацию и триангуляцию можно считать чисто техническими задачами.

Основной интерес представляет вычисление карты диспаратета



# Вычисление карты диспаритета



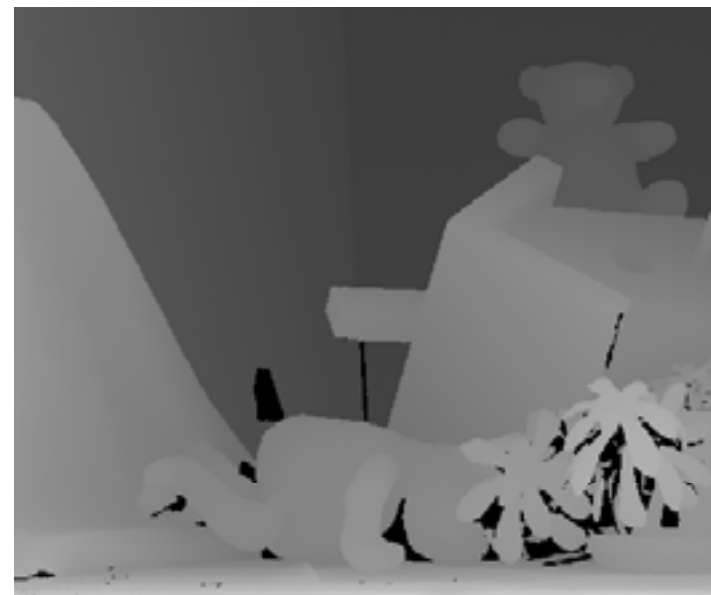
Какой диспаритет  
соответствует этой точке?

?



# Вычисление карты диспаритета

---





# Параллакс

---



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





# Сравнение алгоритмов

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

- Стандартные тестовые наборы
- Рейтинг алгоритмов

Error Threshold = 1 Error Threshold <input type="text"/>		Sort by nonocc ▼			Sort by all ▼			Sort by disc ▼			Average percent of bad pixels ( <a href="#">explanation</a> )			
Algorithm	Avg.	<u>Tsukuba</u> ground truth			<u>Venus</u> ground truth			<u>Teddy</u> ground truth				<u>Cones</u> ground truth		
	Rank ▼	nonocc	all ▼	disc	nonocc	all ▼	disc	nonocc	all ▼	disc		nonocc	all ▼	disc
<a href="#">AdaptiveBP [17]</a>	4.7	<u>1.11</u> 9	1.37 6	5.79 11	<u>0.10</u> 1	0.21 3	1.44 2	<u>4.22</u> 4	7.06 6	11.8 6	<u>2.48</u> 2	7.92 6	7.32 3	<div><div></div></div> 4.23
<a href="#">CoopRegion [41]</a>	4.8	<u>0.87</u> 1	<u>1.16</u> 1	<u>4.61</u> 1	<u>0.11</u> 2	0.21 2	1.54 4	<u>5.16</u> 11	8.31 7	13.0 8	<u>2.79</u> 7	7.18 3	8.01 10	<div><div></div></div> 4.41
<a href="#">DoubleBP [35]</a>	6.6	<u>0.88</u> 2	1.29 2	4.76 3	<u>0.13</u> 3	0.45 13	1.87 9	<u>3.53</u> 3	8.30 6	9.63 2	<u>2.90</u> 9	8.78 17	7.79 7	<div><div></div></div> 4.19
<a href="#">OutlierConf [42]</a>	7.5	<u>0.88</u> 2	1.43 7	4.74 2	<u>0.18</u> 12	0.26 7	2.40 15	<u>5.01</u> 7	9.12 10	12.8 7	<u>2.78</u> 6	8.57 13	6.99 2	<div><div></div></div> 4.60
<a href="#">SubPixDoubleBP [30]</a>	10.1	<u>1.24</u> 10	1.76 16	5.98 12	<u>0.12</u> 4	0.46 15	1.74 7	<u>3.45</u> 2	8.38 8	10.0 3	<u>2.93</u> 11	8.73 10	7.91 9	<div><div></div></div> 4.39
<a href="#">SurfaceStereo [79]</a>	11.1	<u>1.28</u> 21	1.65 12	6.78 24	<u>0.19</u> 14	0.28 8	2.61 21	<u>3.12</u> 1	<u>5.10</u> 1	<u>8.65</u> 1	<u>2.88</u> 6	<u>7.95</u> 8	<u>8.26</u> 14	<div><div></div></div> 4.06
<a href="#">WarpMat [55]</a>	12.1	<u>1.16</u> 10	1.35 4	6.04 13	<u>0.18</u> 13	0.24 5	2.44 16	<u>5.02</u> 8	9.30 11	13.0 10	<u>3.49</u> 16	8.47 12	9.01 24	<div><div></div></div> 4.98
<a href="#">UndisOcSsn [48]</a>	18.2	<u>1.80</u> 28	2.22 28	7.22 22	<u>0.11</u> 3	0.22 4	1.34 5	<u>8.51</u> 18	9.08 13	16.4 14	<u>2.93</u> 10	<u>8.00</u> 9	<u>7.00</u> 5	<div><div></div></div> 5.30



# План

---

- Введение
- Формулировка задачи
- **Трудности и предположения**
- Локальные методы
- Глобальные методы
- Использование сегментации
- Резюме



# Основные трудности

---

## Цветовые несоответствия

- Шум
- Различная освещенность
- Зеркальные отражения
- ...



# 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

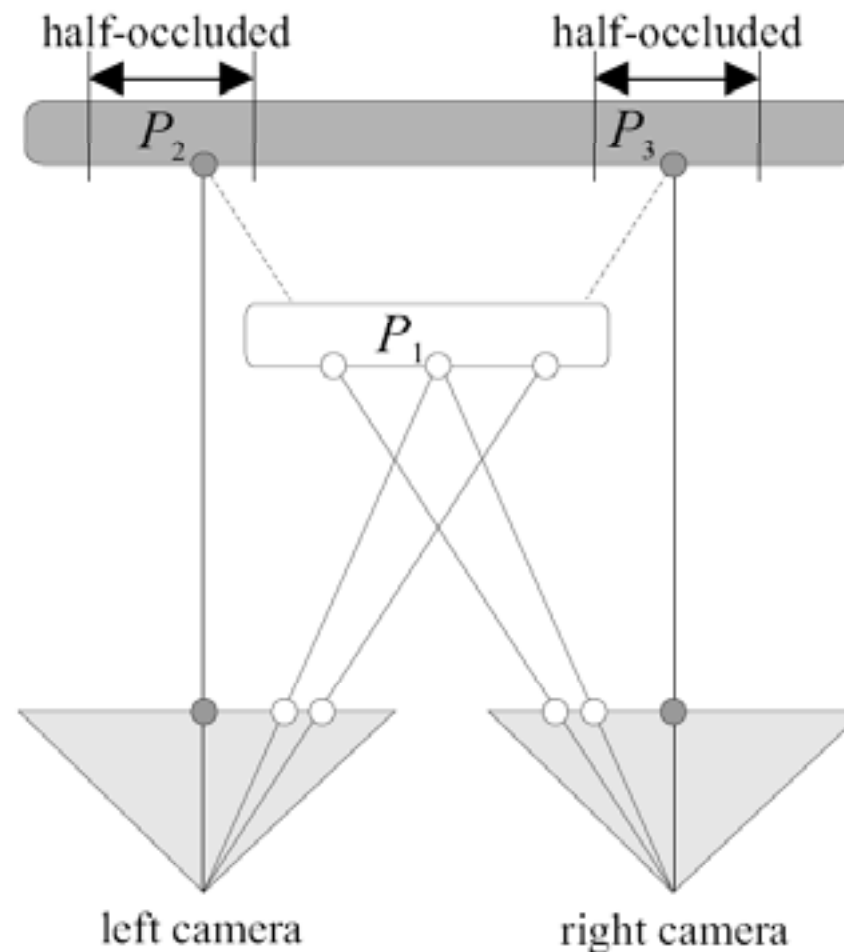


# Основные трудности

## Перекрытия

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

«Невидимые» области называют областями перекрытия





# Основные трудности

---

## Перекрытия



Пример областей перекрытия (выделены красным)





# Основные предположения

---

Эпиполярное ограничение:  
«Соответствующие точки лежат на одной строке»



Самое надежное и обоснованное из всех предположений!





# Основные предположения

---

Соответствие цветов:  
«Цвета соответствующих точек совпадают»



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



# 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!

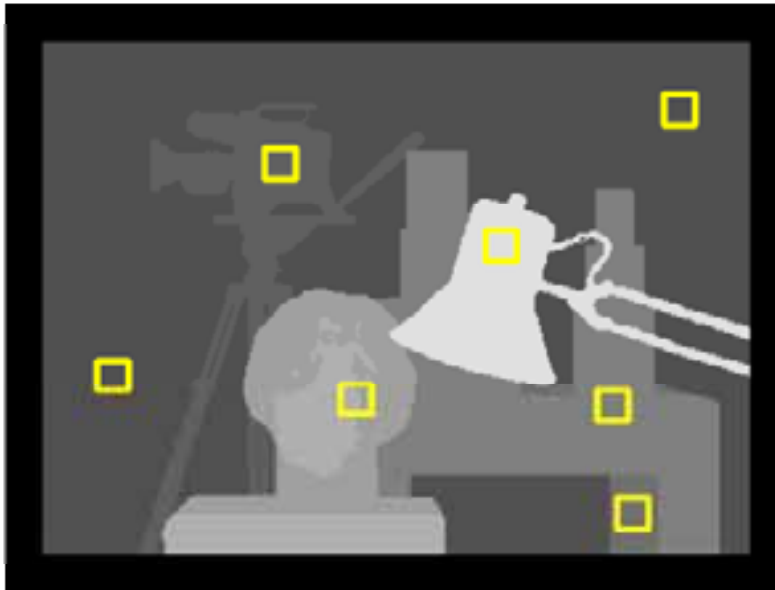


# Основные предположения

---

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»

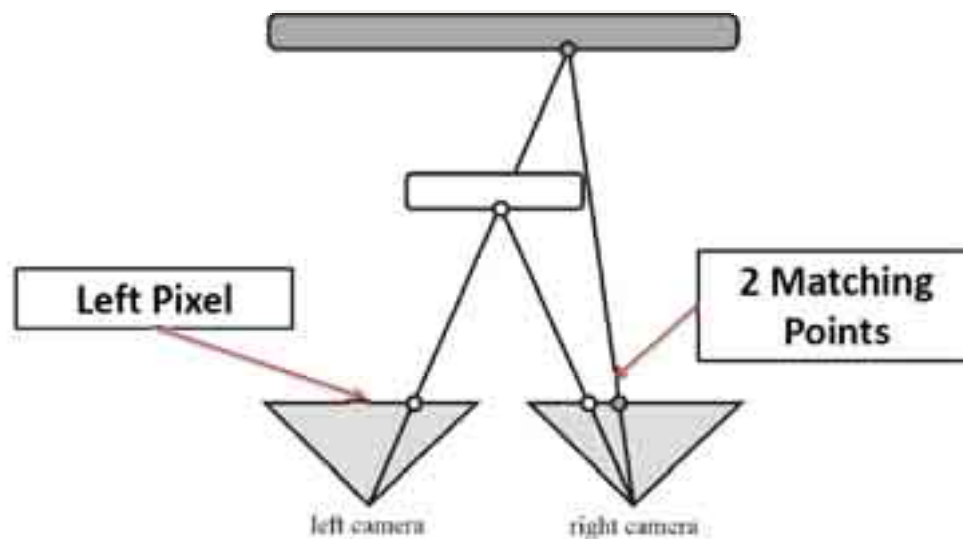
- Нарушается при наличии в сцене прозрачных объектов и горизонтально наклоненных плоскостей



