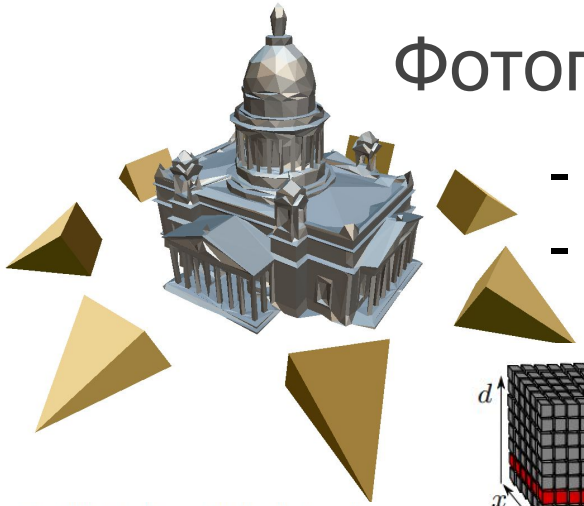
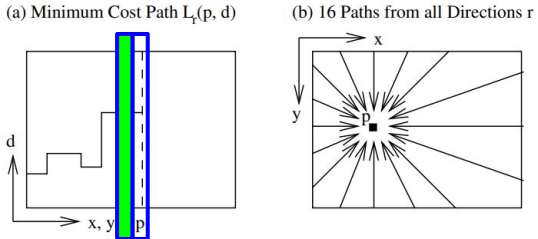
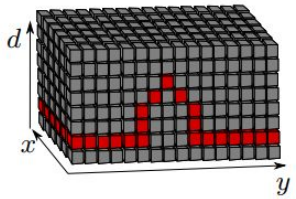
Введение в фотограмметрию
Построение карт глубины
Метод Semi-Global Matching (**SGM**)

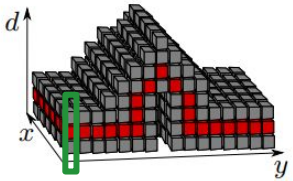
Фотограмметрия. Лекция 12

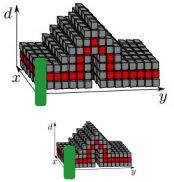
- Карты глубины Semi-Global Matching
- Иерархичный **tSGM**

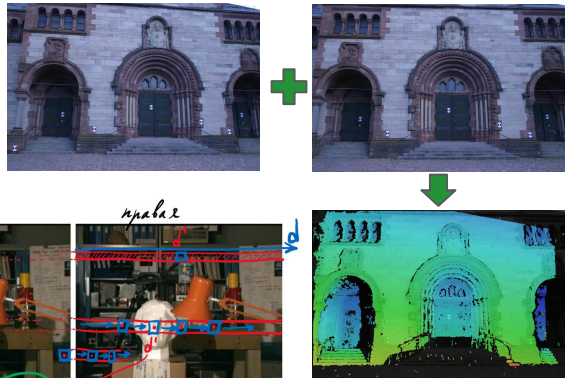




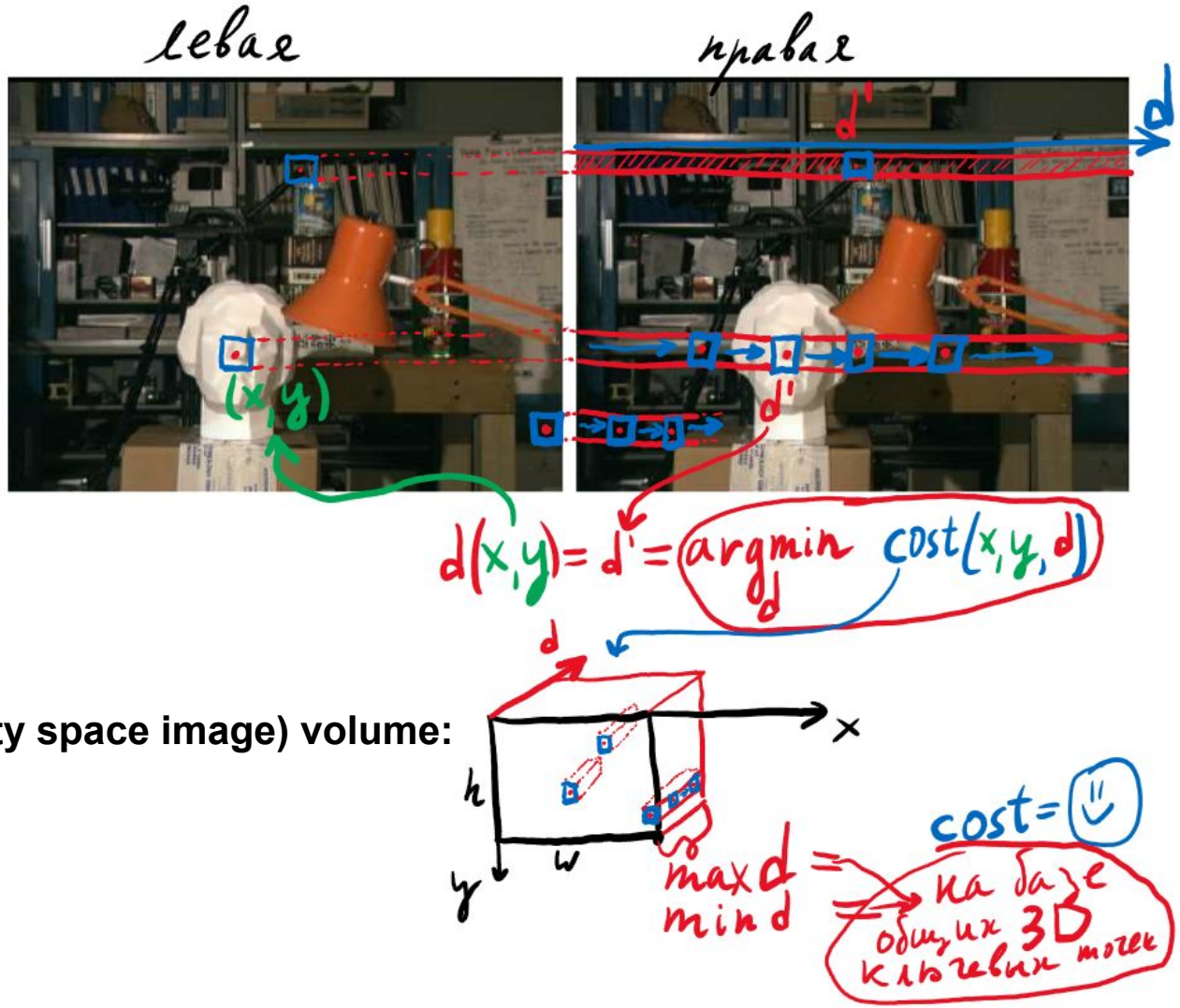








Полярный Николай
polarnick239@gmail.com



DSI (disparity space image) volume:

Как добиться субпиксельной точности?

Как добиться субпиксельной точности?

- При ректификации делать небольшое увеличение картинки
- Parabola fitting

Как добиться субпиксельной точности?

