Введение в фотограмметрию
3D модель из карт глубины

Фотограмметрия. Лекция 16

- **Триангуляция Делоне**
- **Минимальный разрез графа**
- **Полигональная 3D реконструкция**

Полярный Николай
polarnick239@gmail.com


A sandstone block built from lego, blending real objects with 3d prints

A sandstone block built from lego, blending real objects with 3d prints

Где мы сейчас?

- Есть разреженное облако 3D положений **ключевых точек**
- Есть хорошо оптимизированные внутренние калибровки **камер** (intrinsics)
- Есть хорошо оптимизированные положения и ракурсы **камер** (extrinsics)
- Есть (шумные) **карты глубины** описывающие подробно и точно геометрию

Где мы сейчас?

- Есть разреженное облако 3D положений **ключевых точек**
- Есть хорошо оптимизированные внутренние калибровки **камер** (intrinsics)
- Есть хорошо оптимизированные положения и ракурсы **камер** (extrinsics)
- Есть (шумные) **карты глубины** описывающие подробно и точно геометрию

Что хотим дальше?

- Чистить карты глубины от шума на уровне каждой карты глубины (**локально**)
- Чистить карты глубины на базе их противоречий (**глобально**)
- Объединять эти карты глубины в одно общее плотное облако точек
- Строить полигональную 3D модель

Где мы сейчас?

- Есть разреженное облако 3D положений **ключевых точек**
- Есть хорошо оптимизированные внутренние калибровки **камер** (intrinsics)
- Есть хорошо оптимизированные положения и ракурсы **камер** (extrinsics)
- Есть (шумные) **карты глубины** описывающие подробно и точно геометрию

Что хотим дальше?

- Чистить карты глубины от шума на уровне каждой карты глубины (**локально**)
- Чистить карты глубины на базе их противоречий (**глобально**)
- Объединять эти карты глубины в одно общее плотное облако точек
- Строить полигональную 3D модель

Строить 3D модель из облака точек?

Где мы сейчас?

- Есть разреженное облако 3D положений **ключевых точек**
- Есть хорошо оптимизированные внутренние калибровки **камер** (intrinsics)
- Есть хорошо оптимизированные положения и ракурсы **камер** (extrinsics)
- Есть (шумные) **карты глубины** описывающие подробно и точно геометрию

Что хотим дальше?

- Чистить карты глубины от шума на уровне каждой карты глубины (**локально**)
- Чистить карты глубины на базе их противоречий (**глобально**)
- Объединять эти карты глубины в одно общее плотное облако точек
- Строить полигональную 3D модель

Строить 3D модель из облака точек?

Строить 3D модель из карт глубины?

Где мы сейчас?

- Есть разреженное облако 3D положений **ключевых точек**
- Есть хорошо оптимизированные внутренние калибровки **камер** (intrinsics)
- Есть хорошо оптимизированные положения и ракурсы **камер** (extrinsics)
- Есть (шумные) **карты глубины** описывающие подробно и точно геометрию

Что хотим дальше?

- Чистить карты глубины от шума на уровне каждой карты глубины (**локально**)
- Чистить карты глубины на базе их противоречий (**глобально**)
- Объединять эти карты глубины в одно общее плотное облако точек
- Строить полигональную 3D модель

Строить 3D модель из облака точек?

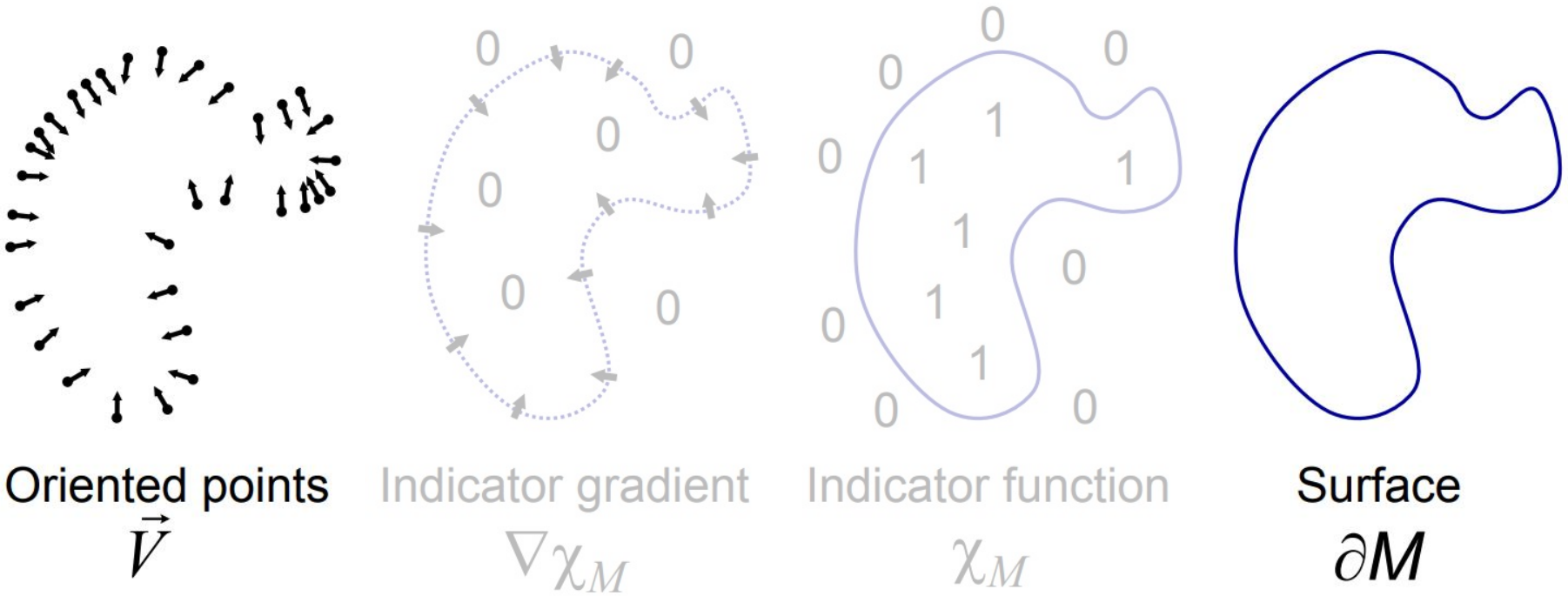
Строить 3D модель из карт глубины?

Строить 3D модель напрямую из спозиционированных фотографий?

Poisson surface reconstruction

На входе $\vec{V}$ - точки с инвертированными нормалями.

Результирующая поверхность - изоповерхность в поле индикатора.



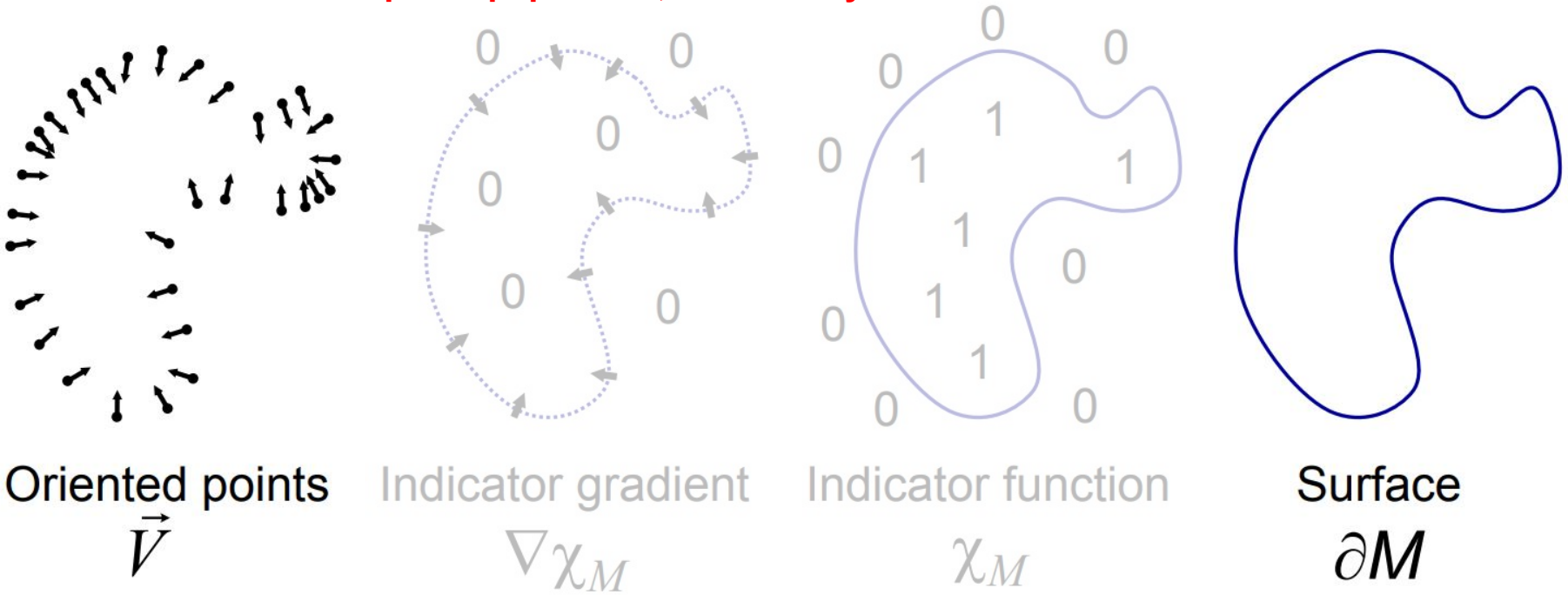
[Poisson Surface Reconstruction, Kazhdan et. al., 2006]

Poisson surface reconstruction

На входе $\vec{V}$ - точки с инвертированными нормалями.

Результирующая поверхность - изоповерхность в поле индикатора.

Есть ли какая-то потеря информации, когда мы учитываем лишь плотное облако точек?



Poisson Surface Reconstruction, Kazhdan et. al., 2006

Poisson surface reconstruction

На входе $\vec{V}$ - точки с инвертированными нормалями.

Результирующая поверхность - изоповерхность в поле индикатора.

Есть ли какая-то потеря информации, когда мы учитываем лишь плотное облако точек?

Oriented points

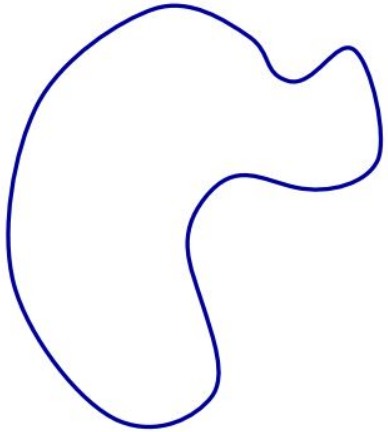
$\vec{V}$

6. Conclusion

We have shown that surface reconstruction can be expressed as a Poisson problem, which seeks the indicator function that best agrees with a set of noisy, non-uniform observations, and we have demonstrated that this approach can robustly recover fine detail from noisy real-world scans.

There are several avenues for future work:

- Extend the approach to exploit sample confidence values.
- Incorporate line-of-sight information from the scanning process into the solution process.
- Extend the system to allow out-of-core processing for huge datasets.