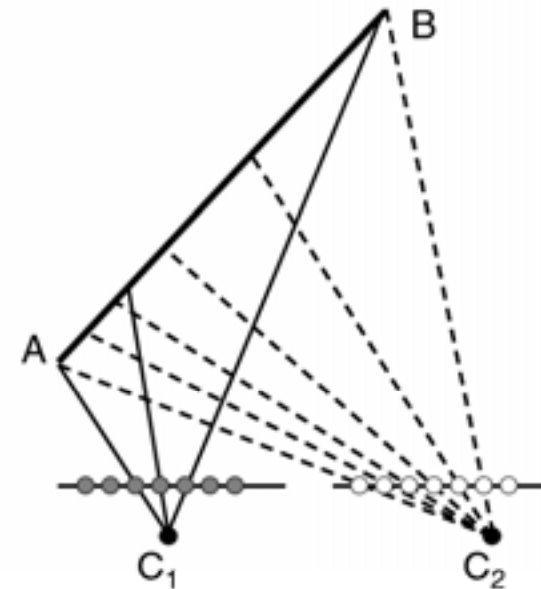
# Основные предположения

Уникальность:

«Каждая точка одного изображения имеет не более одного соответствия на другом»



Прозрачный объект



Горизонтально наклоненная  
плоскость



# Основные предположения

---

Сохранение порядка точек:

«Порядок относительного расположения точек сохраняется и для их соответствий»

- Нарушается при наличии в сцене тонких объектов переднего плана
- Данное предположение можно считать «устаревшим»



# Основные предположения

---

Сохранение порядка точек:

«Порядок относительного расположения точек сохраняется и для их соответствий»



Пример нарушения предположения

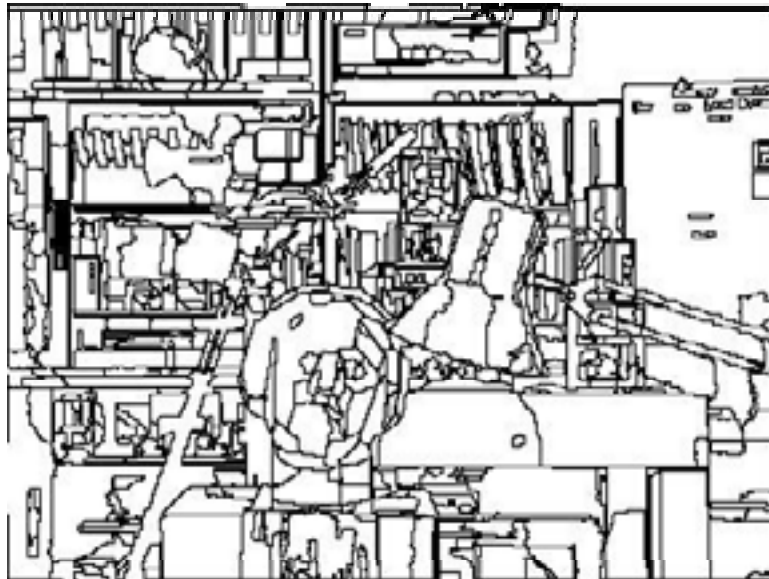




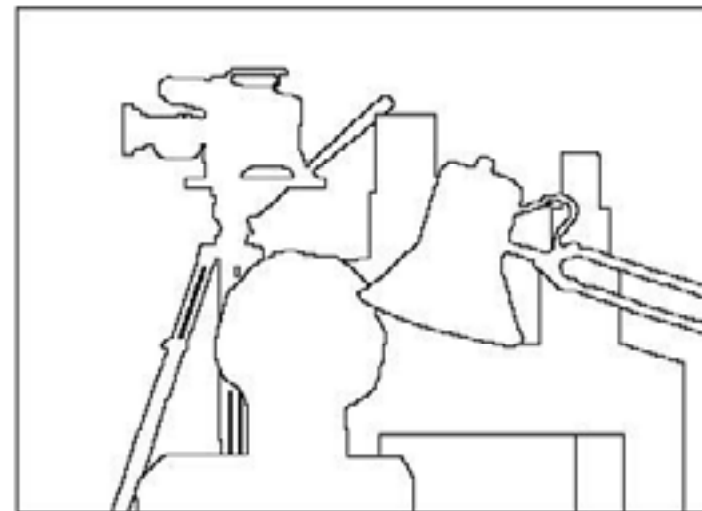
# Основные предположения

---

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



Границы цветных сегментов



Границы диспаритета



# Основные предположения

---

Предположение о сегментации:

«Области разрыва диспаритета совпадают с краями на изображении»

- Лежит в основе алгоритмов, использующих пересегментацию, - лидеров рейтинга Middlebury
- Может нарушаться (ничто не гарантирует, что граница объекта не попадет внутрь цветового сегмента)



# План

---

- Введение
- Формулировка задачи
- Трудности и предположения
- **Локальные методы**
- Глобальные методы
- Использование сегментации
- Резюме



# Локальные методы

---

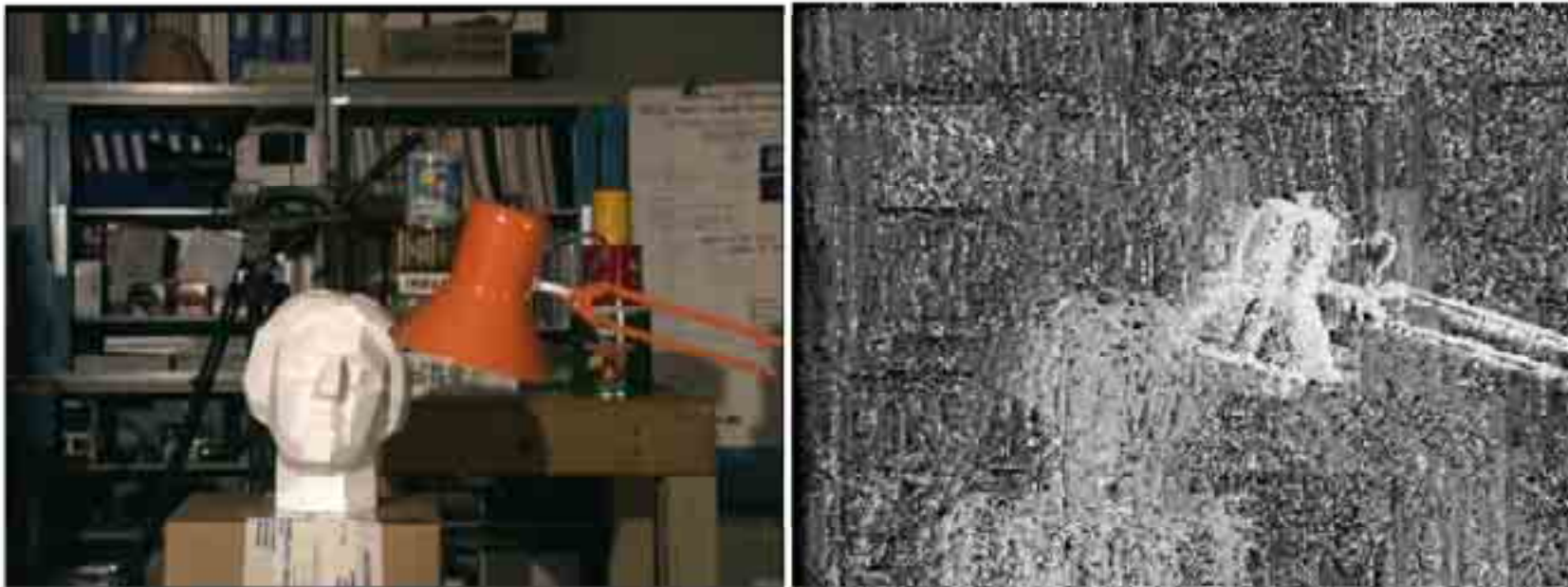
- Локальные, т.к. диспаритет в каждой точке зависит только от ее локальной окрестности
- Используют стратегию WTA (Winner-Take-All), т.е. в каждой точке «побеждает» соответствие с наименьшей «СТОИМОСТЬЮ»