- При ректификации делать небольшое увеличение картинки
- Parabola fitting

Как пофильтровать ошибки (выбросы) и заслоненности (occlusions)?

Как добиться субпиксельной точности?

- При ректификации делать небольшое увеличение картинки
- Parabola fitting

Как пофильтровать ошибки (выбросы) и заслоненности (occlusions)?

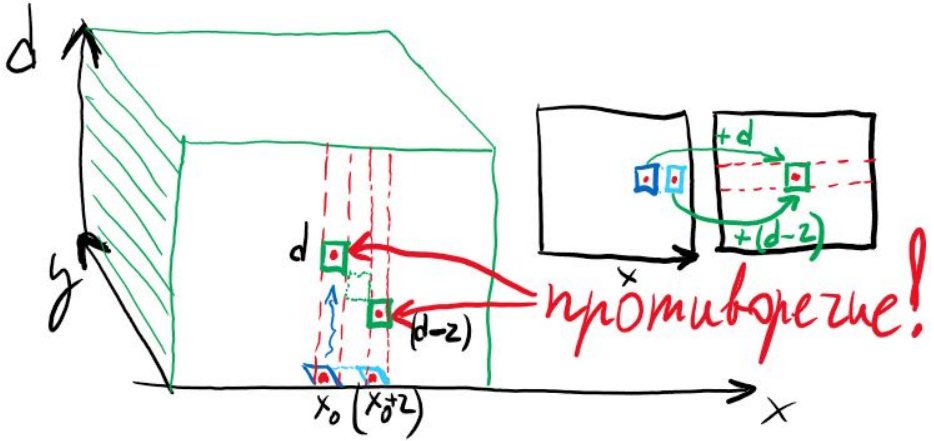
- **Left-right check**: построить обе карты глубины и сверять что переходы симметричны (сумма диспаритета меньше пикселя)

Как добиться субпиксельной точности?

- При ректификации делать небольшое увеличение картинки
- Parabola fitting

Как пофильтровать ошибки (выбросы) и заслоненности (occlusions)?

- **Left-right check**: построить обе карты глубины и сверять что переходы симметричны (сумма диспаритета меньше пикселя)
- Проверяем **one-to-one** mapping через диагональ в DSI volume:

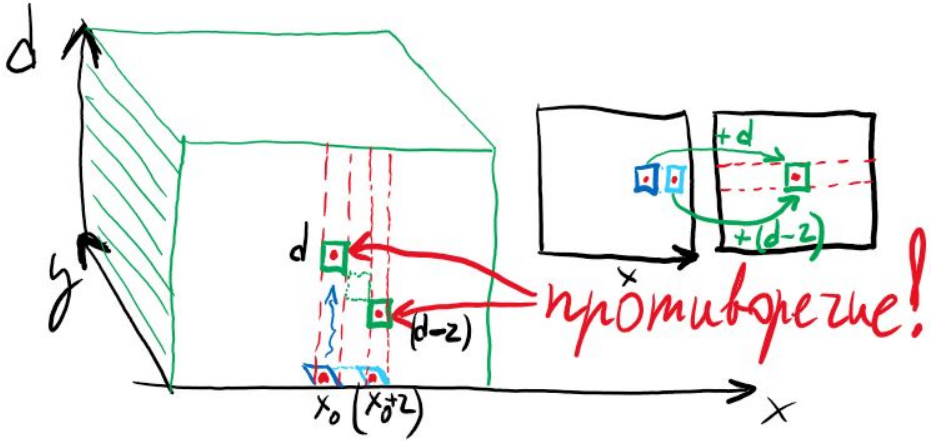


Как добиться субпиксельной точности?

- При ректификации делать небольшое увеличение картинки
- Parabola fitting

Как пофильтровать ошибки (выбросы) и заслоненности (occlusions)?

- **Left-right check**: построить обе карты глубины и сверять что переходы симметричны (сумма диспаритета меньше пикселя)
- Проверяем **one-to-one** mapping через диагональ в DSI volume:



А зачем это нужно когда есть Left-right check?

Как добиться субпиксельной точности?

- При ректификации делать небольшое увеличение картинки
- Parabola fitting

Как пофильтровать ошибки (выбросы) и заслоненности (occlusions)?

- **Left-right check**: построить обе карты глубины и сверять что переходы симметричны (сумма диспаритета меньше пикселя)
- Проверяем **one-to-one** mapping через диагональ в DSI volume:

Как справляться с неоднозначностями - повторяющимися, регулярными и слабо текстурированными поверхностями?

Как добиться субпиксельной точности?

- При ректификации делать небольшое увеличение картинки
- Parabola fitting

Как пофильтровать ошибки (выбросы) и заслоненности (occlusions)?

- **Left-right check**: построить обе карты глубины и сверять что переходы симметричны (сумма диспаритета меньше пикселя)
- Проверяем **one-to-one** mapping через диагональ в DSI volume:

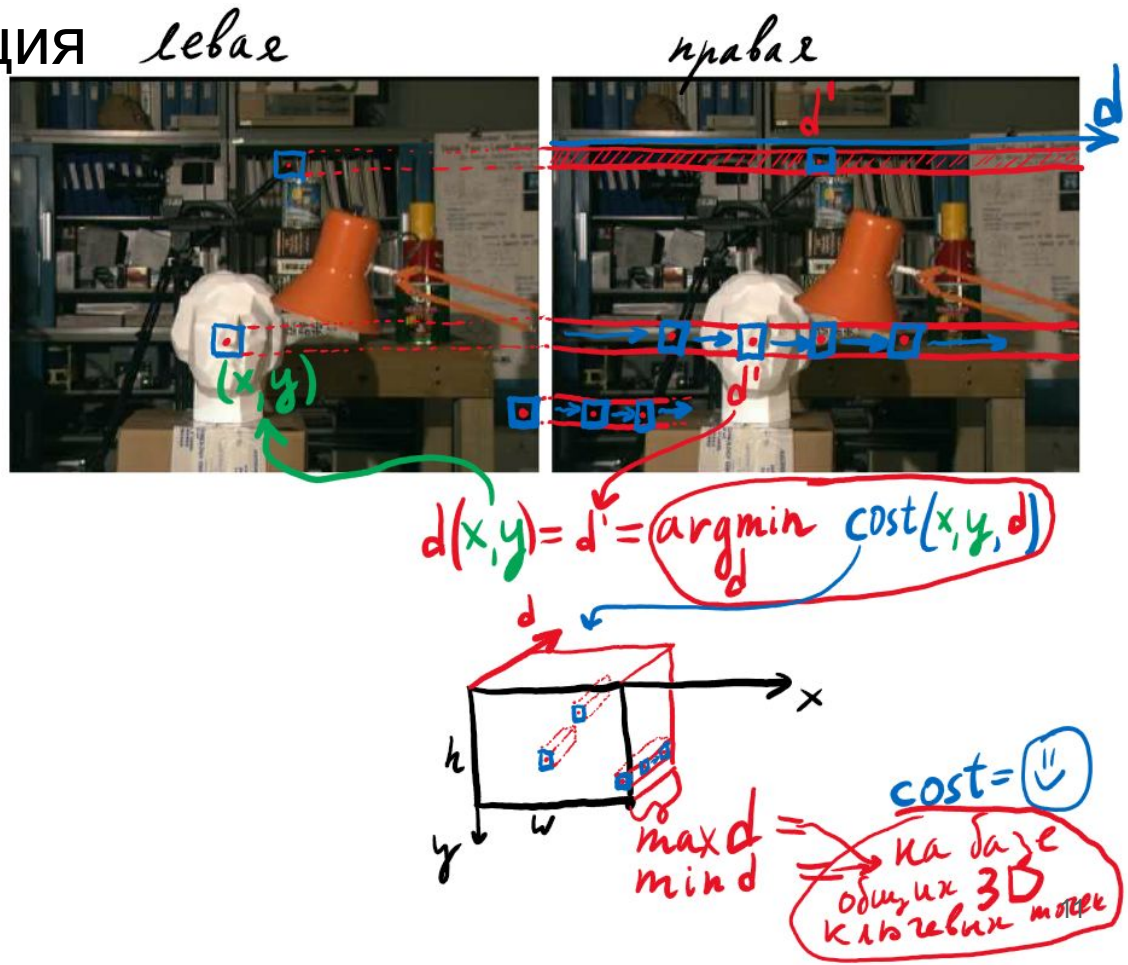
Как справляться с неоднозначностями - повторяющимися, регулярными и слабо текстурированными поверхностями?