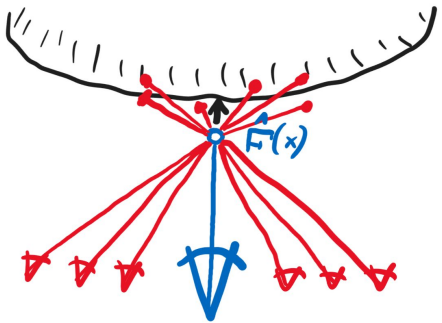


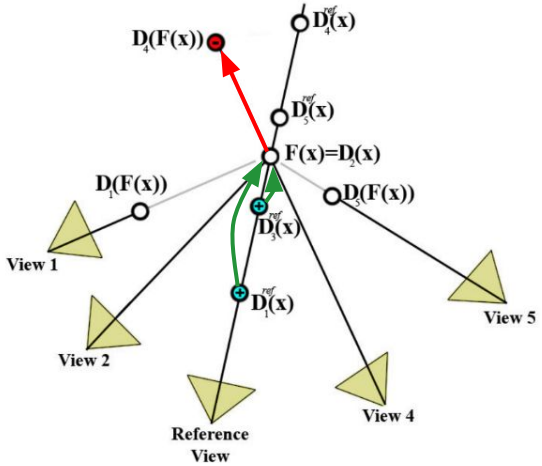
Surface

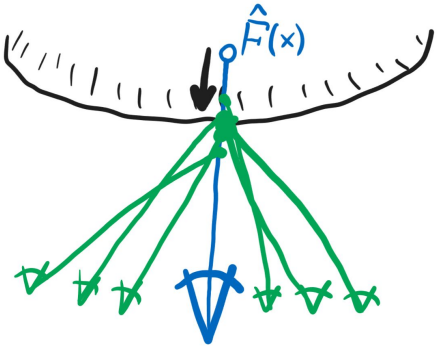
∂M

Poisson Surface Reconstruction, Kazhdan et. al., 2006

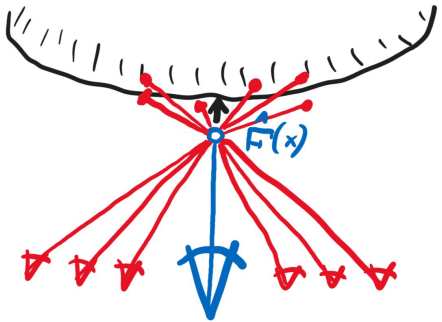
Лучи видимости (visibility rays, line-of-sight)

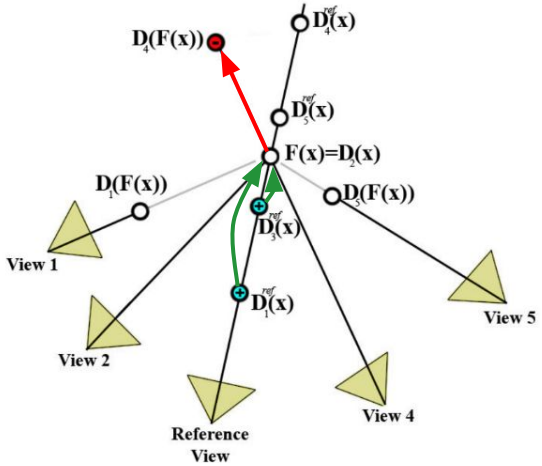


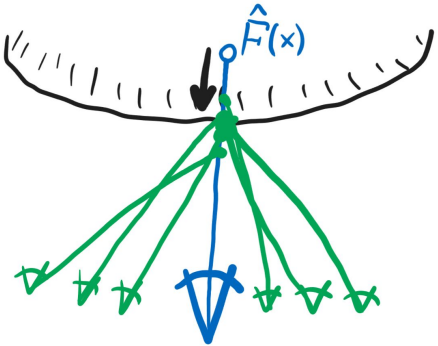




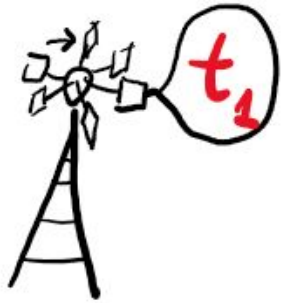
Лучи видимости (visibility rays, line-of-sight)



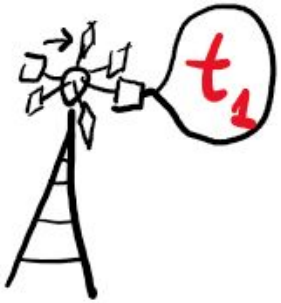




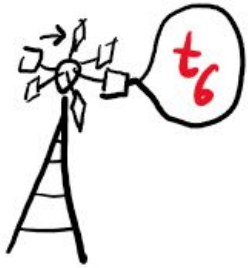
Нас они интересуют хотя бы в примитивной форме “не выброс ли я который надо удалить”?

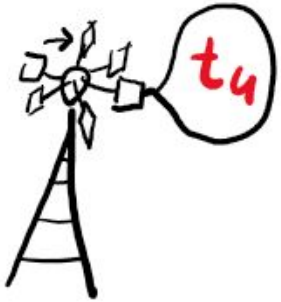


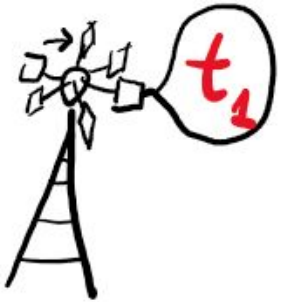
Есть метеовышки



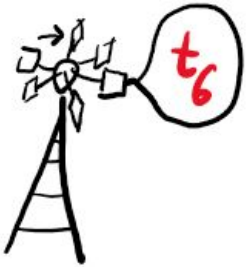
Есть метеовышки

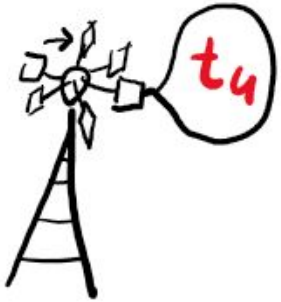






Есть метеовышки - как интерполировать температуру?



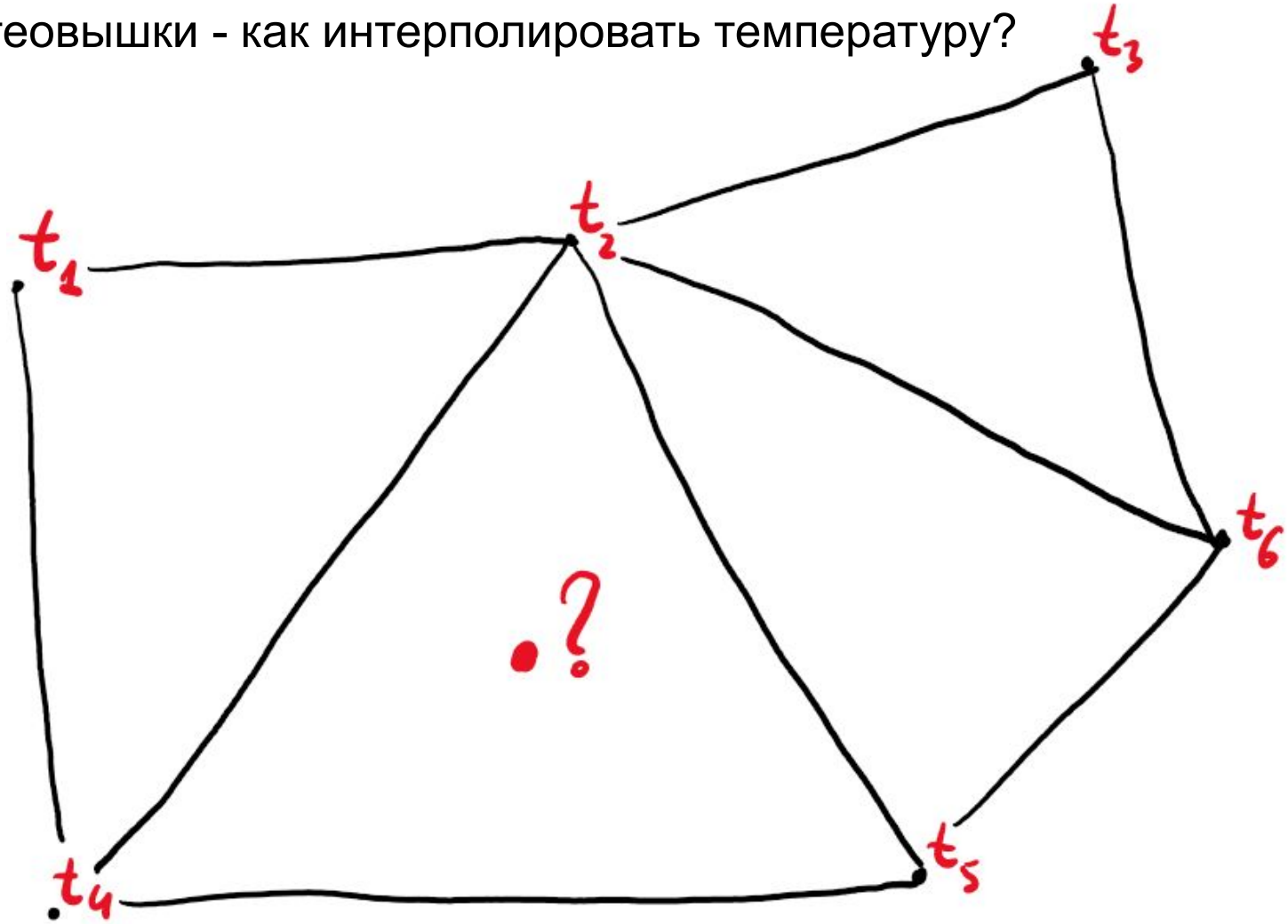


Есть метеовышки - как интерполировать температуру?