# Наивный алгоритм

---

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



Пример работы наивного алгоритма



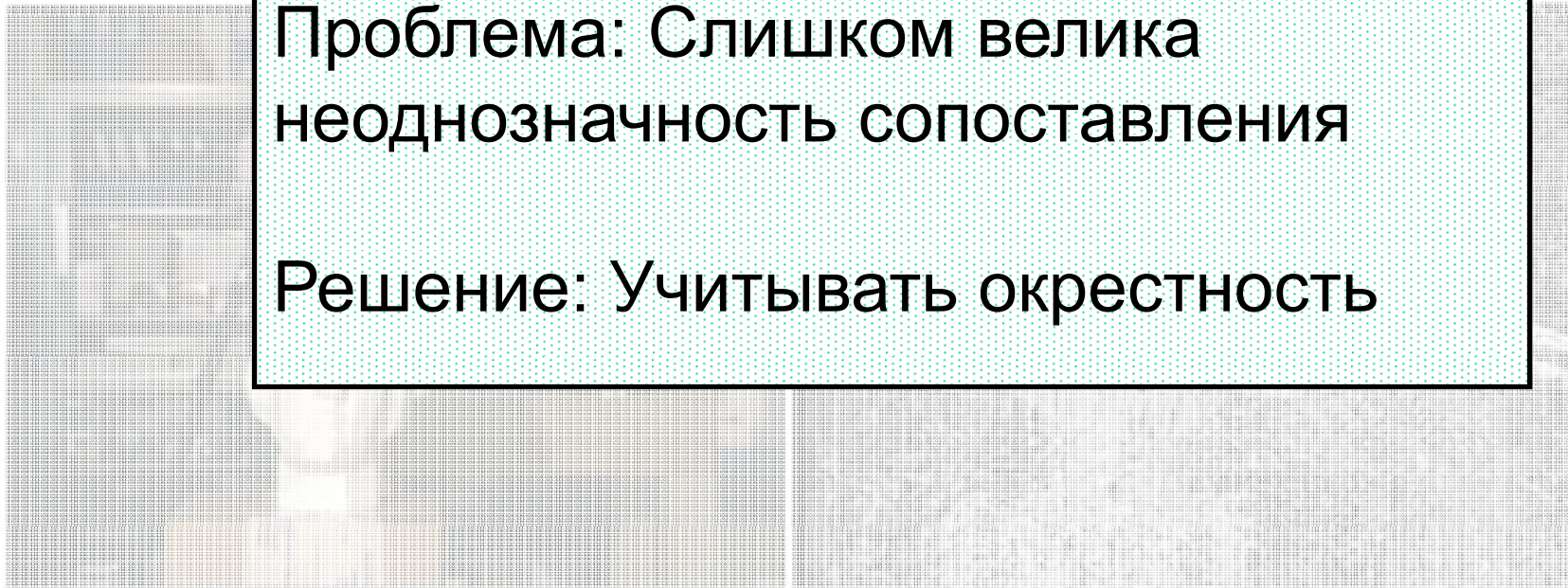
# Наивный алгоритм

---

- Стоимость соответствия — разность интенсивностей пикселей
- Результат

Проблема: Слишком велика  
неоднозначность сопоставления

Решение: Учитывать окрестность



Пример работы наивного алгоритма

Пример: M. Bleyer



# 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:

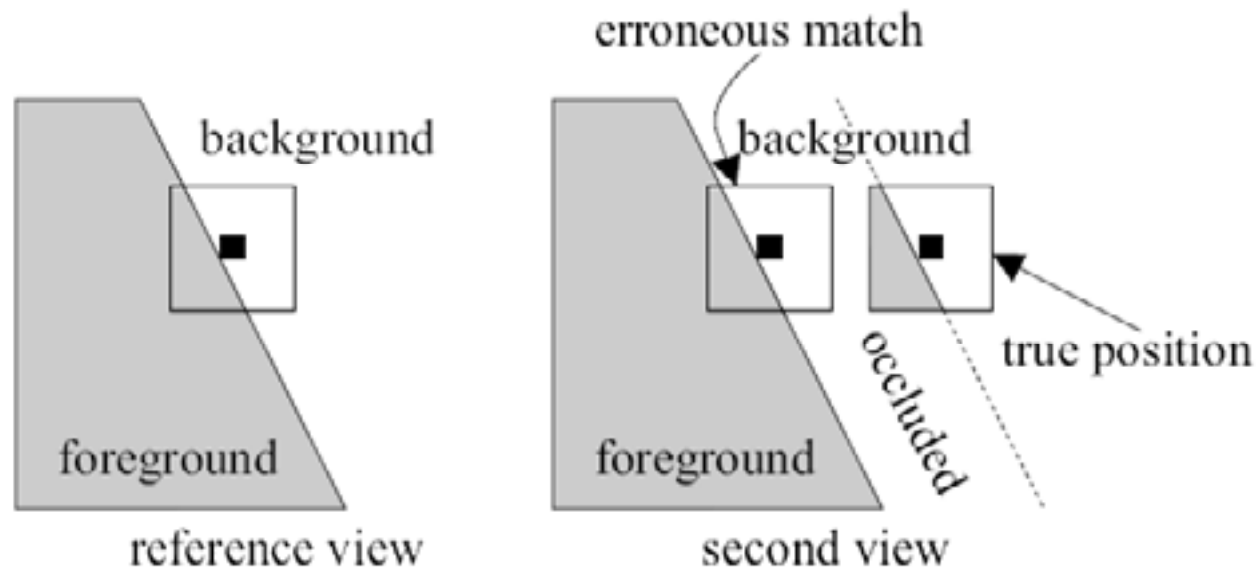
- $d_p$  – искомый диспаритет в пикселе  $p$ ,
- $c(p, q)$  – функция стоимости,
- $W_p$  – окно вокруг пикселя  $p$ ,
- $d_{\max}$  – максимально возможный диспаритет

- 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

---

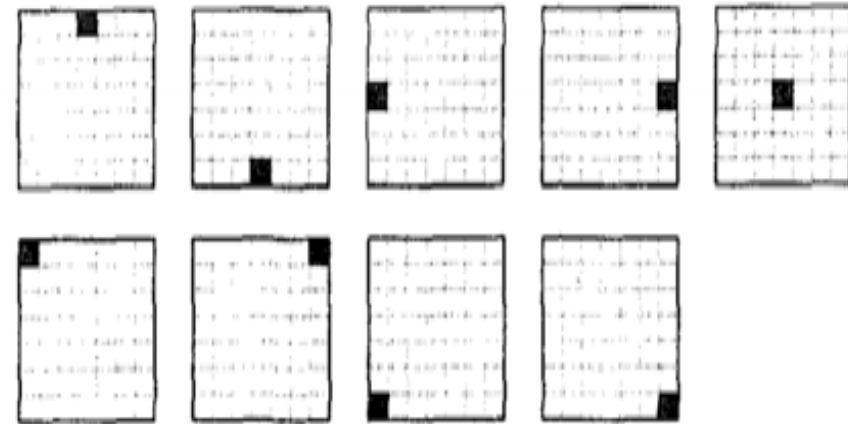
- Сопоставление окон неявным образом реализует предположение о гладкости
  - Все пиксели внутри окна полагаются имеющими одинаковый диспаритет, что, вообще говоря, неверно
- Выход — исключить границы из рассмотрения с помощью адаптивных окон
  - Окна с адаптивной структурой
  - Окна с адаптивными весами



# 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.**

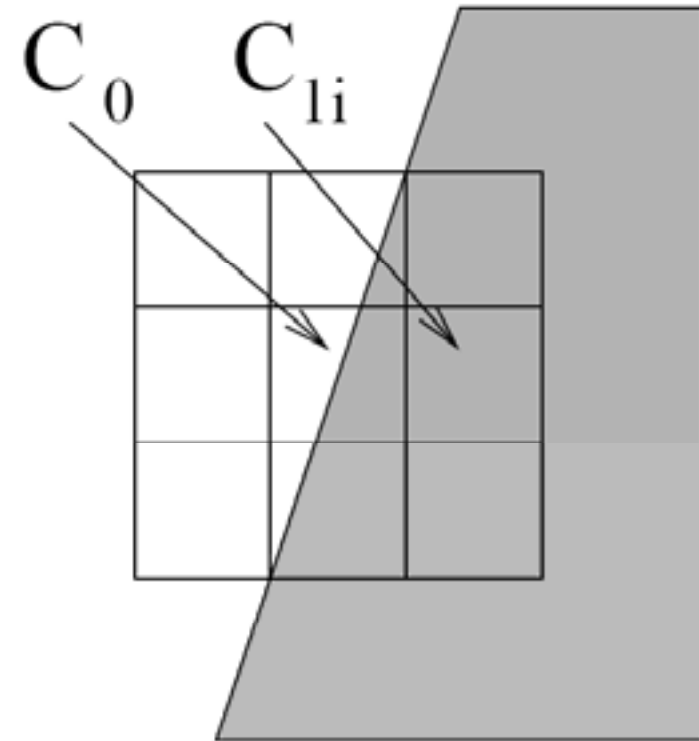


# Окна с адаптивной структурой

---

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  $w(p, q)$ ?

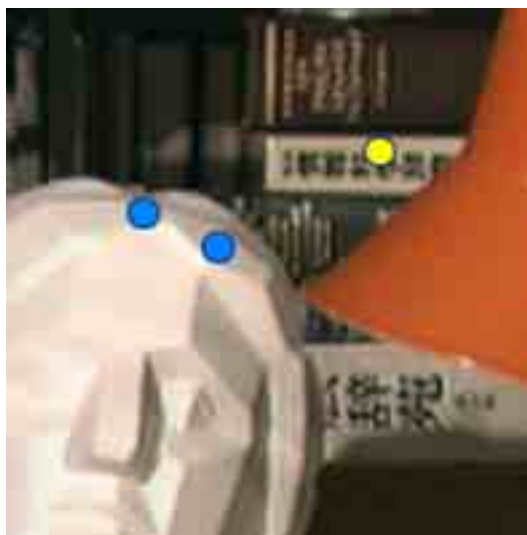


# Вычисление веса

- Эвристические подсказки
  - Цвет
  - Расстояние
  - Границы



Цвет



Расстояние



Граница



# Адаптивные окна. Резюме

---

- Достоинство: гибкость
- Недостаток: скорость
  - Метод «скользящего окна» неприменим
  - Тем не менее, достаточно эффективны
- Развитие локальных методов на сегодняшний день идет именно в направлении адаптивных окон



# 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**



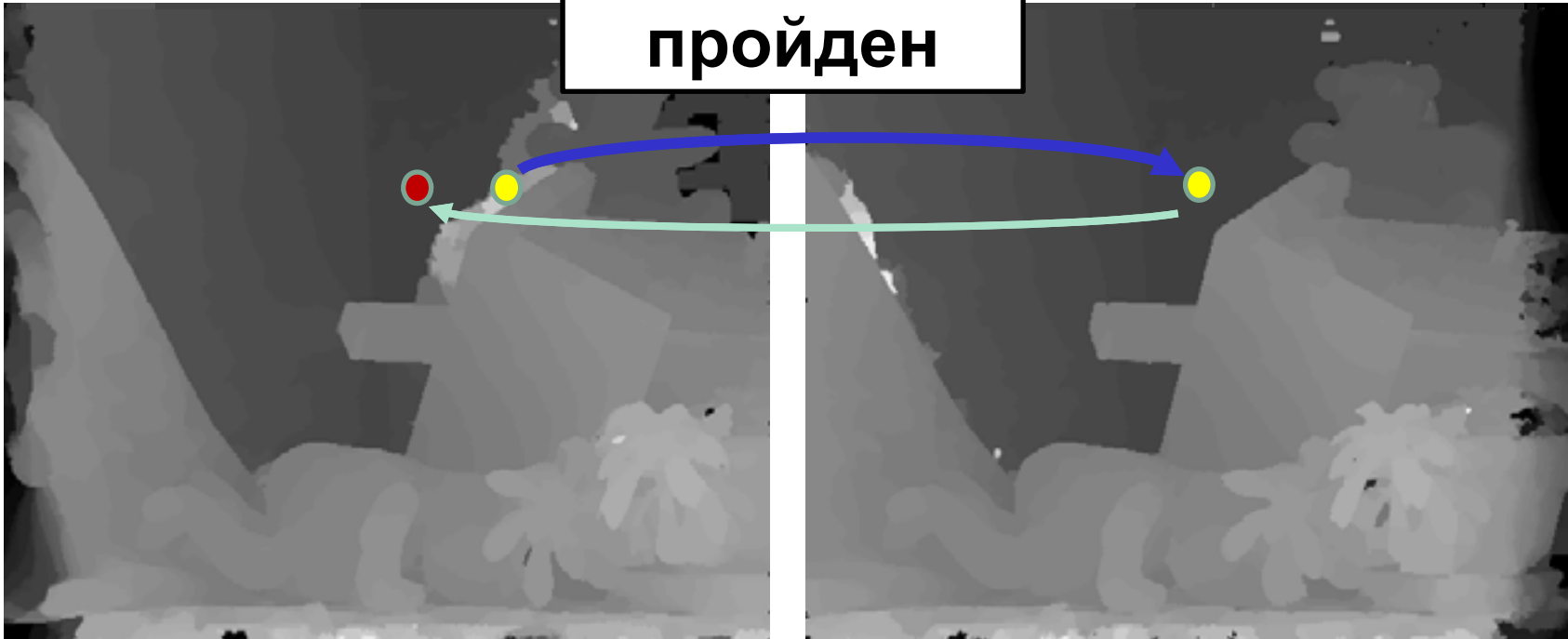
Карта диспаритета (левое  
изображение)

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:



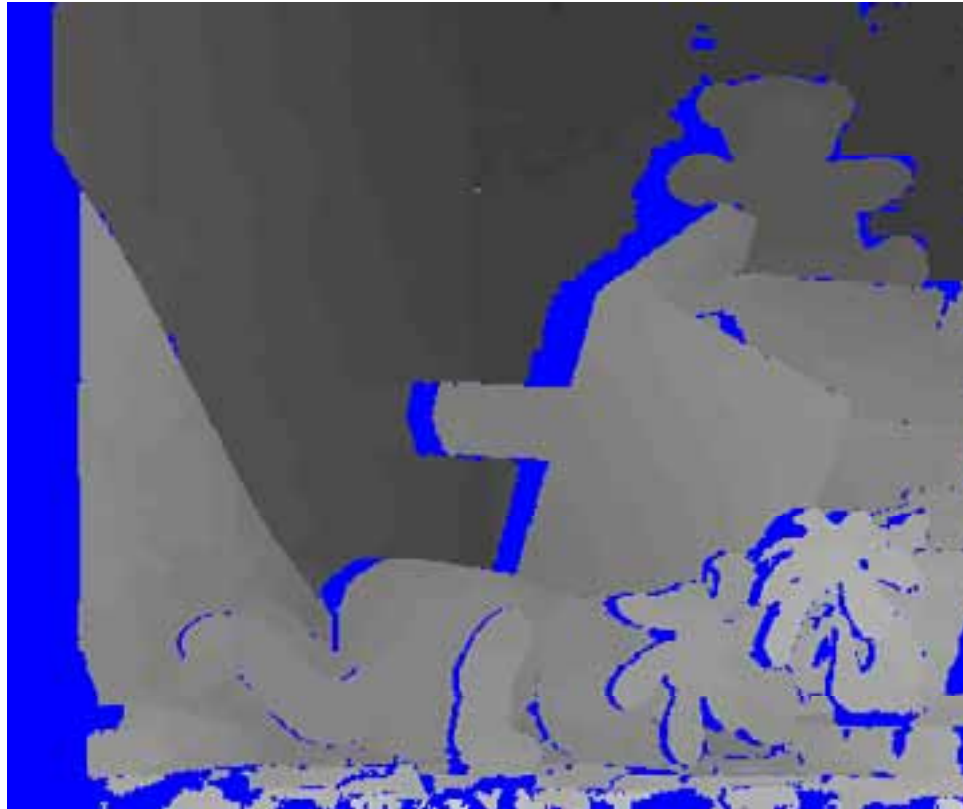
Диспаритет одного из двух пикселей слева будет обнулен в результате кросс-проверки

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



# Борьба с перекрытиями

---

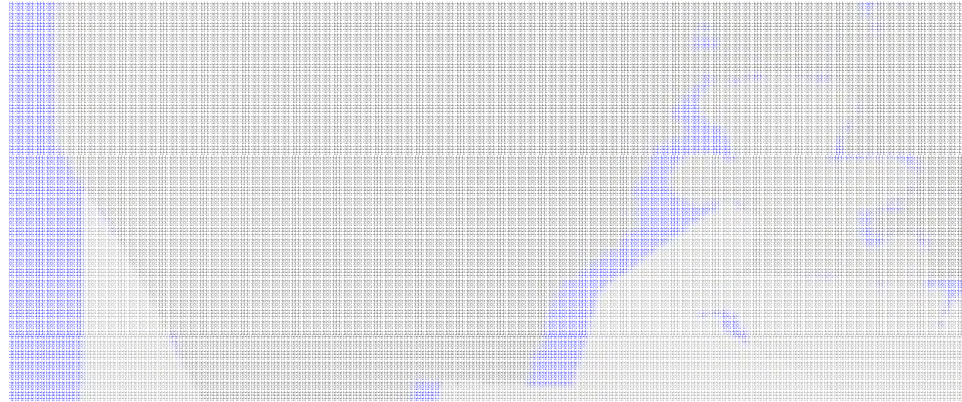


Пиксели, не прошедшие кросс-проверку  
(выделены синим)

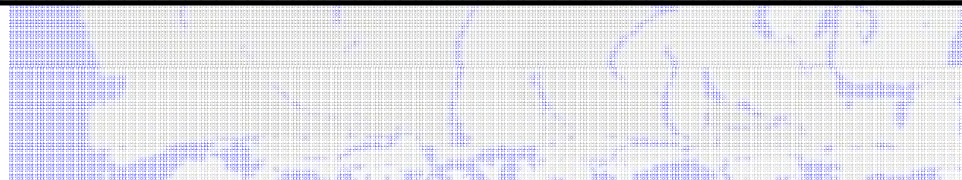


# Борьба с перекрытиями

---



**«Пустые» области необходимо  
заполнить**



Пиксели, не прошедшие кросс-проверку  
(выделены синим)

Пример: M. Bleyer

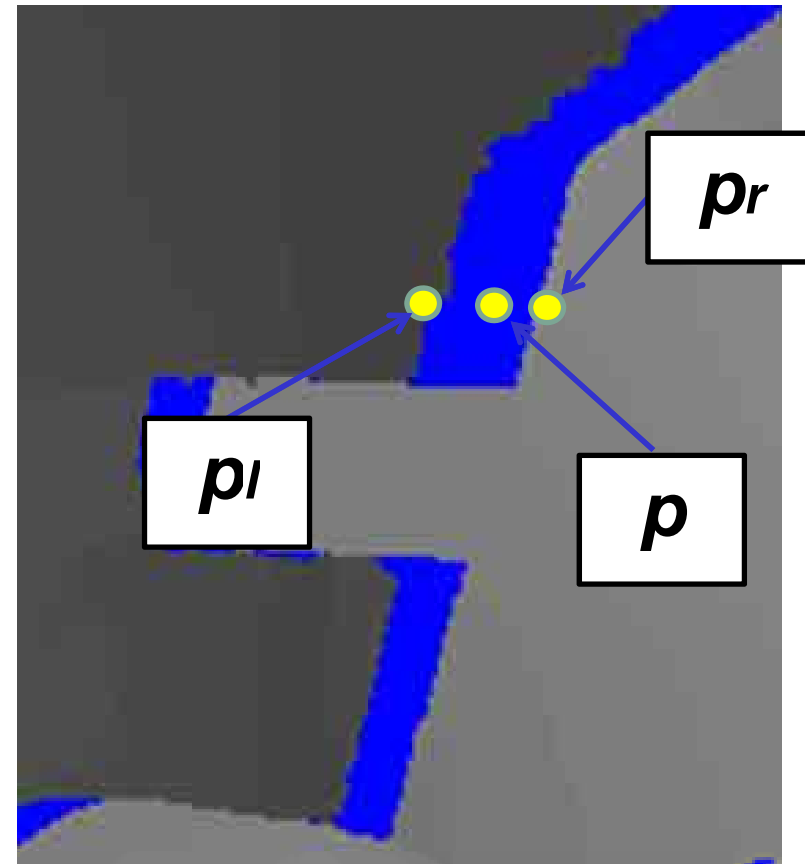


# Заполнение областей перекрытия

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
- Формулировка задачи
- Трудности и предположения
- Локальные методы
- **Глобальные методы**
- Использование сегментации
- Резюме





# 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



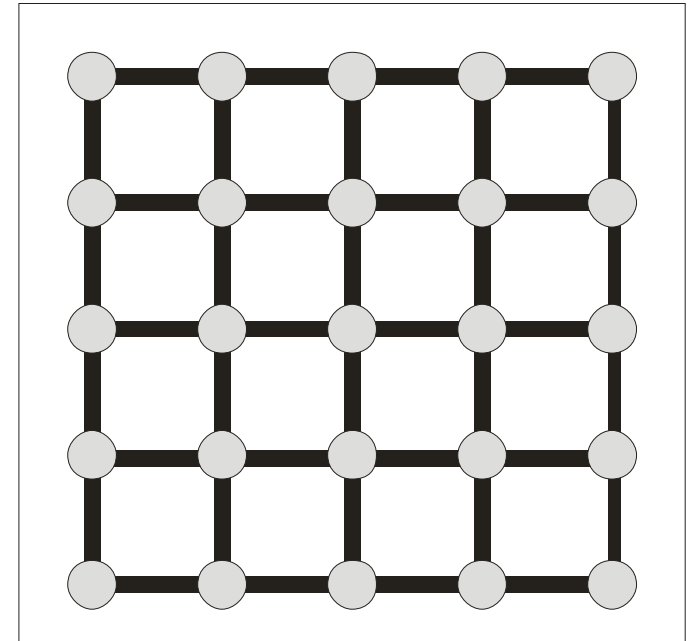
# Глобальные методы

---

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

Метки – значения диспаритета.

Граф – решетка, узлы – пиксели.



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



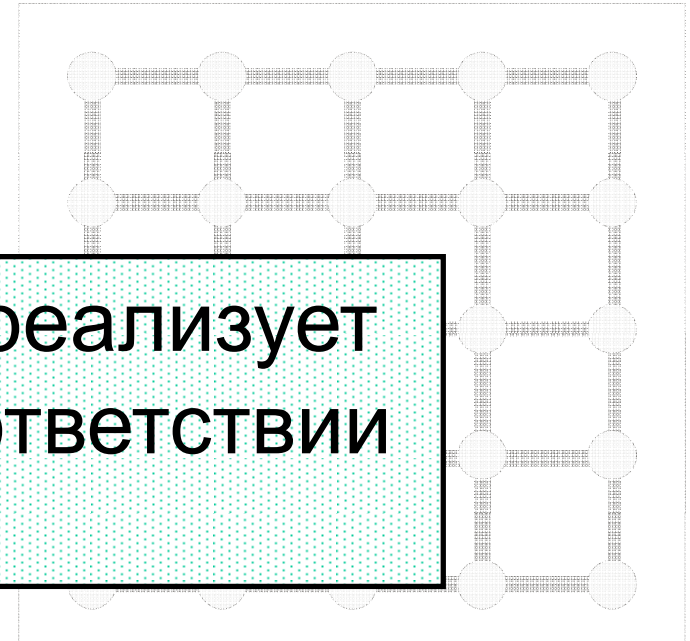
# Глобальные методы

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

Метки —

Унарный потенциал реализует  
предположение о соответствии  
ЦВЕТОВ

Граф — решетка, узлы — пиксели.



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



# Глобальные методы

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

Метки —

Парный потенциал явным  
образом реализует  
предположение о гладкости

Граф — решетка, узлы — пиксели.