- Решать задачу глобальной оптимизации
- Предпочитать “связные” диспаритеты (хотим гладкость)
- Штрафовать за отличающиеся диспаритеты (хотим мало разрывов)

Глобальная оптимизация

Минимизируем энергию:

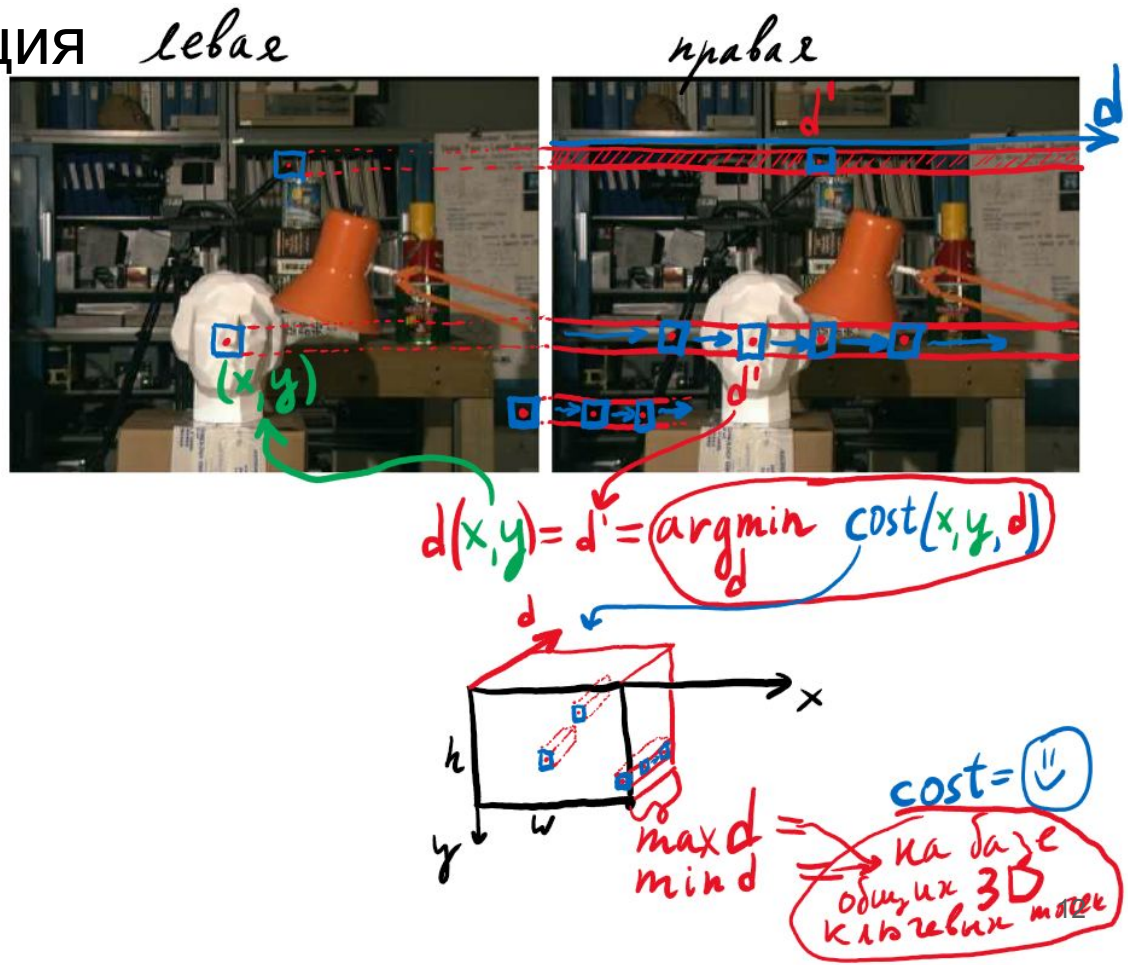
$$E(d) = \text{???}$$



Глобальная оптимизация

Минимизируем энергию:

$$E(d) = E_{\mathrm{D}}(d) + \lambda E_{\mathrm{S}}(d)$$



Глобальная оптимизация

Минимизируем энергию: $E(d) = E_{\mathrm{D}}(d) + \lambda E_{\mathrm{S}}(d)$

Где **data term**: $$E_{\mathrm{D}}(d) = \sum_{(x,y)} C(x, y, d(x, y))$$

Глобальная оптимизация

Минимизируем энергию: $E(d) = E_{\mathrm{D}}(d) + \lambda E_{\mathrm{S}}(d)$

Где **data term**: $E_{\mathrm{D}}(d) = \sum_{(x,y)} C(x, y, d(x, y))$

И **smoothness term**:

$$E_{\mathrm{S}}(d) = \sum_{(x,y)} \rho(d(x, y) - d(x + 1, y)) + \rho(d(x, y) - d(x, y + 1))$$

где ρ - монотонная функция штрафа

Глобальная оптимизация

Минимизируем энергию: $E(d) = E_{\mathrm{D}}(d) + \lambda E_{\mathrm{S}}(d)$

Где **data term**: $E_{\mathrm{D}}(d) = \sum_{(x,y)} C(x, y, d(x, y))$

И **smoothness term**:

$$E_{\mathrm{S}}(d) = \sum_{(x,y)} \rho(d(x, y) - d(x + 1, y)) + \rho(d(x, y) - d(x, y + 1))$$

где ρ - монотонная функция штрафа

Как ослабить штраф за разрыв на границе объекта?