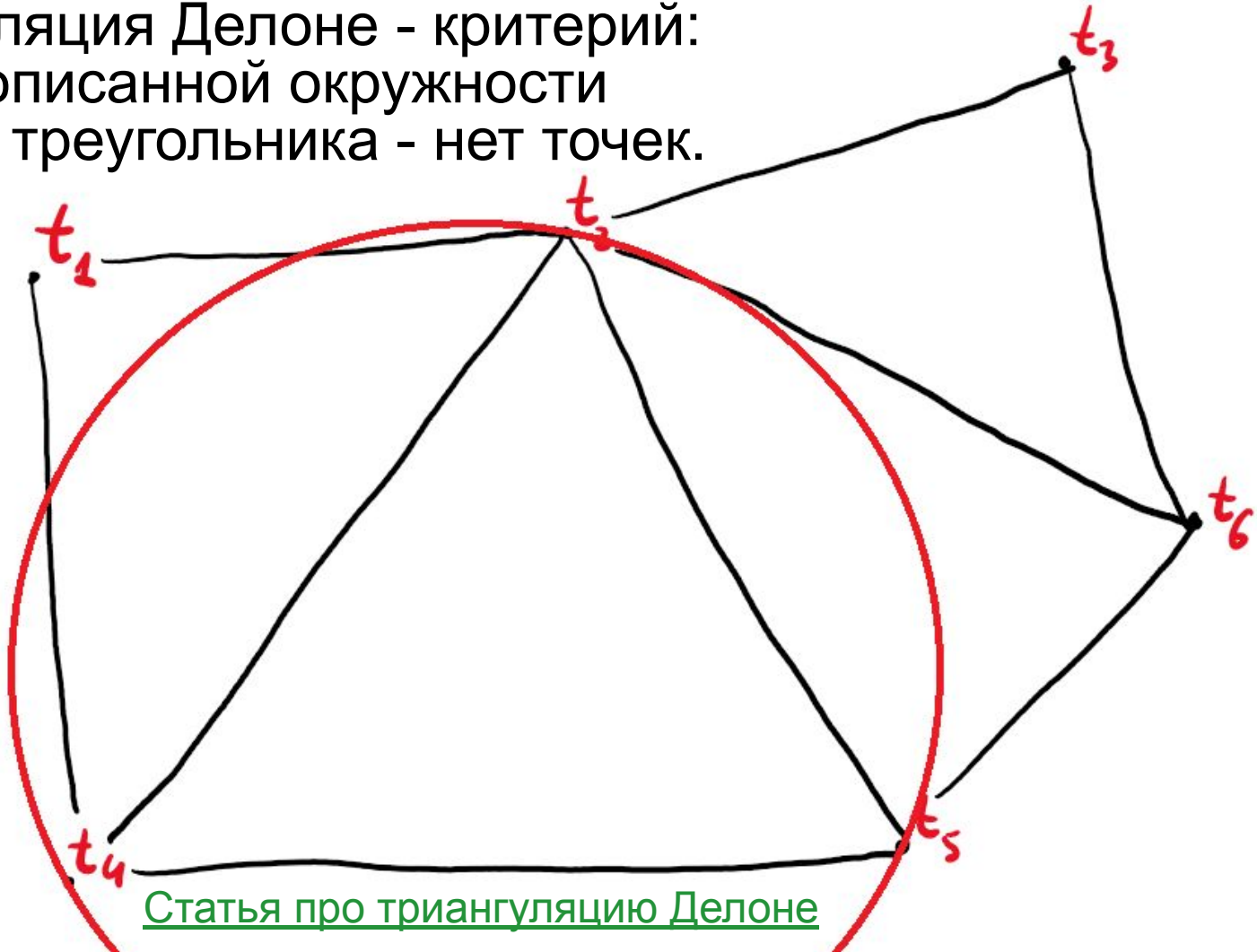
А что если есть всего три метеовышки?

Есть метеовышки - как интерполировать температуру?

Есть метеовышки - как интерполировать температуру?

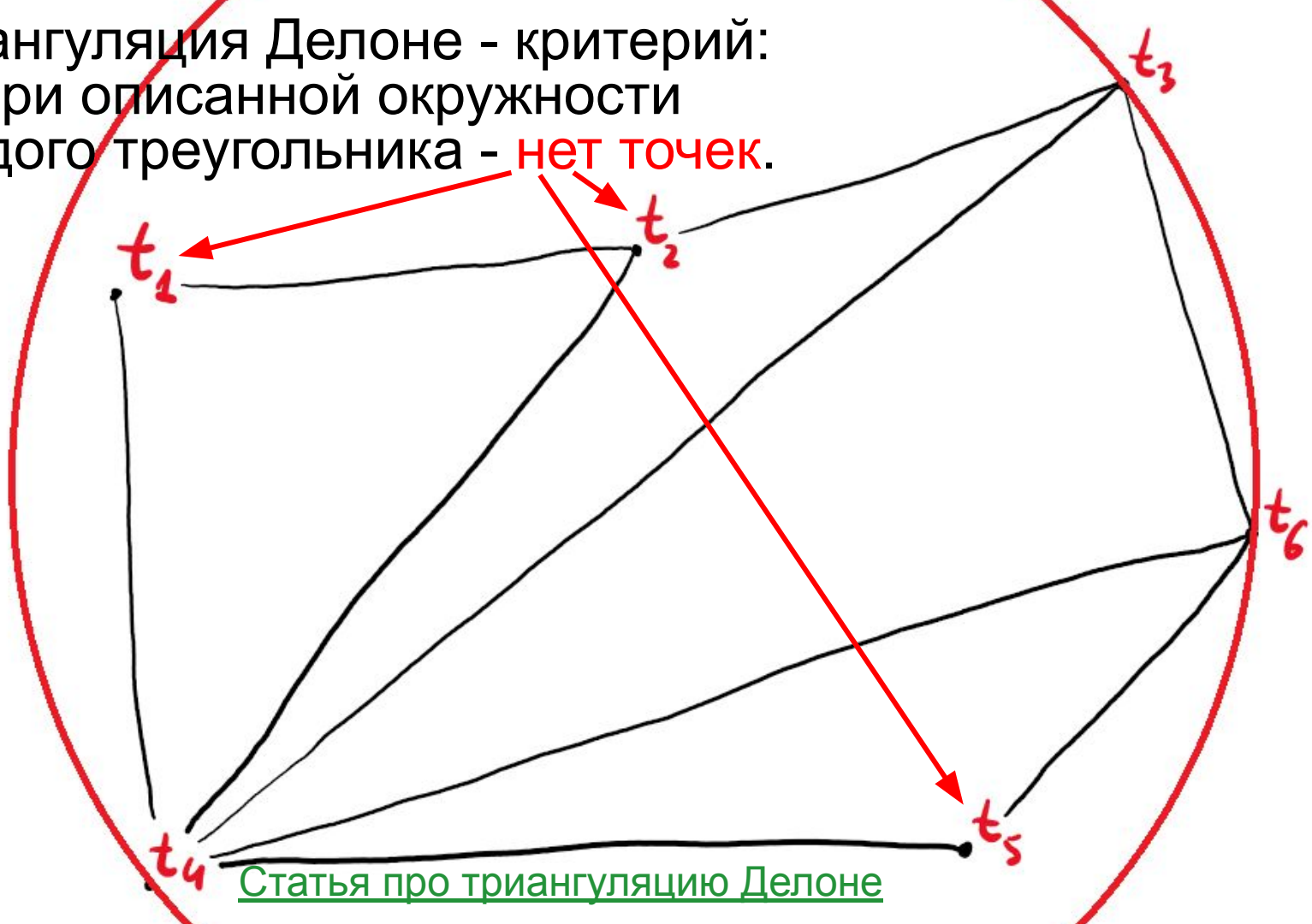


Триангуляция Делоне - критерий: внутри описанной окружности каждого треугольника - нет точек.



Статья про триангуляцию Делоне

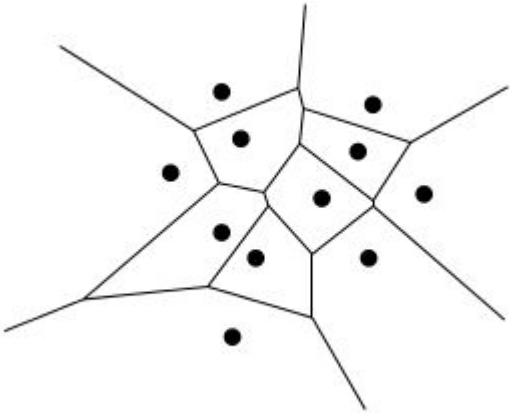
Триангуляция Делоне - критерий:
внутри описанной окружности
каждого треугольника - нет точек.

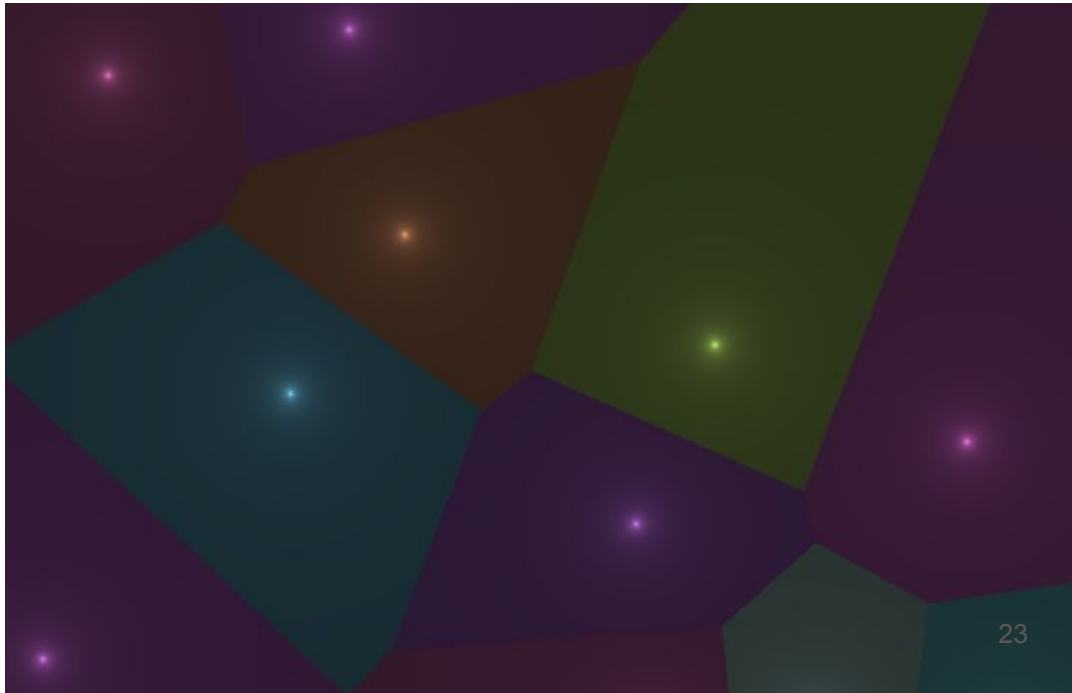


Статья про триангуляцию Делоне

Диаграмма Вороного

Есть точки p_i на плоскости. *i*-ая Ячейка диаграммы Вороного - такой кусок пространства, для любой точки которого ближайшей точкой изначального множества является p_i.