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



# Унарный потенциал

---

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

где:

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

Например:

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

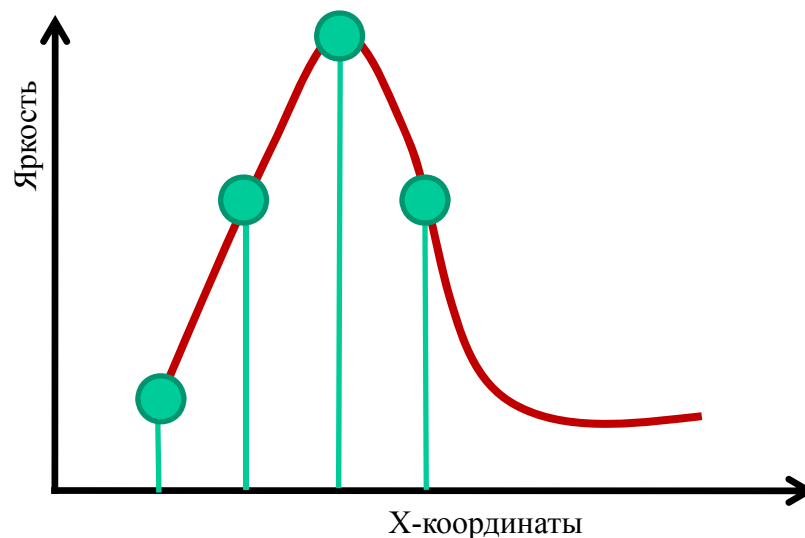
$$m(p, q) = \min(|I_p - I_q|, k)$$

Зачастую применяется метрика Birchfield-Tomasi, устойчивая к погрешности сэмплирования

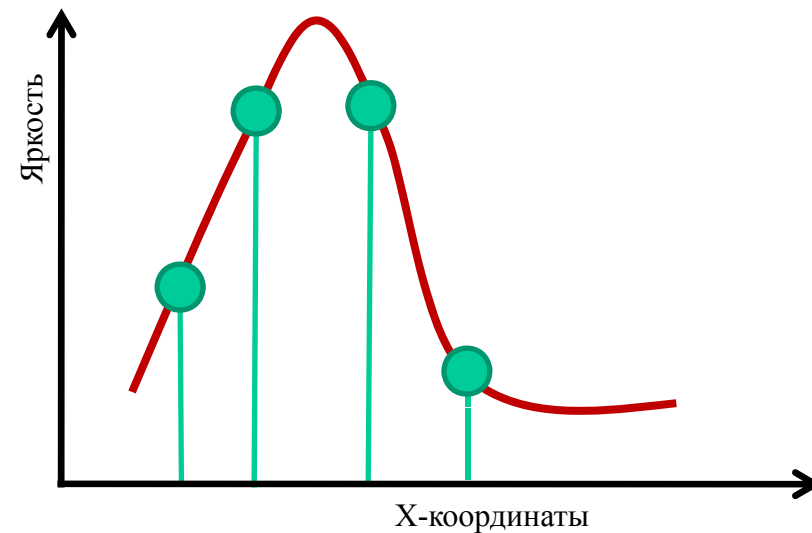


# Метрика Birchfield-Tomasi

Проблема: Непрерывный сигнал может быть сэмплирован по-разному. Цвета соответствующих пикселей могут не совпасть.



(Левое изображение)



(Правое изображение)

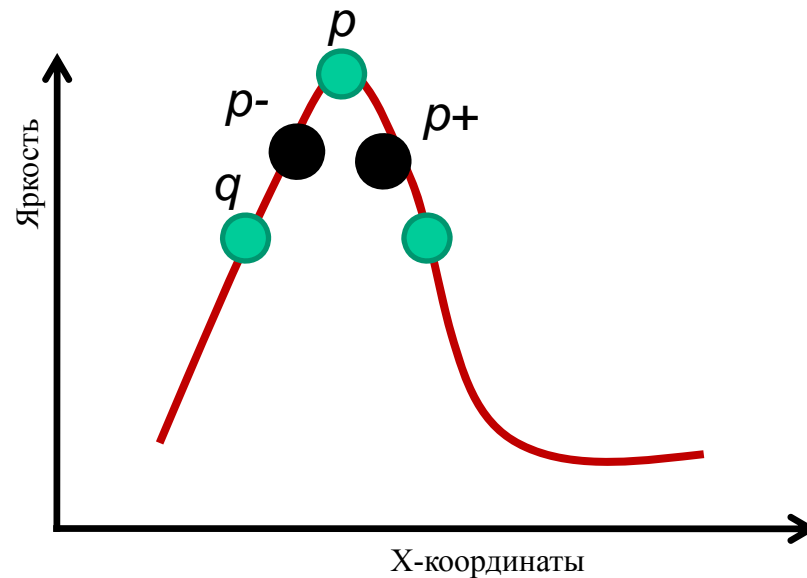
S. Birchfield, C. Tomasi. A Pixel Dissimilarity Measure That Is Insensitive to Image Sampling. *PAMI*, 1998.



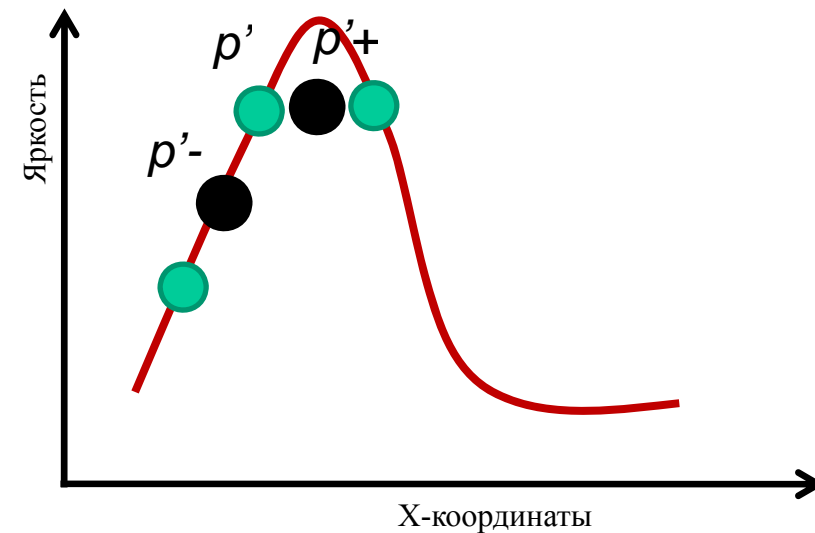
# Метрика Birchfield-Tomasi

Решение: Учитывать средние яркости между рассматриваемыми пикселями и их соседями:

$$m(p, p') = \min(|p - p'|, |p - -p'|, |p_+ - p'|, |p - p'_-|, |p - p'_+|)$$



(Левое изображение)



(Правое изображение)

S. Birchfield, C. Tomasi. A Pixel Dissimilarity Measure That Is Insensitive to Image Sampling. *PAMI*, 1998.



# 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:

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





# 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:

**Линейная модель**

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

или

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



# 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

Модель Поттса

косты

Например:

$$s(d_p, d_q) = |d_p - d_q| \cdot P \quad \text{или}$$

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



# Проблема сохранения границ

---

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





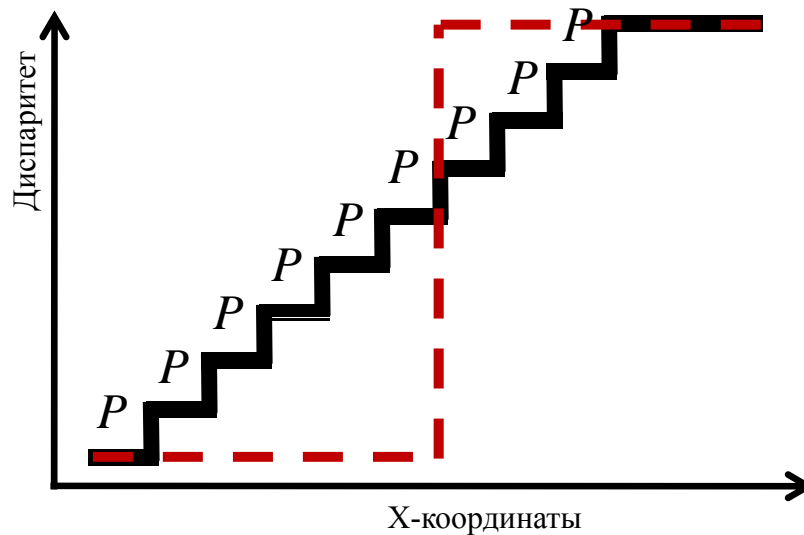
# Проблема сохранения границ

Случай линейной модели:

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

Вклад в энергию:

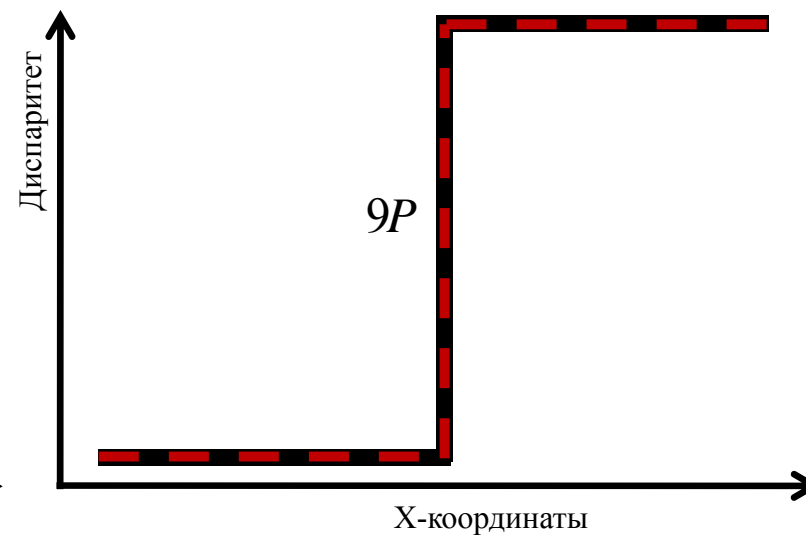
**$9P$**



(неверное решение)

Вклад в энергию:

**$9P$**



(верное решение)



# Проблема сохранения границ

Случай линейной модели:

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

Вклад в энергию:

$9P$

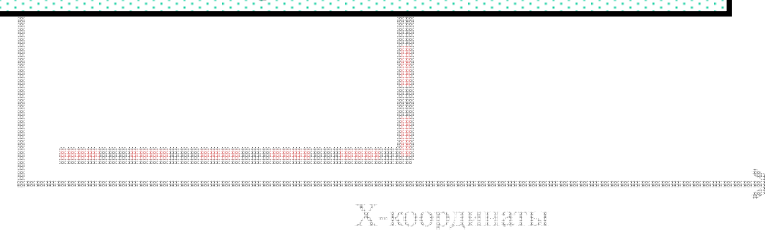
Вклад в энергию:

$9P$

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



(неверное решение)



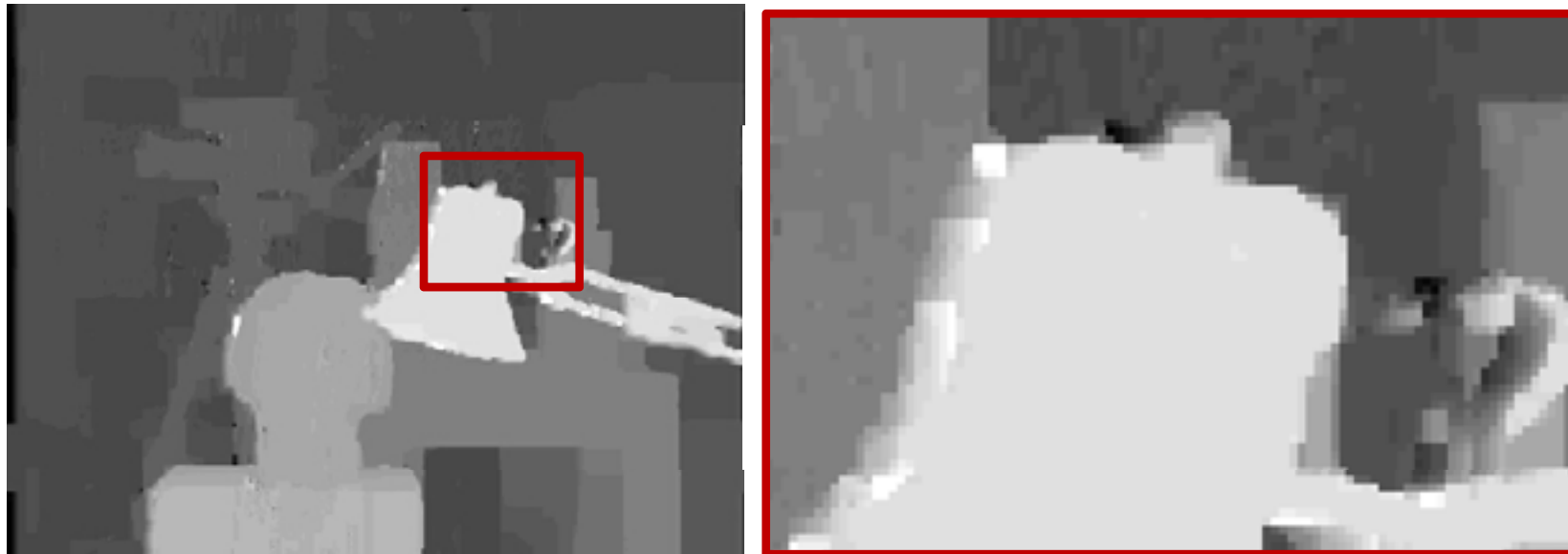
(верное решение)