Глобальная оптимизация

Минимизируем энергию: $E(d) = E_{\mathrm{D}}(d) + \lambda E_{\mathrm{S}}(d)$

Где **data term**: $E_{\mathrm{D}}(d) = \sum_{(x,y)} C(x, y, d(x, y))$

И **smoothness term**:

$$E_{\mathrm{S}}(d) = \sum_{(x,y)} \rho(d(x, y) - d(x + 1, y)) + \rho(d(x, y) - d(x, y + 1))$$

где ρ - монотонная функция штрафа

Как ослабить штраф за разрыв на границе объекта?

Домножать с учетом перепада яркости (или силы градиента):

$$\rho_{\mathrm{D}}(d(x, y) - d(x + 1, y)) \cdot \rho_{\mathrm{I}}(\|I(x, y) - I(x + 1, y)\|)$$

Глобальная оптимизация

Минимизируем энергию: $E(d) = E_{\mathrm{D}}(d) + \lambda E_{\mathrm{S}}(d)$

Где **data term**: $E_{\mathrm{D}}(d) = \sum_{(x,y)} C(x, y, d(x, y))$

И **smoothness term**:

$$E_{\mathrm{S}}(d) = \sum_{(x,y)} \rho(d(x, y) - d(x + 1, y)) + \rho(d(x, y) - d(x, y + 1))$$

где ρ - монотонная функция штрафа

Это **NP**-полная задача, приближенное решение можно искать с помощью разных методов, например через **Markov Random Fields (MRF)** или минимальный разрез графа.

Глобальная оптимизация

Минимизируем энергию: $E(d) = E_{\mathrm{D}}(d) + \lambda E_{\mathrm{S}}(d)$

Где **data term**: $E_{\mathrm{D}}(d) = \sum_{(x,y)} C(x, y, d(x, y))$

И **smoothness term**:

$$E_{\mathrm{S}}(d) = \sum_{(x,y)} \rho(d(x, y) - d(x + 1, y)) + \rho(d(x, y) - d(x, y + 1))$$

где ρ - монотонная функция штрафа

Это **NP**-полная задача, приближенное решение можно искать с помощью разных методов, например через **Markov Random Fields (MRF)** или минимальный разрез графа. **Но это медленно!**

Semi-Global Matching (**SGM**)

$$E(D) = \sum_{\mathbf{p}} \left[C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1] \right]$$

все пиксели

локальные costs

соседние пиксели p

плата за отличие disparity на 1

штраф за разрыв (отличие ≥ 1)

Semi-Global Matching (**SGM**)

$$E(D) = \sum_{\mathbf{p}} \left[C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1] \right]$$

все пиксели

локальные costs

соседние пиксели p

плата за отличие disparity на 1

штраф за разрыв (отличие ≥ 1)

Semi-Global Matching (**SGM**)

$$E(D) = \sum_{\mathbf{p}} \left[C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1] \right]$$

все пиксели

локальные costs

соседние пиксели p

плата за отличие disparity на 1

штраф за разрыв (отличие ≥ 1)

Semi-Global Matching (**SGM**)

$$E(D) = \sum_{\mathbf{p}} \left[C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1] \right]$$

все пиксели

локальные costs

соседние пиксели p

плата за отличие disparity на 1

штраф за разрыв (отличие ≥ 1)

Как соотносятся штрафы P1 и P2? Кто больше?

Semi-Global Matching (**SGM**)

$$E(D) = \sum_{\mathbf{p}} \left[C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1] \right]$$

все пиксели

локальные costs

соседние пиксели p

плата за отличие disparity на 1

штраф за разрыв (отличие ≥ 1)

$P_2 \geq P_1$

Как добавить послабление на границах объектов?
Как снизить там штрафы за разрывы?

Semi-Global Matching (**SGM**)

$$E(D) = \sum_{\mathbf{p}} \left[C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1] \right]$$

все пиксели

локальные costs

соседние пиксели p

плата за отличие disparity на 1

штраф за разрыв (отличие ≥ 1)

$P_2 \geq P_1$

можно задать последующие на границах $P_2 = \frac{P_2'}{|I_{b\mathbf{p}} - I_{b\mathbf{q}}|}$

Semi-Global Matching (**SGM**)

$$E(D) = \sum_{\mathbf{p}} C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1]$$

У пикселя 8 соседей, из-за взаимного влияния - вычислительно тяжело.

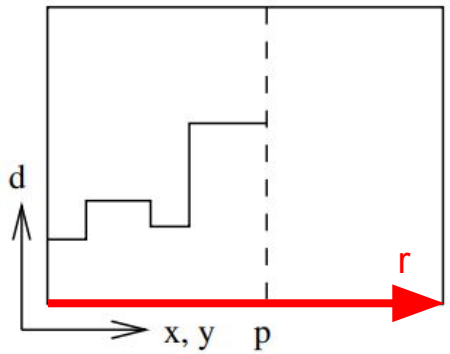
Semi-Global Matching (**SGM**)

$$E(D) = \sum_{\mathbf{p}} C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1]$$

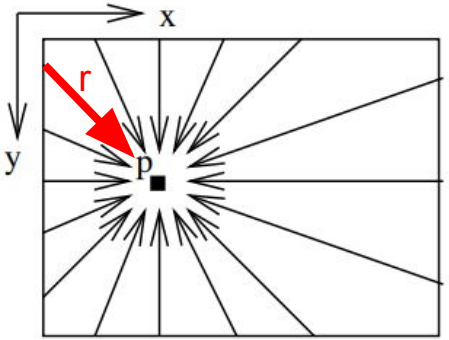
У пикселя 8 соседей, из-за взаимного влияния - вычислительно тяжело.

Давайте фиксируем одно направление и будем учитывать только соседа по этому направлению **r**:

(a) Minimum Cost Path $L_r(p, d)$



(b) 16 Paths from all Directions r



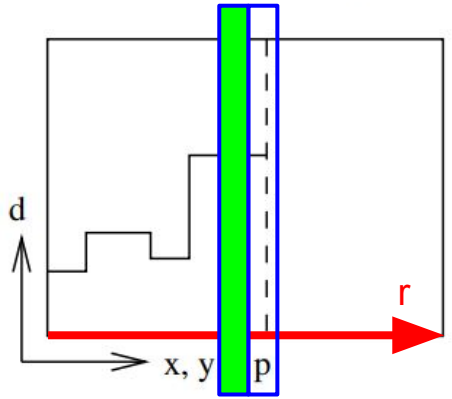
Semi-Global Matching (**SGM**)

$$E(D) = \sum_{\mathbf{p}} C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1]$$

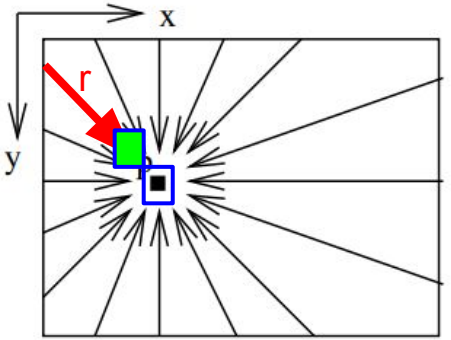
У пикселя 8 соседей, из-за взаимного влияния - вычислительно тяжело.