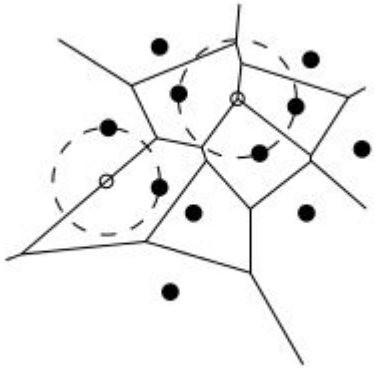


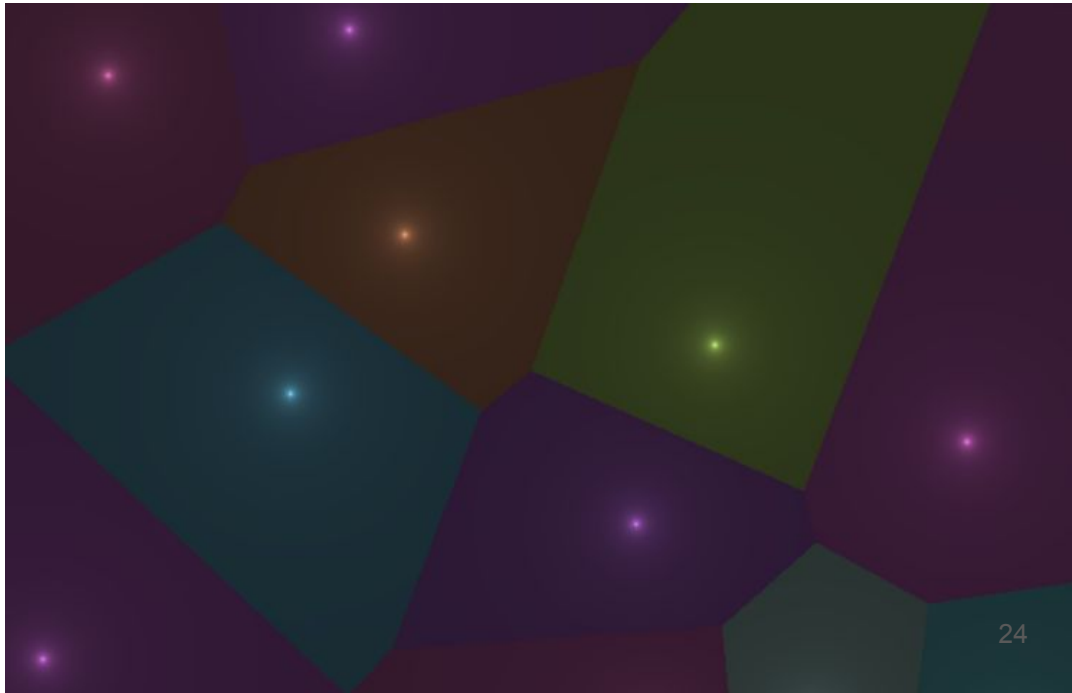


[Статья про диаграмму Вороного]

Диаграмма Вороного

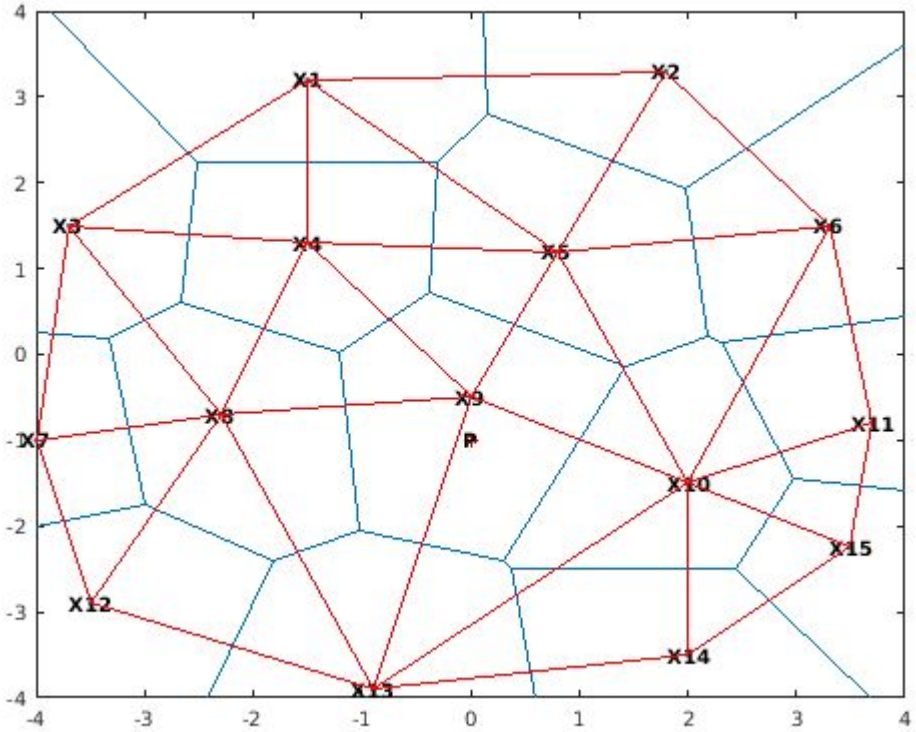
Есть точки p_i на плоскости. *i*-ая Ячейка диаграммы Вороного - такой кусок пространства, для любой точки которого ближайшей точкой изначального множества является p_i.



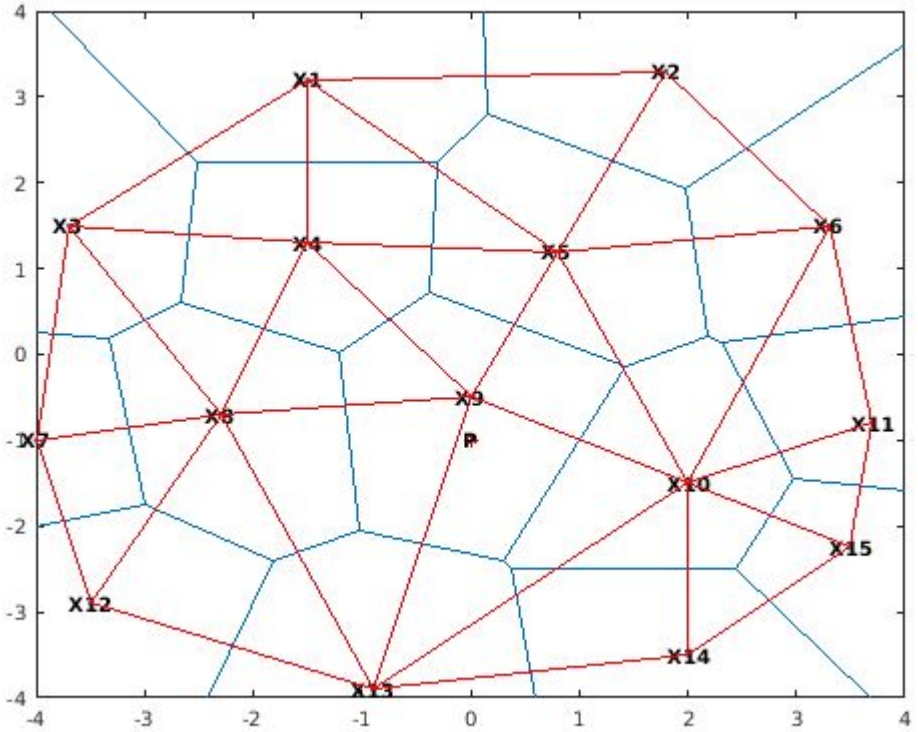


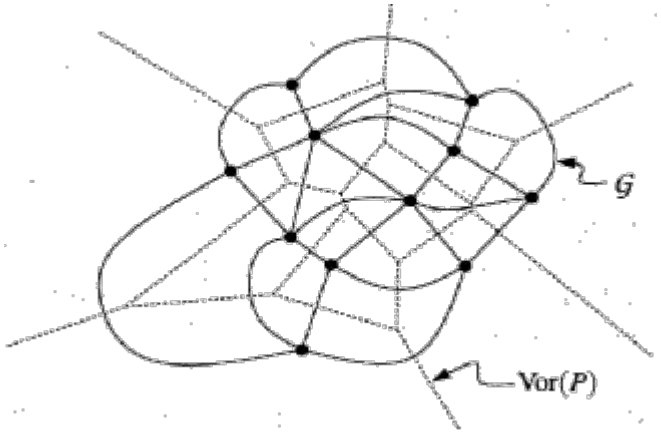
[Статья про диаграмму Вороного]

Двойственность диаграммы Вороного и триангуляции Делоне

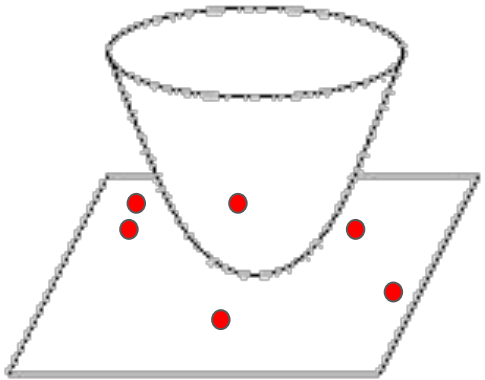


Двойственность диаграммы Вороного и триангуляции Делоне



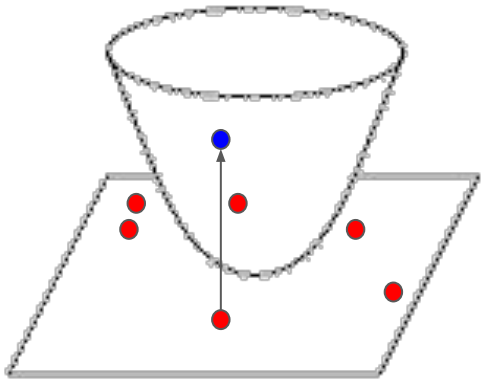


Тройственность диаграммы Вороного, триангуляции Делоне и выпуклой оболочки



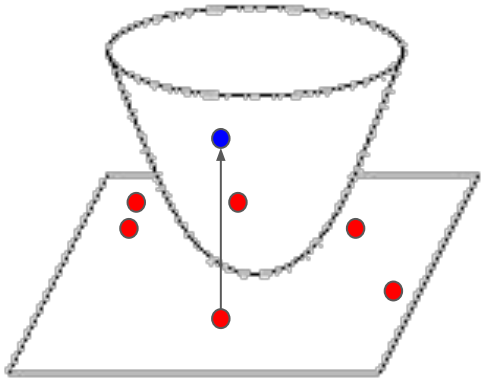
Триангуляция Делоне точек на плоскости

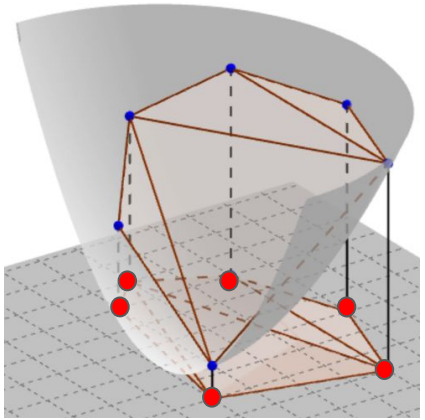
Тройственность диаграммы Вороного, триангуляции Делоне и выпуклой оболочки



Триангуляция Делоне точек на плоскости

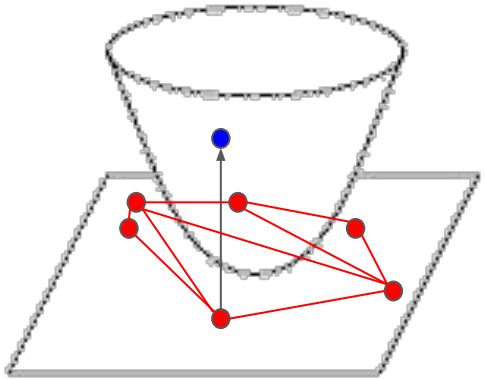
Тройственность диаграммы Вороного, триангуляции Делоне и выпуклой оболочки