Рисунки: M. Bleyer



# Проблема сохранения границ

Случай линейной модели:  $s(d_p, d_q) = |d_p - d_q| \cdot P$



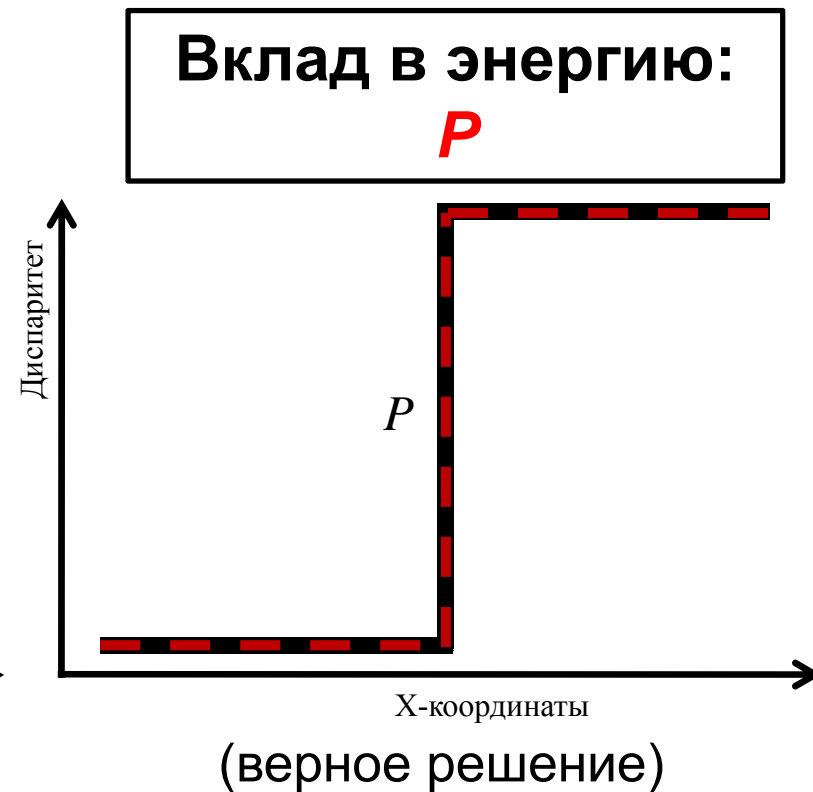
Пример результатов для линейной модели. Справа показан увеличенный фрагмент



# Проблема сохранения границ

Случай модели Поттса:

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



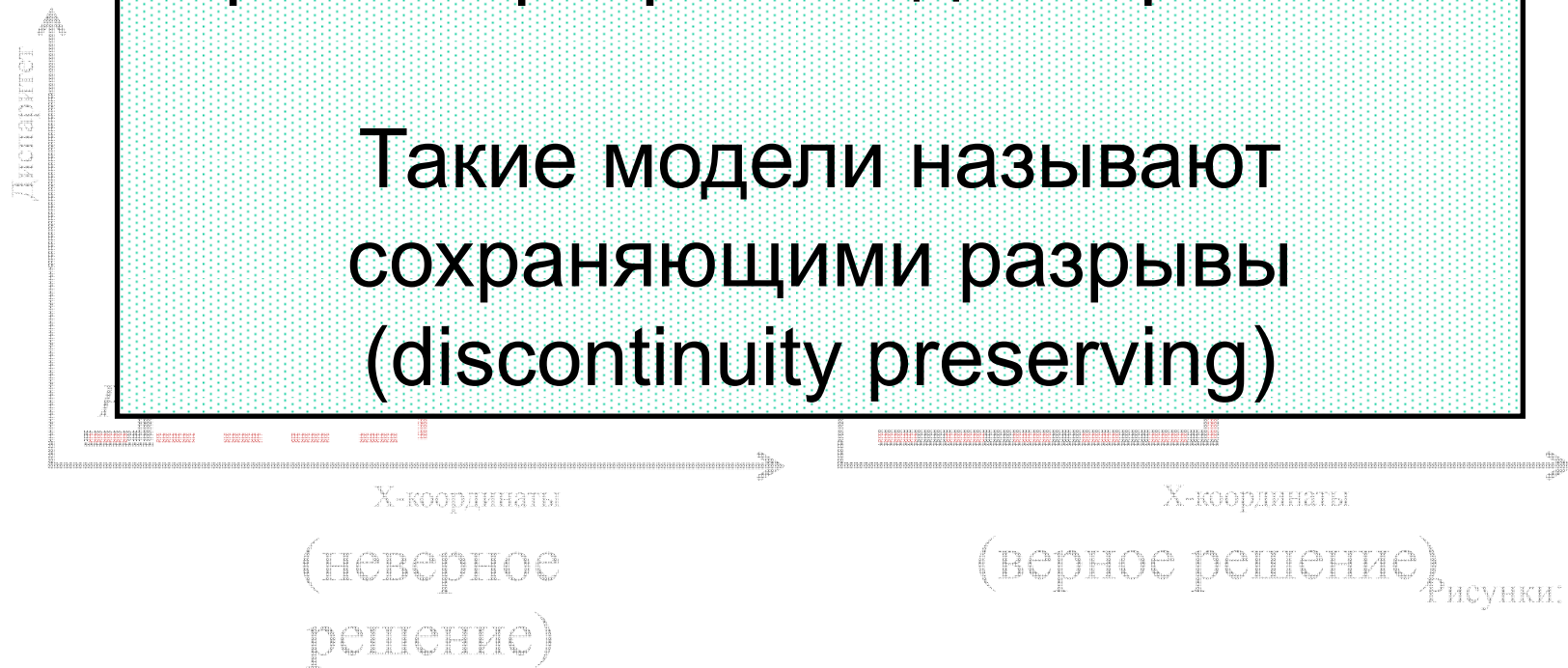


# Проблема сохранения границ

Случай модели Поттса:  $s(d_p, d_q) = \begin{cases} 0, d_p = d_q \\ P, d_p \neq d_q \end{cases}$

Модель Поттса не препятствует  
резким разрывам диспаритета.

Такие модели называют  
сохраняющими разрывы  
(discontinuity preserving)



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 увеличенный 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 [Ishikawa, 2003]
- Но необходимо использовать модели, сохраняющие границы, а они невыпуклы относительно  $|d_p - d_q|$
- Задача становится NP-полной
- Необходимы приближительные алгоритмы



# Используемые подходы

---

- Expansion Move, Swap Move
  - Основаны на Graph Cuts
  - Энергия должна быть субмодулярна
- Fusion Move
  - Основан на QPBO
  - Энергия произвольная
- Loopy Belief Propagation
  - Энергия произвольная
  - Сходимость не гарантируется
- TRW-S
  - Энергия произвольная



# Используемые подходы

---

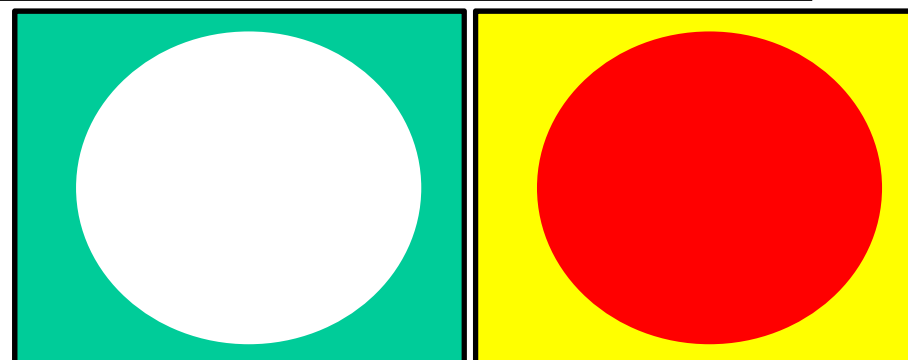
- Expansion Move, Swap Move
  - Основаны на Graph Cuts
  - Энергия должна быть субмодулярна
- Fusion Move
  - Основан на QPBO
  - Энергия произвольная
- Loopy Belief Propagation
  - Энергия произвольная
  - Сходимость не гарантируется
- TRW-S
  - Энергия произвольная



# Алгоритм Fusion Move

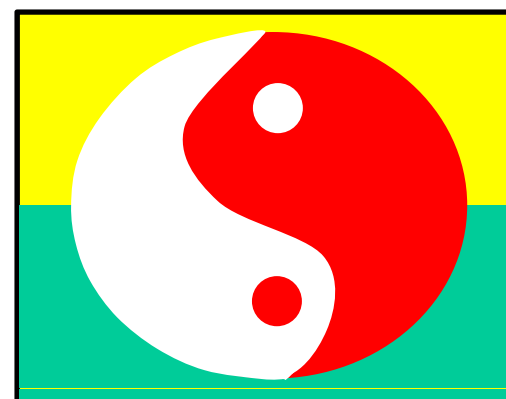
Два произвольных решения  
«смешиваются», энергия  
уменьшается

$\alpha$ -расширение – частный  
случай алгоритма Fusion  
Move, при котором второе  
решение включает только  
метку  $\alpha$



Решение 1

Решение 2



Одна из возможных разметок после  
смешения решений 1 и 2

V. Lempitsky, C. Rother, A. Blake. LogCut - Efficient Graph Cut Optimization for Markov Random Fields. *ICCV*, 2007.



# Fusion Move для стерео

---

- Возможность параллельных вычислений
  - Одно ядро работает с диапазоном диспаратетов 0..8
  - Другое ядро – с диапазоном 9..16
  - Результаты смешиваются с помощью Fusion Move
- Возможность комбинировать сильные стороны разных алгоритмов
  - Один алгоритм хорошо работает на границах объектов и плохо в однородных областях
  - Другой – наоборот
  - Результаты смешиваются
- Возможность оптимизации непрерывных параметров