Давайте фиксируем одно направление и будем учитывать только соседа по этому направлению **r**:

(a) Minimum Cost Path $L_r(p, d)$



(b) 16 Paths from all Directions r



Semi-Global Matching (**SGM**)

$$E(D) = \sum_{\mathbf{p}} C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1]$$

У пикселя 8 соседей, из-за взаимного влияния - вычислительно тяжело.

Давайте фиксируем одно направление и будем учитывать только соседа по этому направлению **r**:

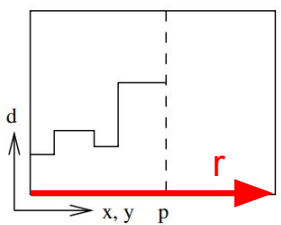
$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min\big(L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d),$$
$$L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d-1) + P_1, L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d+1) + P_1,$$
$$\min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2\big) - \min_k L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, k)$$

cost

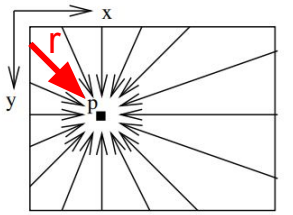
направление

пиксель шаг r назад

(a) Minimum Cost Path $L_r(p, d)$



(b) 16 Paths from all Directions r



Semi-Global Matching (**SGM**)

$$E(D) = \sum_{\mathbf{p}} C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1]$$

У пикселя 8 соседей, из-за взаимного влияния - вычислительно тяжело.

Давайте фиксируем одно направление и будем учитывать только соседа по этому направлению **r**:

$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min\Big(L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, d),\; L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, d-1) + P_1,\; L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, d+1) + P_1,\; \min_i L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, i) + P_2\Big) - \min_k L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, k)$$

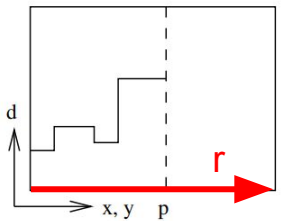
cost

направление

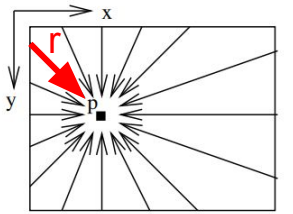
то же смещение

пиксель шаг r назад

(a) Minimum Cost Path $L_r(p, d)$



(b) 16 Paths from all Directions r



Semi-Global Matching (**SGM**)

$$E(D) = \sum_{\mathbf{p}} C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1]$$

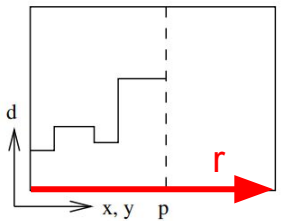
У пикселя 8 соседей, из-за взаимного влияния - вычислительно тяжело.

Давайте фиксируем одно направление и будем учитывать только соседа по этому направлению **r**:

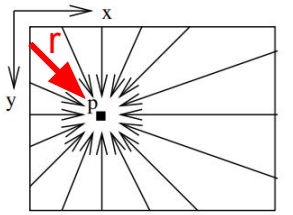
$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min\Big(L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d),\; L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d-1) + P_1,\; L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d+1) + P_1,\; \min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2\Big) - \min_k L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, k)$$

cost; направление; соседнее смещение; пиксель шаг r назад

(a) Minimum Cost Path $L_r(p, d)$



(b) 16 Paths from all Directions r



Semi-Global Matching (**SGM**)

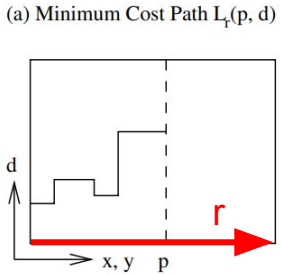
$$E(D) = \sum_{\mathbf{p}} C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \, \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1]$$

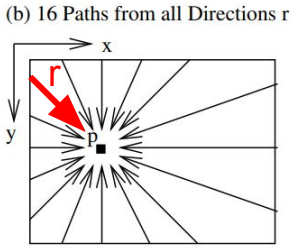
У пикселя 8 соседей, из-за взаимного влияния - вычислительно тяжело.

Давайте фиксируем одно направление и будем учитывать только соседа по этому направлению **r**:

$$L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min\big(L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d),$$
$$L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d-1) + P_1, L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d+1) + P_1,$$
$$\min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2\big) - \min_k L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, k)$$

cost; направление; разрыв; против переполнения; пиксель шаг r назад





Semi-Global Matching (**SGM**)

1) Ректифицировали стереопару (ради гарантии на горизонтальные сдвиги)

Semi-Global Matching (**SGM**)

1) Ректифицировали стереопару (ради гарантии на горизонтальные сдвиги)
2) Преподсчитали **Census** для каждого пикселя у обеих фотографий

Semi-Global Matching (**SGM**)

1) Ректифицировали стереопару (ради гарантии на горизонтальные сдвиги)
2) Преподсчитали **Census** для каждого пикселя у обеих фотографий
3) **Matching cost computation**: посчитали cost во всем DSI volume

Semi-Global Matching (**SGM**)

1) Ректифицировали стереопару (ради гарантии на горизонтальные сдвиги)
2) Преподсчитали **Census** для каждого пикселя у обеих фотографий
3) **Matching cost computation**: посчитали cost во всем DSI volume
4) **Cost support aggregation**: прошли 16-ю волнами - оценили энергию во всех гипотезах всего объема

Semi-Global Matching (**SGM**)

1) Ректифицировали стереопару (ради гарантии на горизонтальные сдвиги)
2) Преподсчитали **Census** для каждого пикселя у обеих фотографий
3) **Matching cost computation**: посчитали cost во всем DSI volume
4) **Cost support aggregation**: прошли 16-ю волнами - оценили энергию во всех гипотезах всего объема
5) **Winner takes all (WTA)**: для каждого пикселя нашли disparity с минимальной энергией - это победитель

Но как объединять энергию ведь у нас 16 разных направлений?

Semi-Global Matching (**SGM**)

1) Ректифицировали стереопару (ради гарантии на горизонтальные сдвиги)
2) Преподсчитали **Census** для каждого пикселя у обеих фотографий
3) **Matching cost computation**: посчитали cost во всем DSI volume
4) **Cost support aggregation**: прошли 16-ю волнами - оценили энергию во всех гипотезах всего объема
5) **Winner takes all (WTA)**: для каждого пикселя нашли disparity с минимальной энергией - это победитель
6) **Refinement**: субпиксельно уточнили disparity-победителя с учетом энергии соседей

Semi-Global Matching (**SGM**)