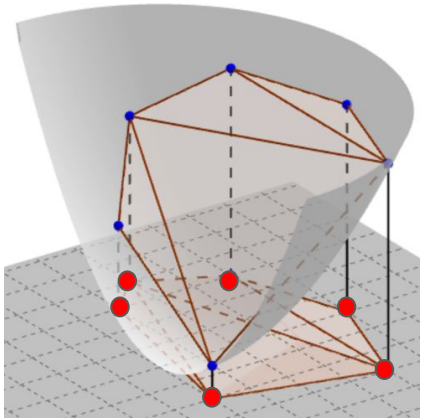


Выпуклая оболочка 3D точек на параболоиде

Тройственность диаграммы Вороного, триангуляции Делоне и выпуклой оболочки

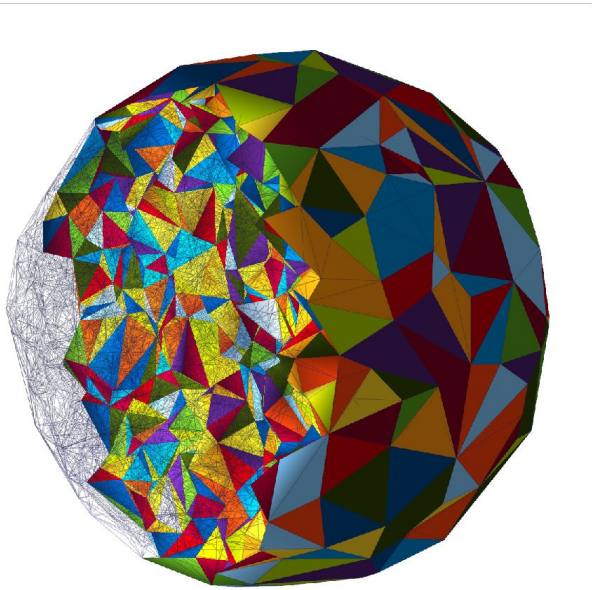


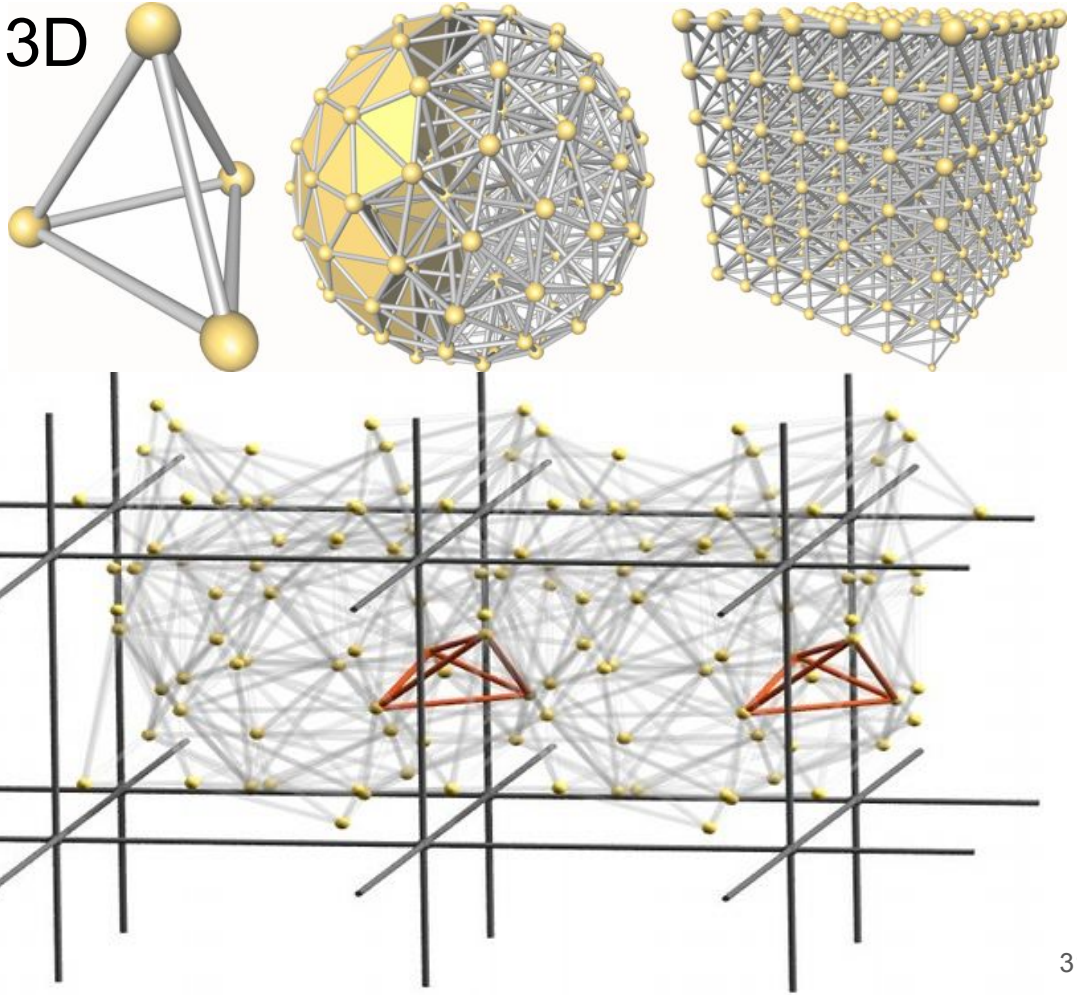
Триангуляция Делоне точек на плоскости



Выпуклая оболочка 3D точек на параболоиде

Триангуляция Делоне в 3D

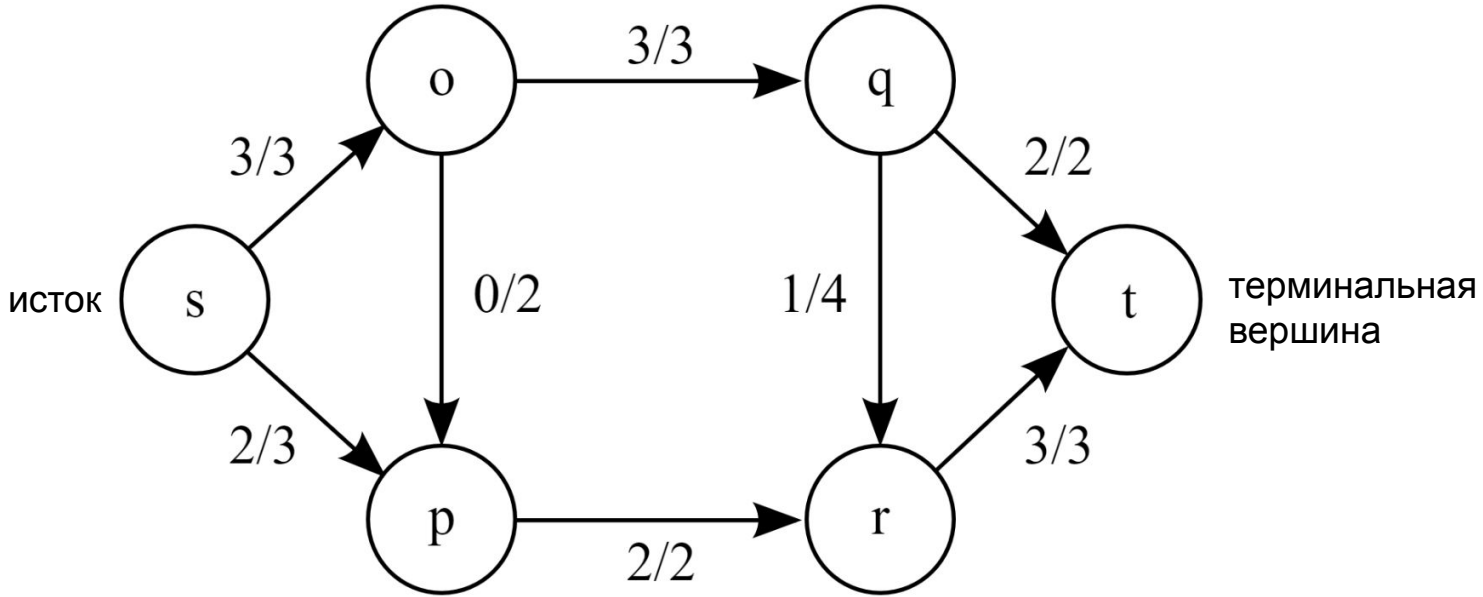




CGAL: 3D triangulation

Максимальный поток в графе (Graph **Max-Flow**)

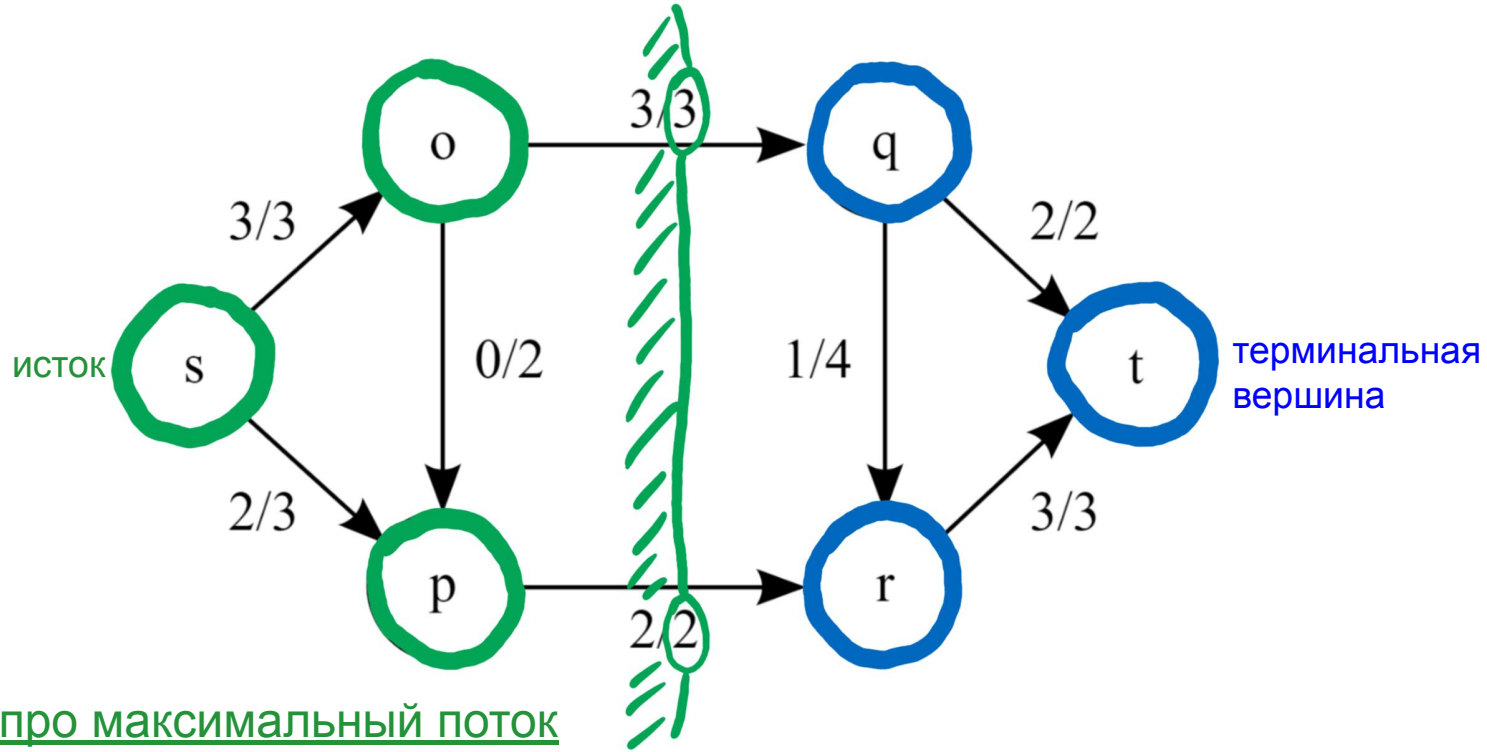
Есть ориентированный граф, на ребрах указана максимальная пропускная способность. Найти **максимальный поток** - максимальную пропускную способность сети из истока ***s*** в сток ***t*** (терминальная вершина).