# Минимизация энергии

---

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





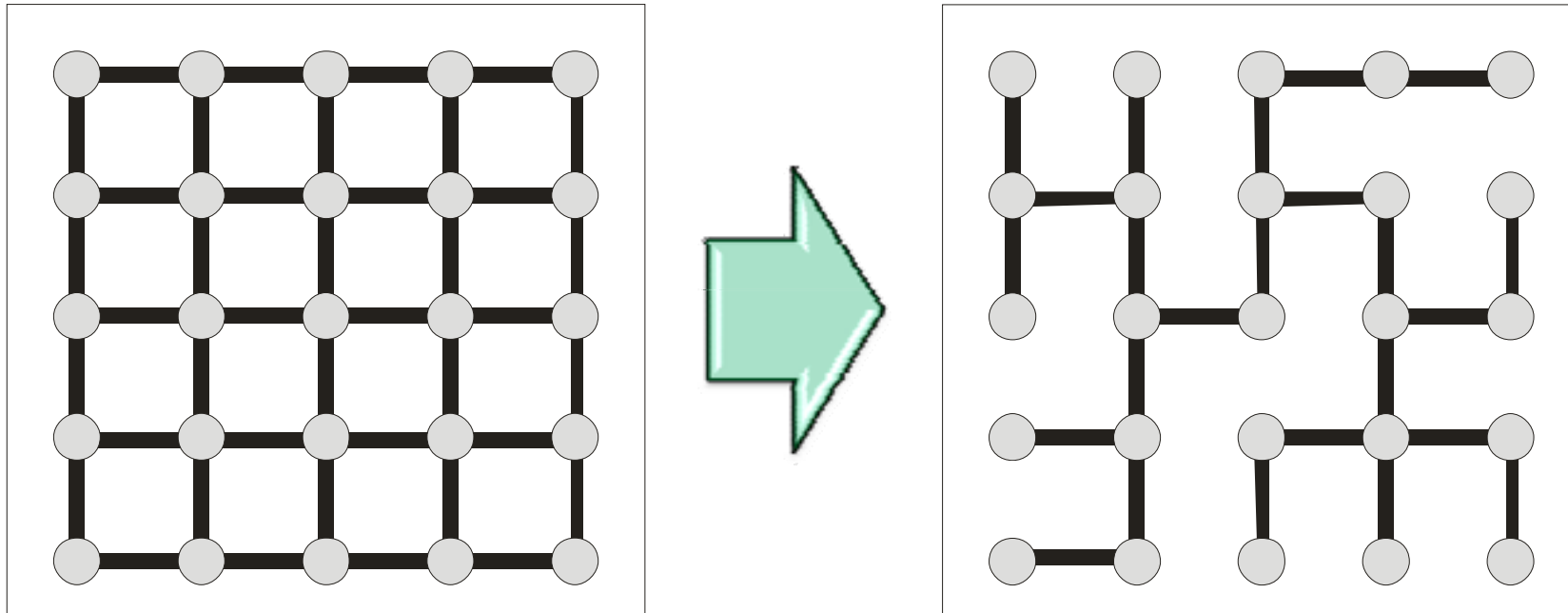
# Минимизация энергии

- Плотное стерео — задача многоклассовой разметки
- Эффективное решение на графе общего вида существует лишь для выпуклых относительно  $|d_p - d_q|$  парных потенциалов [Ishikawa, 2003]
- **Альтернативный подход заключается в уходе от графов общего вида**
- Необходимы приближительные алгоритмы

Н. Ishikawa. Exact Optimization for Markov Random Fields with Convex Priors. *PAMI*, 2003.



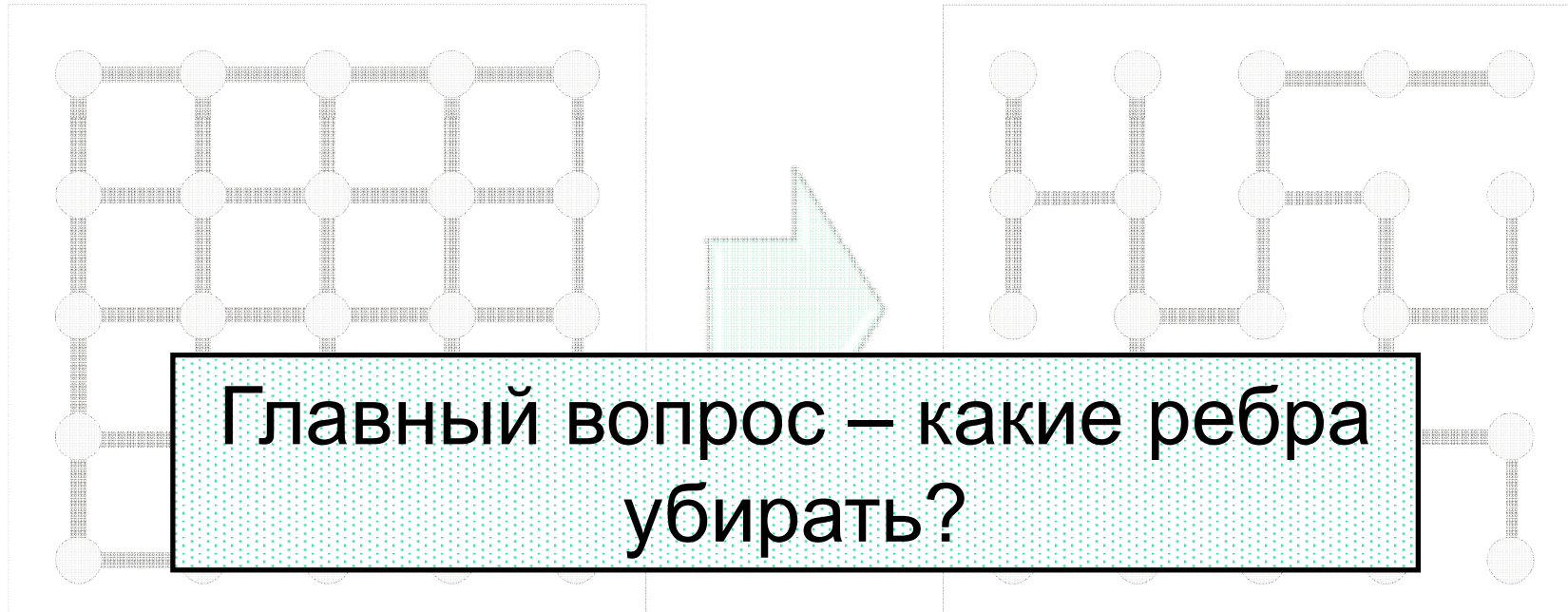
# Переход к деревьям



- Отсутствие циклов позволяет использовать метод динамического программирования
  - Глобальный минимум, произвольная энергия, высокая скорость работы



# Переход к деревьям



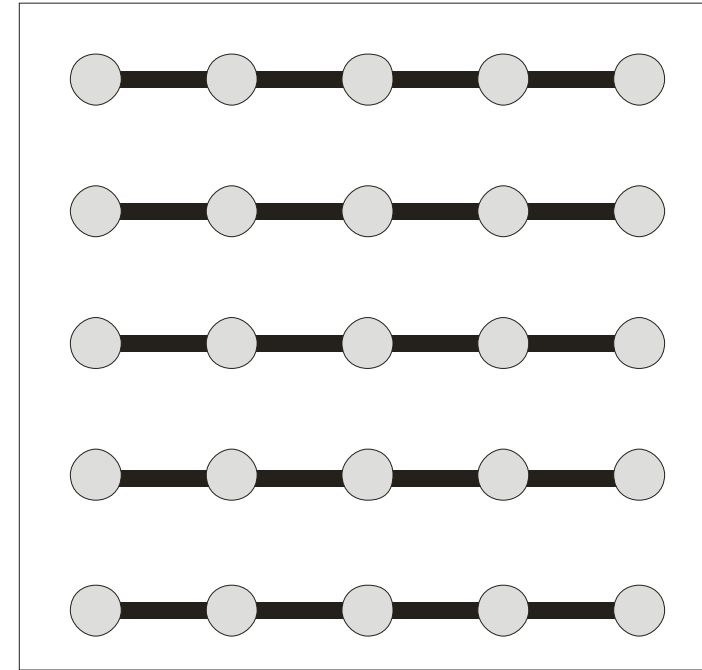
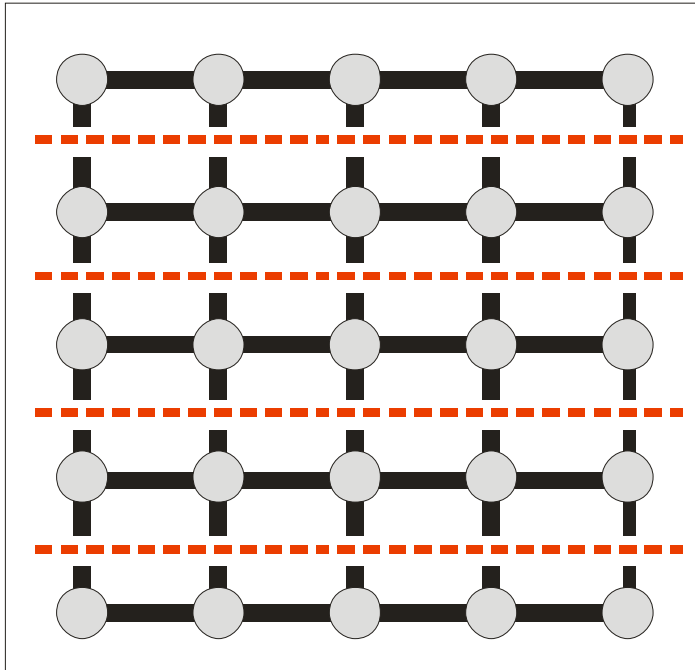
- Отсутствие циклов позволяет использовать метод динамического программирования
  - Глобальный минимум, произвольная энергия, высокая скорость работы

Рисунки: M. Bleyer



# Алгоритм Scanline Optimization

---



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



# Алгоритм Scanline Optimization

---



Очевидная проблема – рассогласованность строк между собой (horizontal streaking)



# 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)

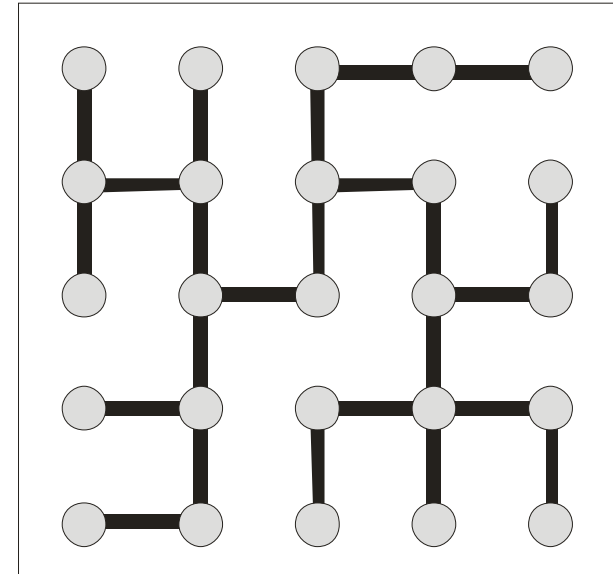


Рисунок: M. Bleyer



# 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

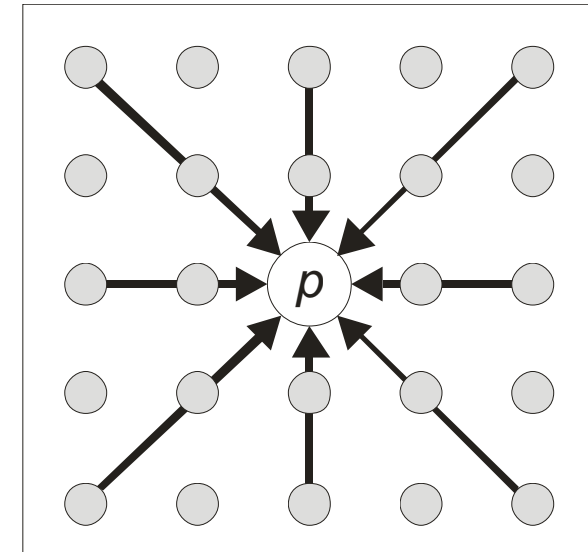


Рисунок: M. Bleyer





# Алгоритм Semi-Global Matching

---



- Рассогласованности нет, но есть «изолированные» пиксели
- Алгоритм заточен под работу с аэрофотоснимками