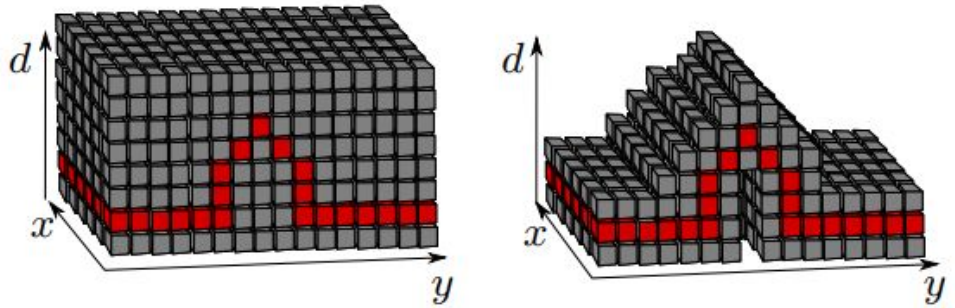
1) Ректифицировали стереопару (ради гарантии на горизонтальные сдвиги)
2) Преподсчитали **Census** для каждого пикселя у обеих фотографий
3) **Matching cost computation**: посчитали cost во всем DSI volume
4) **Cost support aggregation**: прошли 16-ю волнами - оценили энергию во всех гипотезах всего объема
5) **Winner takes all (WTA)**: для каждого пикселя нашли disparity с минимальной энергией - это победитель
6) **Refinement**: субпиксельно уточнили disparity-победителя с учетом энергии соседей
7) **Left-right check** или проверка **one-to-one mapping** по диагонали ради удаления ошибок и регионов где не видит один из кадров (**occlusion**)

Semi-Global Matching (**SGM**)

1) Ректифицировали стереопару (ради гарантии на горизонтальные сдвиги)
2) Преподсчитали **Census** для каждого пикселя у обеих фотографий
3) **Matching cost computation**: посчитали cost во всем DSI volume
4) **Cost support aggregation**: прошли 16-ю волнами - оценили энергию во всех гипотезах всего объема
5) **Winner takes all (WTA)**: для каждого пикселя нашли disparity с минимальной энергией - это победитель
6) **Refinement**: субпиксельно уточнили disparity-победителя с учетом энергии соседей
7) **Left-right check** или проверка **one-to-one mapping** по диагонали ради удаления ошибок и регионов где не видит один из кадров (**occlusion**)

Но **O(W*H*D) памяти!** (и времени) Как сделать лучше?

SURE tSGM



Но **O(W*H*D) памяти!** (и времени) Как сделать лучше?

SURE tSGM

Воспользуемся **coarse-to-fine** схемой!
(т.е. будем прогрессировать по пирамиде детализаций стереопары, постепенно уточняя карту диспаритета)

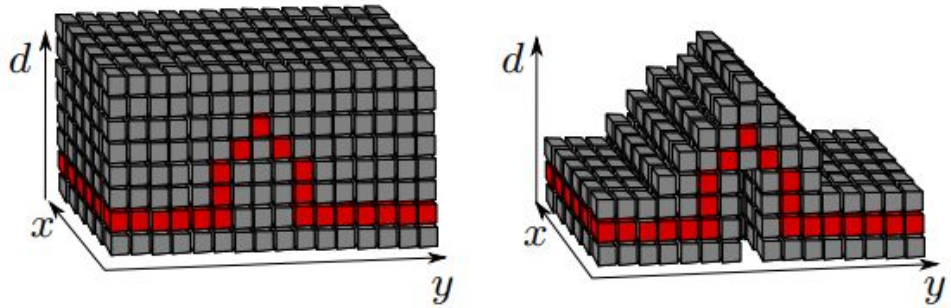
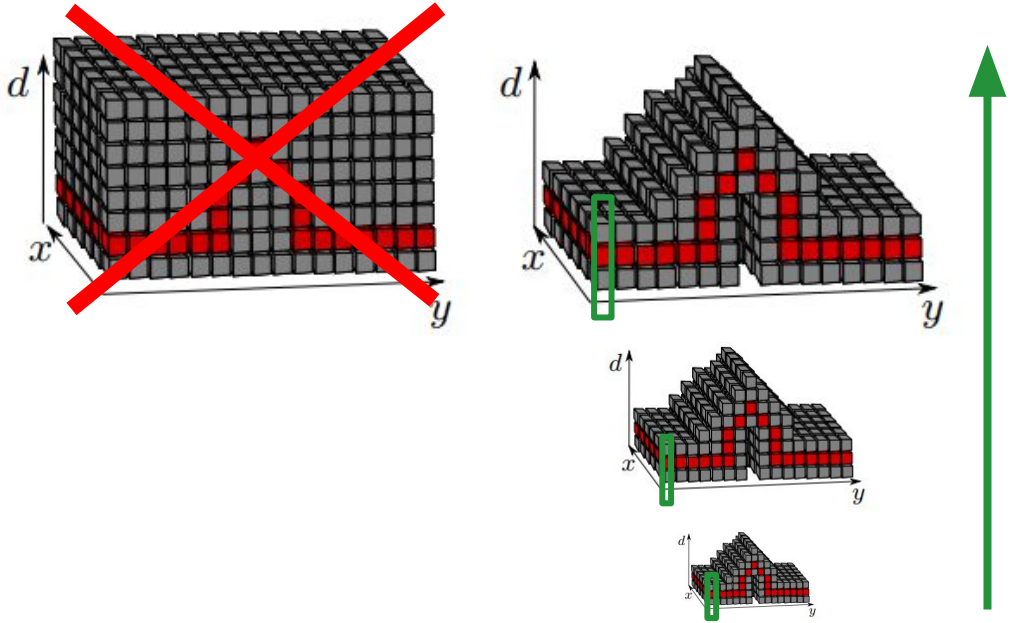


Figure 2: Cost structures of classic SGM (left) and tSGM (right). Red cubes represent costs for the true correspondences. Gray cubes mark the costs of potential correspondences, thus the disparity search ranges.

SURE tSGM

Как при переходе от менее детальной стереопары и карты глубины к более детальной решить в каком диапазоне требуется выполнять поиск?
(т.е. агрегацию энергии по 16 направлениям)



SURE tSGM

Определив карту диспаритета очередного уровня - определяем диапазон дальнейшего поиска:

SURE tSGM

Определив карту диспаритета очередного уровня - определяем диапазон дальнейшего поиска:

1) Если пиксель был успешно сопоставлен - его диапазон это минимум и максимум по диспаритетам в окне 7x7

SURE tSGM

Определив карту диспаритета очередного уровня - определяем диапазон дальнейшего поиска:

1) Если пиксель был успешно сопоставлен - его диапазон это минимум и максимум по диспаритетам в окне 7x7
2) Если пиксель не был сопоставлен - минимум и максимум по диспаритетам в окне 31x31, а в сам пиксель кладем медиану диспаритета по этому же окну 31x31

SURE tSGM

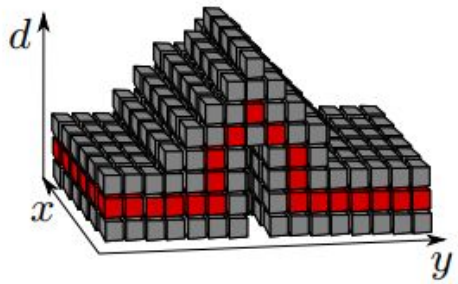
Определив карту диспаритета очередного уровня - определяем диапазон дальнейшего поиска:

1) Если пиксель был успешно сопоставлен - его диапазон это минимум и максимум по диспаритетам в окне 7x7
2) Если пиксель не был сопоставлен - минимум и максимум по диспаритетам в окне 31x31, а в сам пиксель кладем медиану диспаритета по этому же окну 31x31