Статья про максимальный поток

Минимальный разрез в графе (Graph **Min-Cut**)

Есть ориентированный граф, на ребрах указана максимальная пропускная способность. Найти **минимальный разрез** - разбиение вершин на **два множества** (мощность разреза - пропускной способности ребер между множествами).



Статья про максимальный поток

“GrabCut” — Interactive Foreground Extraction using Iterated Graph Cuts

Carsten Rother* Vladimir Kolmogorov† Andrew Blake‡
Microsoft Research Cambridge, UK

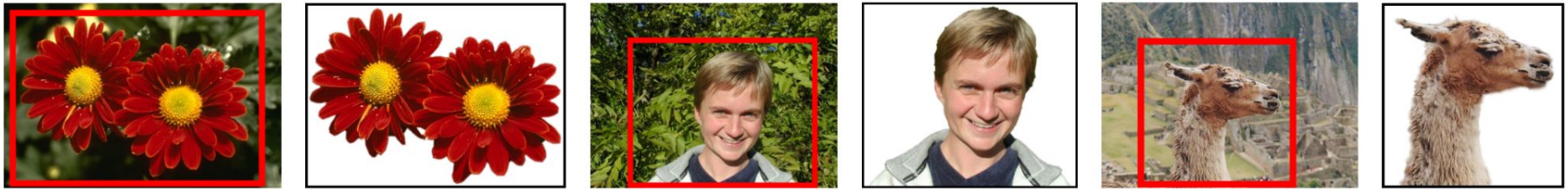


Figure 1: **Three examples of GrabCut.** The user drags a rectangle loosely around an object. The object is then extracted automatically.

Abstract

The problem of efficient, interactive foreground/background segmentation in still images is of great practical importance in image editing. Classical image segmentation tools use either texture (colour) information, e.g. Magic Wand, or edge (contrast) information, e.g. Intelligent Scissors. Recently, an approach based on optimization by graph-cut has been developed which successfully combines both types of information. In this paper we extend the

“GrabCut” — Interactive Foreground Extraction using Iterated Graph Cuts, Rother et. al., 2004

more powerful, iterative version of the optimisation. Secondly, the

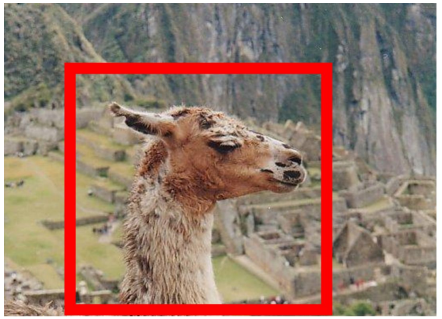
free of colour bleeding from the source background. In general, degrees of interactive effort range from editing individual pixels, at the labour-intensive extreme, to merely touching foreground and/or background in a few locations.

1.1 Previous approaches to interactive matting

In the following we describe briefly and compare several state of the art interactive tools for segmentation: Magic Wand, Intelligent

and Knockout. Fig. 2 shows their results on a matting task, together

2 Image segmentation by graph cut



First, the segmentation approach of Boykov and Jolly , the foundation on which GrabCut is built, is described in some detail.

2.1 Image segmentation

Their paper [Boykov and Jolly 2001] addresses the segmentation of a monochrome image, given an initial trimap T. The image is an array $\mathbf{z} = (z_1, \ldots, z_n, \ldots, z_N)$ of grey values, indexed by the (single) index n. The segmentation of the image is expressed as an array of "opacity" values $\underline{\alpha} = (\alpha_1, \ldots, \alpha_N)$ at each pixel. Generally $0 \leq \alpha_n \leq 1$, but for hard segmentation $\alpha_n \in \{0, 1\}$, with 0 for background and 1 for foreground. The parameters $\underline{\theta}$ describe image foreground and background grey-level distributions, and consist of histograms of grey values:

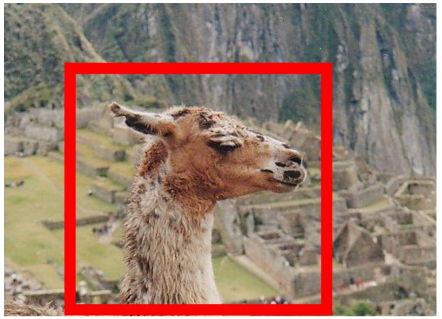
$$\underline{\theta} = \{h(z; \alpha), \alpha = 0, 1\}, \tag{1}$$

one for background and one for foreground. The histograms are assembled directly from labelled pixels from the respective trimap regions T_B, T_F. (Histograms are normalised to sum to 1 over the grey-level range: $\int_z h(z; \alpha) = 1$.)

The segmentation task is to infer the unknown opacity variables $\underline{\alpha}$ from the given image data $\mathbf{z}$ and the model $\underline{\theta}$.

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts, Rother et. al., 2004

2 Image segmentation by graph cut



First, the segmentation approach of Boykov and Jolly , the foundation on which GrabCut is built, is described in some detail.

2.1 Image segmentation

Their paper [Boykov and Jolly 2001] addresses the segmentation of a monochrome image, given an initial trimap T. The image is an array $\mathbf{z} = (z_1, \ldots, z_n, \ldots, z_N)$ of grey values, indexed by the (single) index n. The segmentation of the image is expressed as an array of "opacity" values $\underline{\alpha} = (\alpha_1, \ldots, \alpha_N)$ at each pixel. Generally $0 \leq \alpha_n \leq 1$, but for hard segmentation $\alpha_n \in \{0, 1\}$, with 0 for background and 1 for foreground. The parameters $\underline{\theta}$ describe image foreground and background grey-level distributions, and consist of histograms of grey values:

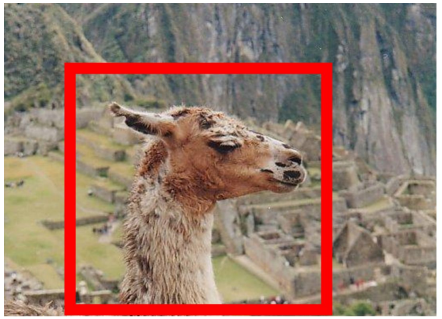
$$\underline{\theta} = \{h(z; \alpha), \alpha = 0, 1\}, \tag{1}$$

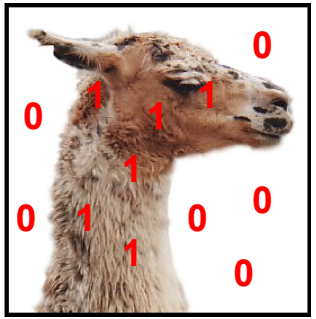
one for background and one for foreground. The histograms are assembled directly from labelled pixels from the respective trimap regions T_B, T_F. (Histograms are normalised to sum to 1 over the grey-level range: $\int_z h(z; \alpha) = 1$.)

The segmentation task is to infer the unknown opacity variables $\underline{\alpha}$ from the given image data $\mathbf{z}$ and the model $\underline{\theta}$.

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts, Rother et. al., 2004

2 Image segmentation by graph cut





First, the segmentation approach of Boykov and Jolly , the foundation on which GrabCut is built, is described in some detail.

2.1 Image segmentation

Their paper [Boykov and Jolly 2001] addresses the segmentation of a monochrome image, given an initial trimap T. The image is an array $\mathbf{z} = (z_1, \ldots, z_n, \ldots, z_N)$ of grey values, indexed by the (single) index n. The segmentation of the image is expressed as an array of "opacity" values $\underline{\alpha} = (\alpha_1, \ldots, \alpha_N)$ at each pixel. Generally $0 \leq \alpha_n \leq 1$, but for hard segmentation $\alpha_n \in \{0, 1\}$, with 0 for background and 1 for foreground. The parameters $\underline{\theta}$ describe image foreground and background grey-level distributions, and consist of histograms of grey values:

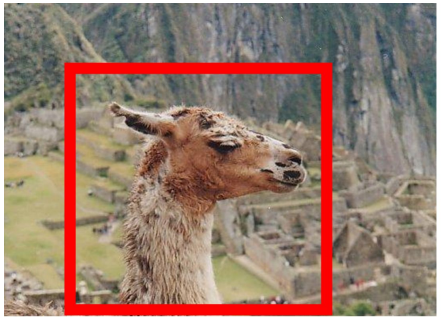
$$\underline{\theta} = \{h(z; \alpha), \alpha = 0, 1\}, \tag{1}$$

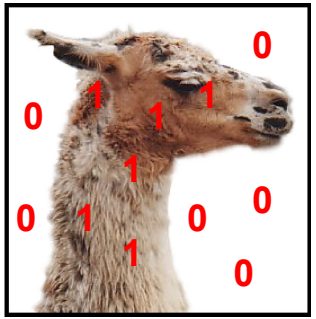
one for background and one for foreground. The histograms are assembled directly from labelled pixels from the respective trimap regions T_B, T_F. (Histograms are normalised to sum to 1 over the grey-level range: $\int_z h(z; \alpha) = 1$.)

The segmentation task is to infer the unknown opacity variables $\underline{\alpha}$ from the given image data $\mathbf{z}$ and the model $\underline{\theta}$.

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts, Rother et. al., 2004

2 Image segmentation by graph cut





First, the segmentation approach of Boykov and Jolly , the foundation on which GrabCut is built, is described in some detail.

2.1 Image segmentation

Their paper [Boykov and Jolly 2001] addresses the segmentation of a monochrome image, given an initial trimap T. The image is an array $\mathbf{z} = (z_1, \ldots, z_n, \ldots, z_N)$ of grey values, indexed by the (single) index n. The segmentation of the image is expressed as an array of "opacity" values $\underline{\alpha} = (\alpha_1, \ldots, \alpha_N)$ at each pixel. Generally $0 \leq \alpha_n \leq 1$, but for hard segmentation $\alpha_n \in \{0, 1\}$, with 0 for background and 1 for foreground. The parameters $\underline{\theta}$ describe image foreground and background grey-level distributions, and consist of histograms of grey values:

$$\underline{\theta} = \{h(z; \alpha), \alpha = 0, 1\}, \quad (1)$$

one for background and one for foreground. The histograms are assembled directly from labelled pixels from the respective trimap regions T_B, T_F. (Histograms are normalised to sum to 1 over the grey-level range: $\int_z h(z; \alpha) = 1$.)

The segmentation task is to infer the unknown opacity variables $\underline{\alpha}$ from the given image data $\mathbf{z}$ and the model $\underline{\theta}$.

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts, Rother et. al., 2004

2 Image segmentation by graph cut

First, the segmentation approach of Boykov and Jolly , the foundation on which GrabCut is built, is described in some detail.