# Алгоритм Semi-Global Matching

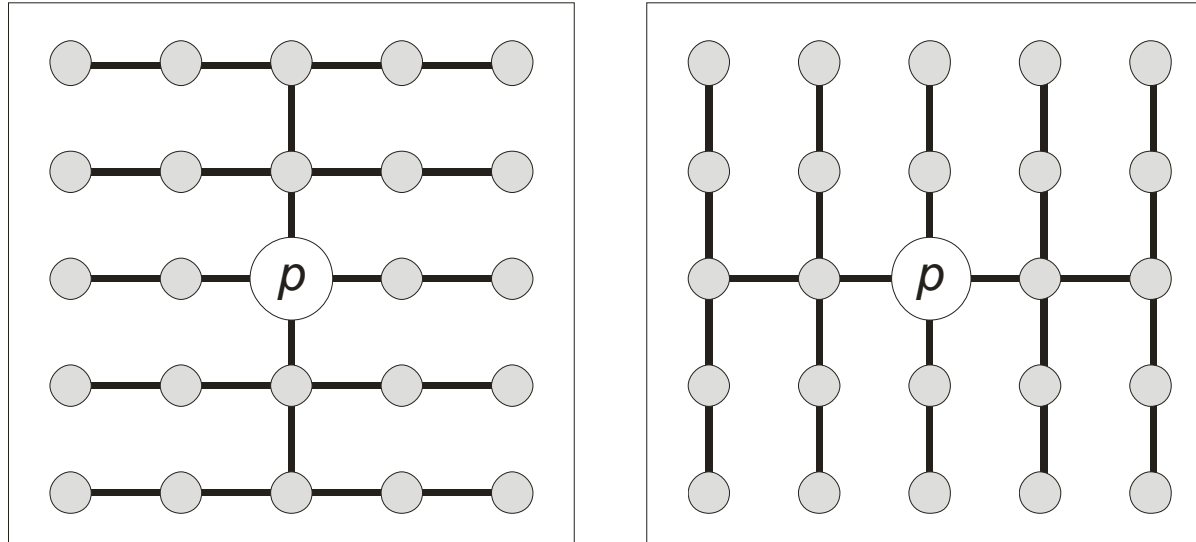
---



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



# Алгоритм Simple Tree



Рисунки: М. Bleyer

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

М. Bleyer, М. 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

- Модифицируем унарный потенциал:

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

Как задать функцию  $O(p)$ ?

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

где:

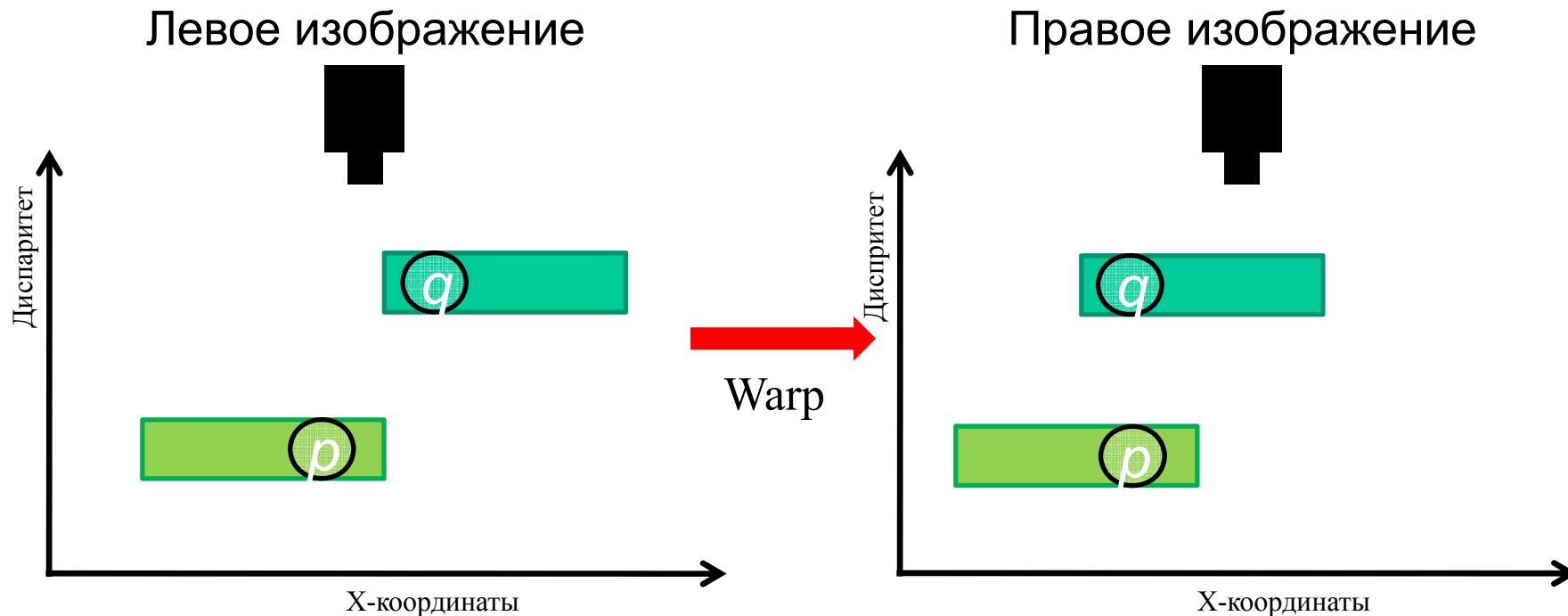
$O(p)$  — функция, возвращающая 1, если пиксель принадлежит области перекрытия, иначе 0

$P_{occ}$  — штраф за перекрытие



# Ограничение видимости

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





# Модифицированная энергия

---

- 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

---

- Наилучшие по качеству методы на сегодняшний день
- Методы, основанные на динамическом программировании, - отличный компромисс между качеством и скоростью
- Узкое место глобальных методов – функционалы энергии, а не алгоритмы оптимизации



# План

---

- Введение
- Формулировка задачи
- Трудности и предположения
- Локальные методы
- Глобальные методы
- **Использование сегментации**
- Резюме



# Использование сегментации

---





# Использование сегментации

---

- Очень популярно в последние годы
- Методы с сегментацией дают наилучшие результаты на стандартных наборах данных
  - Вероятно, проблема в «переобучении» на наборах Middlebury
  - Явное доминирование в верхней части рейтинга

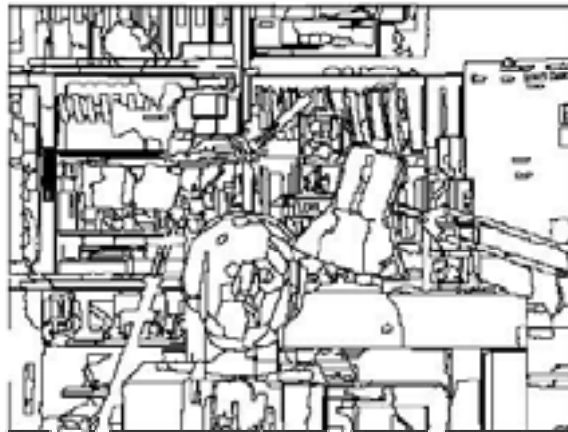


# Использование сегментации

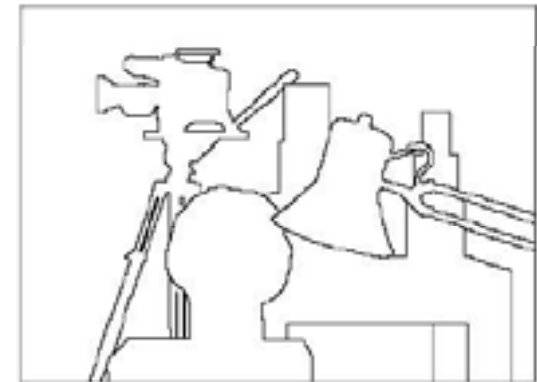
- Используется предположение о сегментации («области разрыва диспаритета совпадают с краями на изображении»)
- Для его реализации применяют пересегментацию (т.е. сегменты достаточно мелкие, «с запасом»)



Исходное  
изображение



Результат  
сегментации



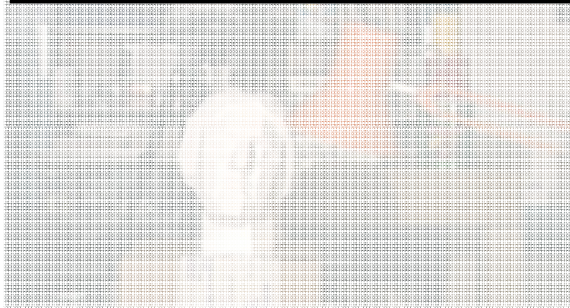
Границы объектов



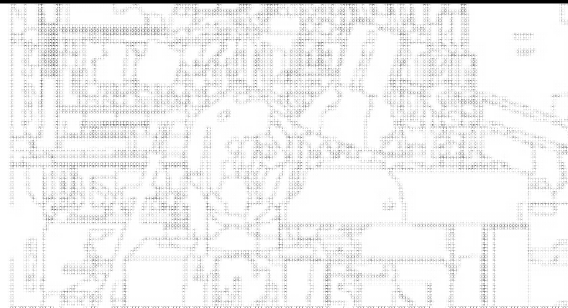
# Использование сегментации

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

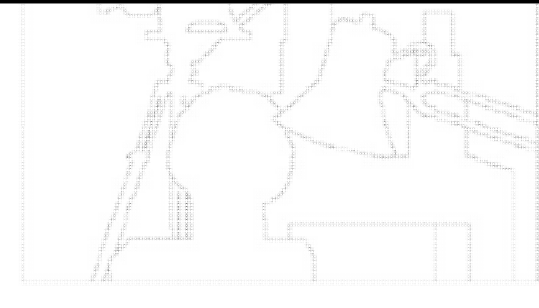
Происходит переход из пиксельного пространства в пространство сегментов. Гладкость внутри сегментов форсируется



Исходное  
изображение



Результат сегментации



Границы объектов



# Базовый алгоритм с сегментацией

---

- Произвести сегментацию
  - Mean-shift, EDISON и др.
- Инициализировать решение
  - Любой локальный алгоритм на пикселях
- Аппроксимировать сегменты гладкими поверхностями
  - Модель: плоскость, B-сплайн
  - Средство: RANSAC, голосование и т.д.
- Уточнить разметку 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





# План

---

- Введение
- Формулировка задачи
- Трудности и предположения
- Локальные методы
- Глобальные методы
- Использование сегментации
- **Резюме**



# Что не вошло

---

- Учет цвета в стерео
- Борьба с перепадами яркости
- Совмещение задач стерео и матирования
- Обучение парных потенциалов
- ...



# Выводы

---

- Много уже сделано, но задача не решена
- Главный компромисс: скорость либо качество
- Требуется расширение тестовых выборок
- Явный тренд в сторону глобальных алгоритмов



# На следующих лекциях

---

- Многовидовая реконструкция
  - Многовидовое стерео
  - Материал и текстура
  - Активное стерео
  - Лазерное сканирование
- Реконструкция по 1 виду
  - Линии схода и их вычисление
  - Интерактивные методы реконструкции
  - Автоматическая реконструкция по 1 кадру