Ограничиваем диапазон до 16 и 32 соответственно. И домножаем на два т.к. переход детальности

Итого в пространстве поиска диспаритета следующего уровня:

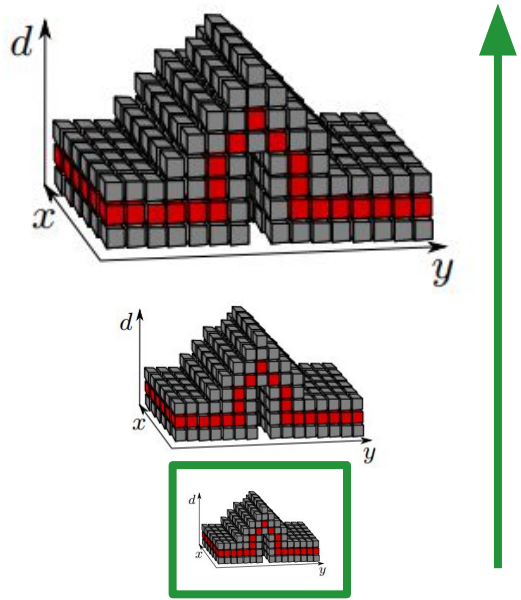
- чаще всего высота столбика - 32 ячейки
- на границах или в сложных местах - 64 ячеек



SURE tSGM

Сэкономили память и ускорили, теперь **O(W*H*64)**

Coarse-to-fine схема - как **мат. индукция**. Нужна **база**! **Как найти первый уровень?**



SURE tSGM

Сэкономили память и ускорили, теперь **O(W*H*64)**

Как найти **первый уровень**?

- Экстраполяцией смещений по сопоставленным ключевым точкам
- Построить с диапазоном поиска размера 64 > ширина самого недетального первого уровня

SURE tSGM

Сэкономили память и ускорили, теперь **O(W*H*64)**

Как найти **первый уровень**?

- Экстраполяцией смещений по сопоставленным ключевым точкам
- Построить с диапазоном поиска размера 64 > ширина самого недетального первого уровня

Бесплатен ли нам по скорости **Left-Right check**?

SURE tSGM

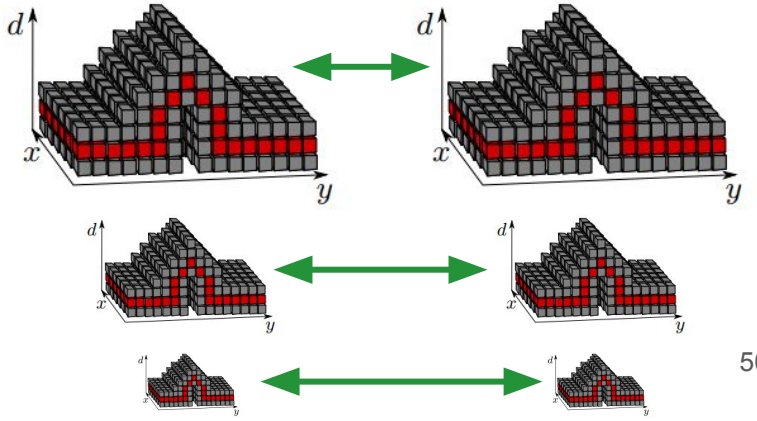
Сэкономили память и ускорили, теперь **O(W*H*64)**

Как найти **первый уровень**?

- Экстраполяцией смещений по сопоставленным ключевым точкам
- Построить с диапазоном поиска размера 64 > ширина самого недетального первого уровня

Бесплатен ли нам по скорости **Left-Right check**?

Левое DSI: Правое DSI:



SURE tSGM

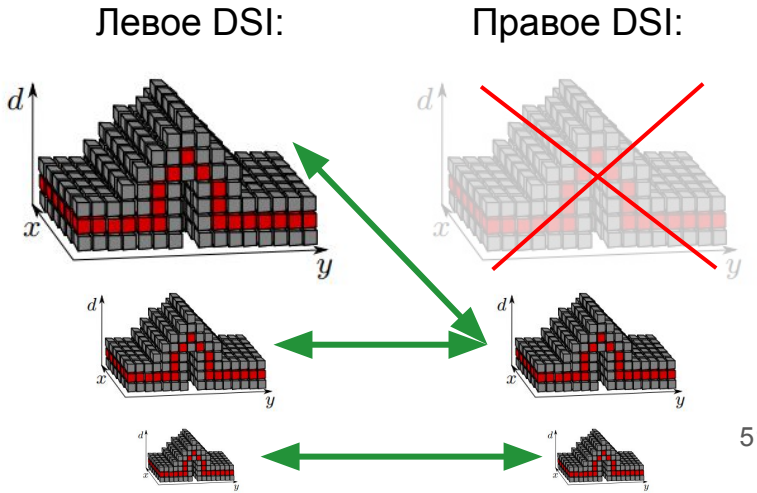
Сэкономили память и ускорили, теперь **O(W*H*64)**

Как найти **первый уровень**?

- Экстраполяцией смещений по сопоставленным ключевым точкам
- Построить с диапазоном поиска размера 64 > ширина самого недетального первого уровня

Как ускорить Left-Rigth check?

- Последний уровень диспаритетов строить только слева
- Сверку на симметрию делать с грубой версией справа - т.е. на один уровень меньше



SURE tSGM

SURE tSGM

SURE tSGM

Итого

1) Всё знали про камеры - положение в пространстве + калибровки

Итого

1) Всё знали про камеры - положение в пространстве + калибровки
2) Знали грубую геометрию сцены - разреженное 3D облако ключевых точек

Итого

1) Всё знали про камеры - положение в пространстве + калибровки
2) Знали грубую геометрию сцены - разреженное 3D облако ключевых точек
3) Строим карты глубины для каждой пары камер:

Итого

1) Всё знали про камеры - положение в пространстве + калибровки
2) Знали грубую геометрию сцены - разреженное 3D облако ключевых точек
3) Строим карты глубины для каждой пары камер:

3.1) **Ректифицировали** - для каждой пары камер построили стереопару:

Итого

1) Всё знали про камеры - положение в пространстве + калибровки
2) Знали грубую геометрию сцены - разреженное 3D облако ключевых точек
3) Строим карты глубины для каждой пары камер:

3.1) **Ректифицировали** - для каждой пары камер построили стереопару:

3.2) Иерархичным **tSGM** предсказывали диапазон поиска диспаритетов

Итого

1) Всё знали про камеры - положение в пространстве + калибровки
2) Знали грубую геометрию сцены - разреженное 3D облако ключевых точек
3) Строим карты глубины для каждой пары камер:

3.1) **Ректифицировали** - для каждой пары камер построили стереопару:

3.2) Иерархичным **tSGM** предсказывали диапазон поиска диспаритетов

3.3) Преподсчитывали **Census** описание патча вокруг пикселя

Итого

1) Всё знали про камеры - положение в пространстве + калибровки
2) Знали грубую геометрию сцены - разреженное 3D облако ключевых точек
3) Строим карты глубины для каждой пары камер:

3.1) **Ректифицировали** - для каждой пары камер построили стереопару:

3.2) Иерархичным **tSGM** предсказывали диапазон поиска диспаритетов

3.3) Преподсчитывали **Census** описание патча вокруг пикселя

3.4) В 16 направлениях считали энергию **Semi-Global Matching (SGM)**

Итого

1) Всё знали про камеры - положение в пространстве + калибровки
2) Знали грубую геометрию сцены - разреженное 3D облако ключевых точек
3) Строим карты глубины для каждой пары камер:

3.1) **Ректифицировали** - для каждой пары камер построили стереопару:

3.2) Иерархичным **tSGM** предсказывали диапазон поиска диспаритетов

3.3) Преподсчитывали **Census** описание патча вокруг пикселя

3.4) В 16 направлениях считали энергию **Semi-Global Matching (SGM)**

3.5) **WTA** - выбирали диспаритет-победитель + **Left-Right check** для фильтрации

Итого

1) Всё знали про камеры - положение в пространстве + калибровки
2) Знали грубую геометрию сцены - разреженное 3D облако ключевых точек
3) Строим карты глубины для каждой пары камер:

3.1) **Ректифицировали** - для каждой пары камер построили стереопару:

3.2) Иерархичным **tSGM** предсказывали диапазон поиска диспаритетов

3.3) Преподсчитывали **Census** описание патча вокруг пикселя

3.4) В 16 направлениях считали энергию **Semi-Global Matching (SGM)**

3.5) **WTA** - выбирали диспаритет-победитель + **Left-Right check** для фильтрации

3.6) По значению диспаритета рассчитывали значение глубины

Итого

1) Всё знали про камеры - положение в пространстве + калибровки
2) Знали грубую геометрию сцены - разреженное 3D облако ключевых точек
3) Строим карты глубины для каждой пары камер:

3.1) **Ректифицировали** - для каждой пары камер построили стереопару:

3.2) Иерархичным **tSGM** предсказывали диапазон поиска диспаритетов

3.3) Преподсчитывали **Census** описание патча вокруг пикселя

3.4) В 16 направлениях считали энергию **Semi-Global Matching (SGM)**

3.5) **WTA** - выбирали диспаритет-победитель + **Left-Right check** для фильтрации

3.6) По значению диспаритета рассчитывали значение глубины

Итого

1) Всё знали про камеры - положение в пространстве + калибровки
2) Знали грубую геометрию сцены - разреженное 3D облако ключевых точек
3) Строим карты глубины для каждой пары камер:

3.1) **Ректифицировали** - для каждой пары камер построили стереопару:

3.2) Иерархичным **tSGM** предсказывали диапазон поиска диспаритетов

3.3) Преподсчитывали **Census** описание патча вокруг пикселя

3.4) В 16 направлениях считали энергию **Semi-Global Matching (SGM)**

3.5) **WTA** - выбирали диспаритет-победитель + **Left-Right check** для фильтрации

3.6) По значению диспаритета рассчитывали значение глубины

Итого

1) Всё знали про камеры - положение в пространстве + калибровки
2) Знали грубую геометрию сцены - разреженное 3D облако ключевых точек
3) Строим карты глубины для каждой пары камер:

3.1) **Ректифицировали** - для каждой пары камер построили стереопару:

3.2) Иерархичным **tSGM** предсказывали диапазон поиска диспаритетов

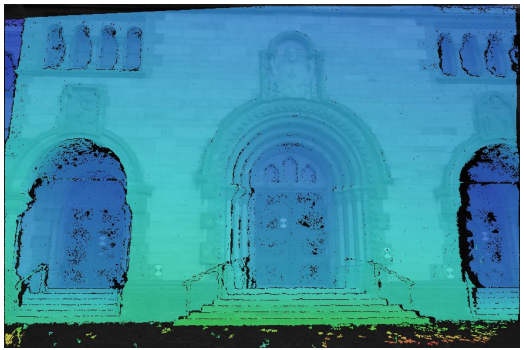
3.3) Преподсчитывали **Census** описание патча вокруг пикселя

3.4) В 16 направлениях считали энергию **Semi-Global Matching (SGM)**

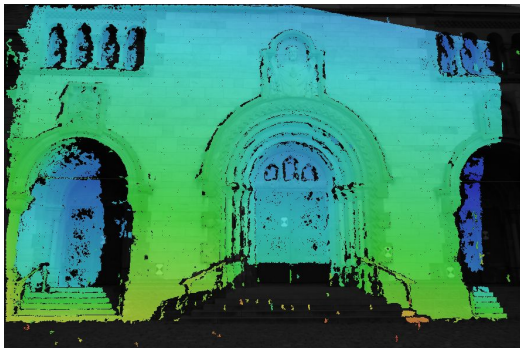
3.5) **WTA** - выбирали диспаритет-победитель + **Left-Right check** для фильтрации

3.6) По значению диспаритета рассчитывали значение глубины

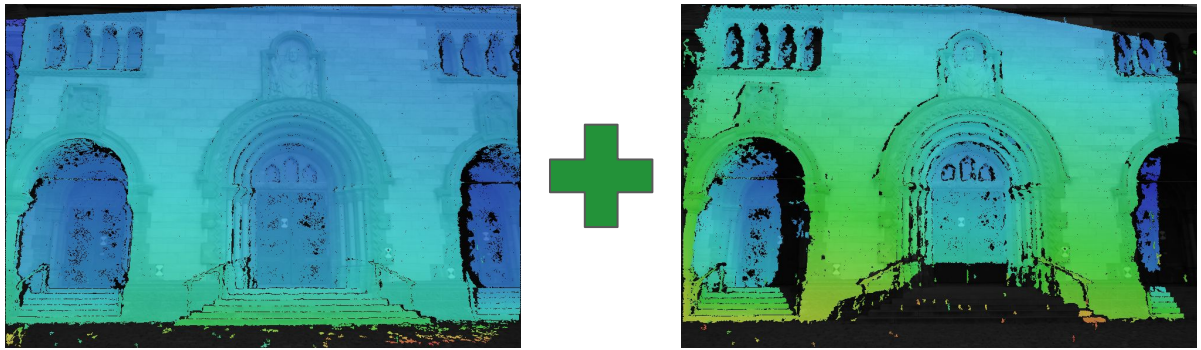
Но ведь мы обрабатывали стерео**пары**!



Но ведь мы обрабатывали стерео**пары**!



Но ведь мы обрабатывали стерео**пары**! Как объединить?



???

Ссылки

Книга (про ректификацию, SGM):

- Computer Vision: Algorithms and Applications, Richard Szeliski

Cost functions:

- Evaluation of Stereo Matching Costs on Images with Radiometric Differences, Hirschmuller, 2008

SGM:

- Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information, Hirschmuller, 2005
- Stereo Processing by Semi-Global Matching and Mutual Information, Hirschmuller, 2008

SURE tSGM:

- SURE: PHOTOGRAMMETRIC SURFACE RECONSTRUCTION FROM IMAGERY, Rothermel et. al., 2013

Вопросы?

Полярный Николай
polarnick239@gmail.com