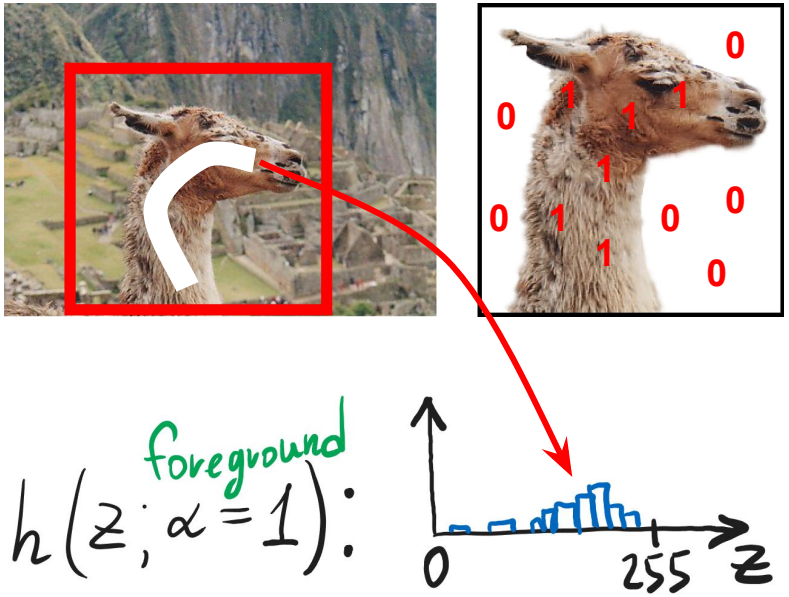
2.1 Image segmentation

Their paper [Boykov and Jolly 2001] addresses the segmentation of a monochrome image, given an initial trimap T. The image is an array $\mathbf{z} = (z_1, \ldots, z_n, \ldots, z_N)$ of grey values, indexed by the (single) index n. The segmentation of the image is expressed as an array of "opacity" values $\underline{\alpha} = (\alpha_1, \ldots, \alpha_N)$ at each pixel. Generally $0 \leq \alpha_n \leq 1$, but for hard segmentation $\alpha_n \in \{0, 1\}$, with 0 for background and 1 for foreground. The parameters $\underline{\theta}$ describe image foreground and background grey-level distributions, and consist of histograms of grey values:

$$\underline{\theta} = \{h(z; \alpha), \alpha = 0, 1\}, \quad (1)$$

one for background and one for foreground. The histograms are assembled directly from labelled pixels from the respective trimap regions T_B, T_F. (Histograms are normalised to sum to 1 over the grey-level range: $\int_z h(z; \alpha) = 1$.)

The segmentation task is to infer the unknown opacity variables $\underline{\alpha}$ from the given image data $\mathbf{z}$ and the model $\underline{\theta}$.



"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts, Rother et. al., 2004

2 Image segmentation by graph cut

First, the segmentation approach of Boykov and Jolly , the foundation on which GrabCut is built, is described in some detail.

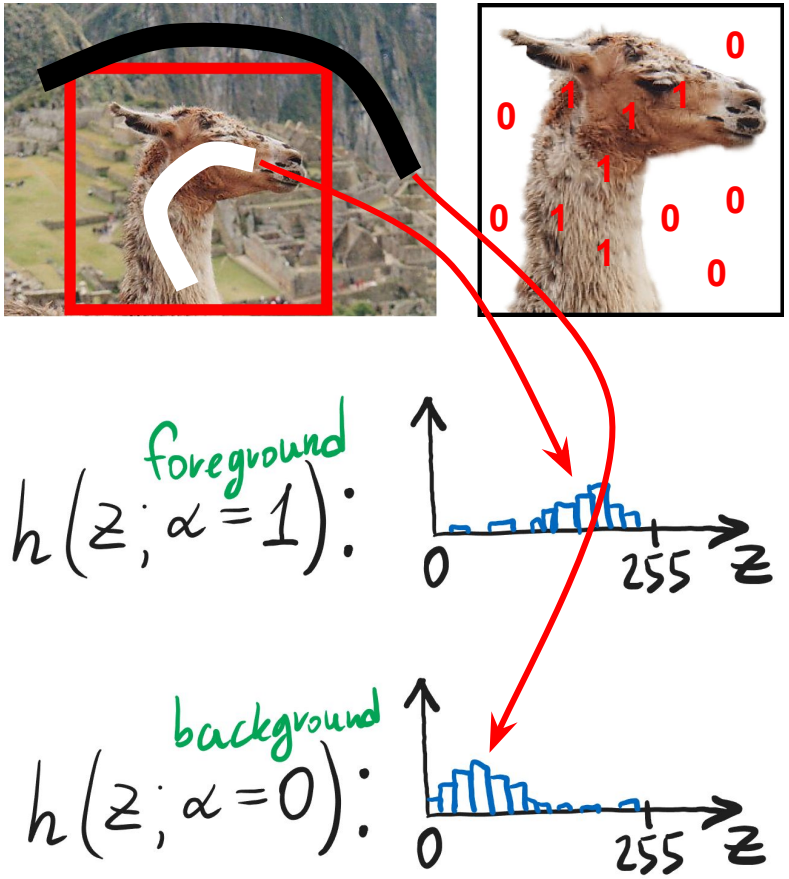
2.1 Image segmentation

Their paper [Boykov and Jolly 2001] addresses the segmentation of a monochrome image, given an initial trimap T. The image is an array $\mathbf{z} = (z_1, \ldots, z_n, \ldots, z_N)$ of grey values, indexed by the (single) index n. The segmentation of the image is expressed as an array of "opacity" values $\underline{\alpha} = (\alpha_1, \ldots, \alpha_N)$ at each pixel. Generally $0 \leq \alpha_n \leq 1$, but for hard segmentation $\alpha_n \in \{0, 1\}$, with 0 for background and 1 for foreground. The parameters $\underline{\theta}$ describe image foreground and background grey-level distributions, and consist of histograms of grey values:

$$\underline{\theta} = \{h(z; \alpha), \alpha = 0, 1\}, \quad (1)$$

one for background and one for foreground. The histograms are assembled directly from labelled pixels from the respective trimap regions T_B, T_F. (Histograms are normalised to sum to 1 over the grey-level range: $\int_z h(z; \alpha) = 1$.)

The segmentation task is to infer the unknown opacity variables $\underline{\alpha}$ from the given image data $\mathbf{z}$ and the model $\underline{\theta}$.



"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts, Rother et. al., 2004

2.2 Segmentation by energy minimisation

An energy function **E** is defined so that its minimum should correspond to a good segmentation, in the sense that it is guided both by the observed foreground and background grey-level histograms and that the opacity is "coherent", reflecting a tendency to solidity of objects. This is captured by a "Gibbs" energy of the form:

$$\mathbf{E}(\underline{\alpha}, \underline{\theta}, \mathbf{z}) = U(\underline{\alpha}, \underline{\theta}, \mathbf{z}) + V(\underline{\alpha}, \mathbf{z}) \ . \qquad (2)$$

The data term U evaluates the fit of the opacity distribution $\underline{\alpha}$ to the data $\mathbf{z}$, given the histogram model $\underline{\theta}$, and is defined to be:

$$U(\underline{\alpha}, \underline{\theta}, \mathbf{z}) = \sum_n -\log h(z_n; \alpha_n) \ . \qquad (3)$$

The smoothness term can be written as

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n) \in \mathbf{C}} dis(m,n)^{-1} \ [\alpha_n \neq \alpha_m] \exp -\beta (z_m - z_n)^2, \ (4)$$

where $[\phi]$ denotes the indicator function taking values $0, 1$ for a predicate ϕ, $\mathbf{C}$ is the set of pairs of neighboring pixels, and where $dis(\cdot)$ is the Euclidean distance of neighbouring pixels. This energy encourages coherence in regions of similar grey-level. In practice, good results are obtained by defining pixels to be neighbours if they are adjacent either horizontally/vertically or diagonally (8-way connectivity). When the constant $\beta = 0$, the smoothness term is simply

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts, Rother et. al., 2004

2.2 Segmentation by energy minimisation

An energy function **E** is defined so that its minimum should correspond to a good segmentation, in the sense that it is guided both by the observed foreground and background grey-level histograms and that the opacity is "coherent", reflecting a tendency to solidity of objects. This is captured by a "Gibbs" energy of the form:

$$\mathbf{E}(\underline{\alpha}, \underline{\theta}, \mathbf{z}) = U(\underline{\alpha}, \underline{\theta}, \mathbf{z}) + V(\underline{\alpha}, \mathbf{z}) \ . \qquad (2)$$

The data term U evaluates the fit of the opacity distribution $\underline{\alpha}$ to the data $\mathbf{z}$, given the histogram model $\underline{\theta}$, and is defined to be:

$$U(\underline{\alpha}, \underline{\theta}, \mathbf{z}) = \sum_n -\log h(z_n; \alpha_n) \ . \qquad (3)$$

The smoothness term can be written as

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n) \in \mathbf{C}} dis(m,n)^{-1} \ [\alpha_n \neq \alpha_m] \exp -\beta (z_m - z_n)^2, \ (4)$$

where $[\phi]$ denotes the indicator function taking values $0, 1$ for a predicate ϕ, $\mathbf{C}$ is the set of pairs of neighboring pixels, and where $dis(\cdot)$ is the Euclidean distance of neighbouring pixels. This energy encourages coherence in regions of similar grey-level. In practice, good results are obtained by defining pixels to be neighbours if they are adjacent either horizontally/vertically or diagonally (8-way connectivity). When the constant $\beta = 0$, the smoothness term is simply

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts, Rother et. al., 2004

2.2 Segmentation by energy minimisation

An energy function **E** is defined so that its minimum should correspond to a good segmentation, in the sense that it is guided both by the observed foreground and background grey-level histograms and that the opacity is "coherent", reflecting a tendency to solidity of objects. This is captured by a "Gibbs" energy of the form:

$$\mathbf{E}(\underline{\alpha}, \underline{\theta}, \mathbf{z}) = U(\underline{\alpha}, \underline{\theta}, \mathbf{z}) + V(\underline{\alpha}, \mathbf{z}) \ . \qquad (2)$$

The data term U evaluates the fit of the opacity distribution $\underline{\alpha}$ to the data $\mathbf{z}$, given the histogram model $\underline{\theta}$, and is defined to be:

$$U(\underline{\alpha}, \underline{\theta}, \mathbf{z}) = \sum_n -\log h(z_n; \alpha_n) \ . \qquad (3)$$

The smoothness term can be written as

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n) \in \mathbf{C}} dis(m,n)^{-1} \ [\alpha_n \neq \alpha_m] \exp -\beta (z_m - z_n)^2, \ (4)$$

where $[\phi]$ denotes the indicator function taking values $0, 1$ for a predicate ϕ, $\mathbf{C}$ is the set of pairs of neighboring pixels, and where $dis(\cdot)$ is the Euclidean distance of neighbouring pixels. This energy encourages coherence in regions of similar grey-level. In practice, good results are obtained by defining pixels to be neighbours if they are adjacent either horizontally/vertically or diagonally (8-way connectivity). When the constant $\beta = 0$, the smoothness term is simply

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts, Rother et. al., 2004

2.2 Segmentation by energy minimisation

An energy function **E** is defined so that its minimum should correspond to a good segmentation, in the sense that it is guided both by the observed foreground and background grey-level histograms and that the opacity is "coherent", reflecting a tendency to solidity of objects. This is captured by a "Gibbs" energy of the form:

$$\mathbf{E}(\underline{\alpha},\underline{\theta},\mathbf{z}) = U(\underline{\alpha},\underline{\theta},\mathbf{z}) + V(\underline{\alpha},\mathbf{z})\ . \qquad (2)$$

The data term U evaluates the fit of the opacity distribution $\underline{\alpha}$ to the data $\mathbf{z}$, given the histogram model $\underline{\theta}$, and is defined to be:

$$U(\underline{\alpha},\underline{\theta},\mathbf{z}) = \sum_n -\log h(z_n;\alpha_n)\,. \qquad (3)$$

The smoothness term can be written as

$$V(\underline{\alpha},\mathbf{z}) = \gamma \sum_{(m,n)\in\mathbf{C}} dis(m,n)^{-1}\,[\alpha_n \neq \alpha_m] \exp -\beta(z_m - z_n)^2, \qquad (4)$$

where $[\phi]$ denotes the indicator function taking values $0,1$ for a predicate ϕ, $\mathbf{C}$ is the set of pairs of neighboring pixels, and where $dis(\cdot)$ is the Euclidean distance of neighbouring pixels. This energy encourages coherence in regions of similar grey-level. In practice, good results are obtained by defining pixels to be neighbours if they are adjacent either horizontally/vertically or diagonally (8-way connectivity). When the constant $\beta = 0$, the smoothness term is simply

Now that the energy model is fully defined, the segmentation can be estimated as a global minimum:

$$\hat{\underline{\alpha}} = \arg\min_{\underline{\alpha}} \mathbf{E}(\underline{\alpha},\underline{\theta}). \qquad (6)$$

Minimisation is done using a standard minimum cut algorithm [Boykov and Jolly 2001; Kolmogorov and Zabih 2002]. This algorithm forms the foundation for hard segmentation, and the next section outlines three developments which have led to the new hard segmentation algorithm within GrabCut. First, the monochrome image model is replaced for colour by a Gaussian Mixture Model (GMM) in place of histograms. Secondly, the one-shot minimum cut estimation algorithm is replaced by a more powerful, iterative procedure that alternates between estimation and parameter learning. Thirdly, the demands on the interactive user are relaxed by allowing incomplete labelling — the user specifies only T_B for the

"GrabCut" — Interactive Foreground Extraction using Iterated Graph Cuts, Rother et. al., 2004

Interactive Graph Cuts
for Optimal Boundary & Region Segmentation of Objects in N-D Images

Yuri Y. Boykov Marie-Pierre Jolly

Imaging and Visualization Department
Siemens Corporate Research
755 College Road East, Princeton, NJ 08540, USA
yuri@scr.siemens.com

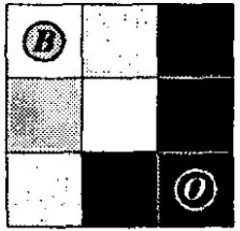
Abstract

In this paper we describe a new technique for general purpose interactive segmentation of N-dimensional images. The user marks certain pixels as "object" or "background" to provide hard constraints for segmentation. Additional soft constraints incorporate both boundary and region information. Graph cuts are used to find the globally optimal segmentation of the N-dimensional image. The obtained solution gives the best balance of boundary and region properties among all segmentations satisfying the constraints. The topology of our segmentation is unrestricted and both "object" and "background" segments may consist of several isolated parts. Some experimental results are presented

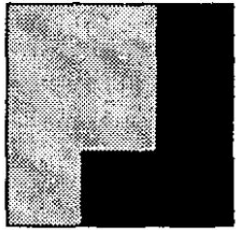
mentation. A globally optimal segmentation can be very efficiently recomputed when the user adds or removes any hard constraints (seeds). This allows the user to get any desired segmentation results quickly via very intuitive interactions. Our method applies to N-D images (volumes).

One of the main advantages of our interactive segmentation method is that it provides a globally optimal solution for an N-dimensional segmentation when the cost function is clearly defined. Some earlier techniques (snakes [14, 4], deformable templates [21], shortest path [15], ratio regions [5], and other [13]) can do that only in 2D applications when a segmentation boundary is a 1D curve. Many techniques either don't have a clear cost function at all (region growing, split and merge, see Chapter 10 in [11]) or compute only an

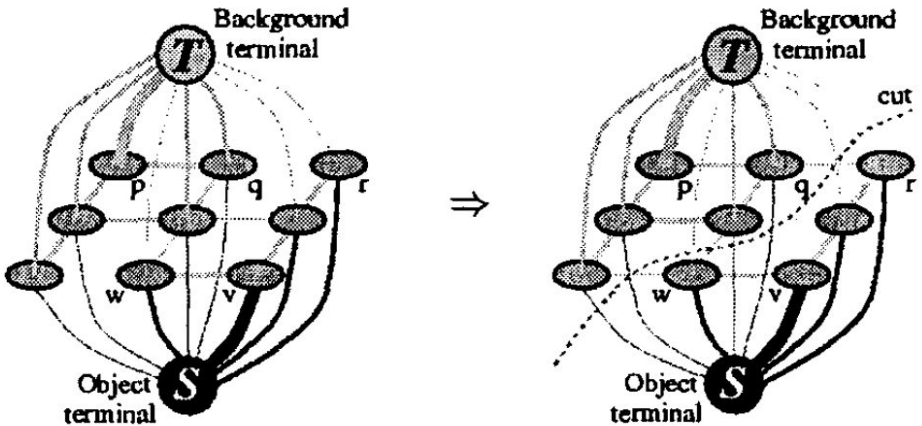
Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Boykov et. al., 2001



(a) Image with seeds.



(d) Segmentation results.

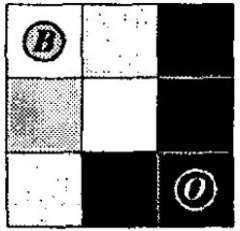


(b) Graph.

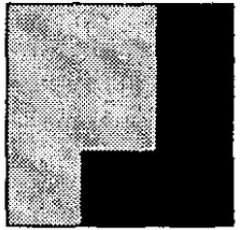
(c) Cut.

Figure 1. A simple 2D segmentation example for a 3×3 image. The seeds are $\mathcal{O} = \{v\}$ and $\mathcal{B} = \{p\}$. The cost of each edge is reflected by the edge's thickness. The regional term (2) and hard constraints (4,5) define the costs of t-links. The boundary term (3) defines the costs of n-links. Inexpensive edges are attractive choices for the minimum cost cut.

Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Boykov et. al., 2001

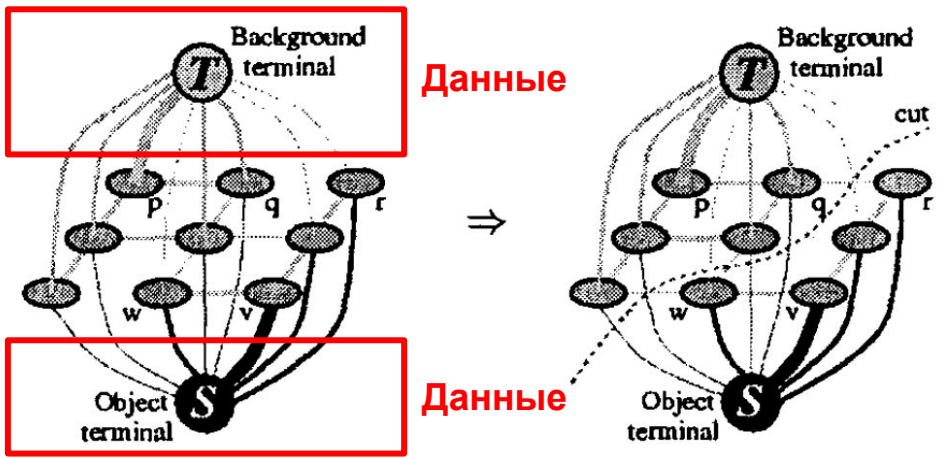


(a) Image with seeds.



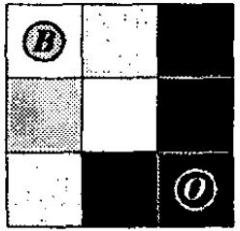
(d) Segmentation results.

Figure 1. A simple 2D segmentation example for a 3×3 image. The seeds are $\mathcal{O} = \{v\}$ and $\mathcal{B} = \{p\}$. The cost of each edge is reflected by the edge's thickness. The regional term (2) and hard constraints (4,5) define the costs of t-links. The boundary term (3) defines the costs of n-links. Inexpensive edges are attractive choices for the minimum cost cut.

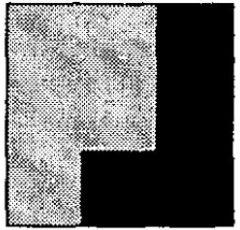


(b) Graph. (c) Cut.

Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Boykov et. al., 2001

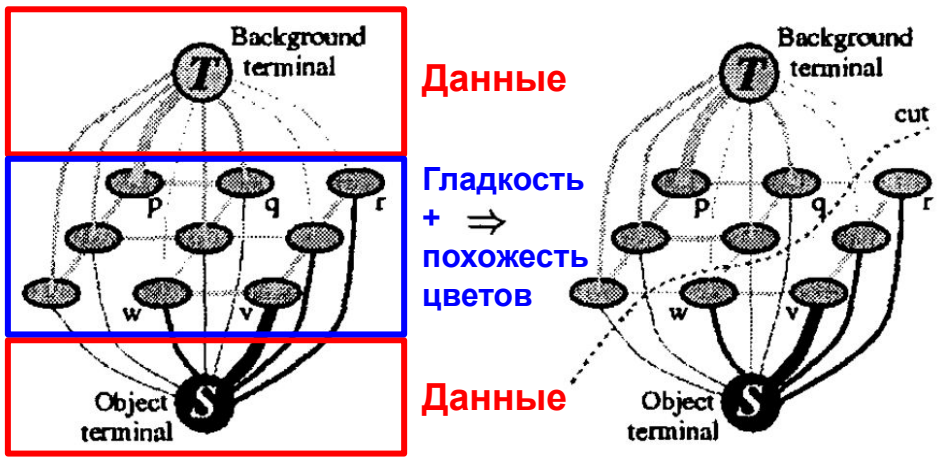


(a) Image with seeds.



(d) Segmentation results.

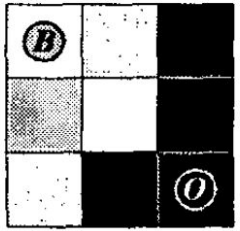
Figure 1. A simple 2D segmentation example for a 3×3 image. The seeds are $\mathcal{O} = \{v\}$ and $\mathcal{B} = \{p\}$. The cost of each edge is reflected by the edge's thickness. The regional term (2) and hard constraints (4,5) define the costs of t-links. The boundary term (3) defines the costs of n-links. Inexpensive edges are attractive choices for the minimum cost cut.



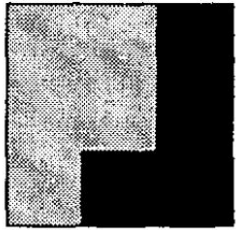
(b) Graph.

(c) Cut.

Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Boykov et. al., 2001

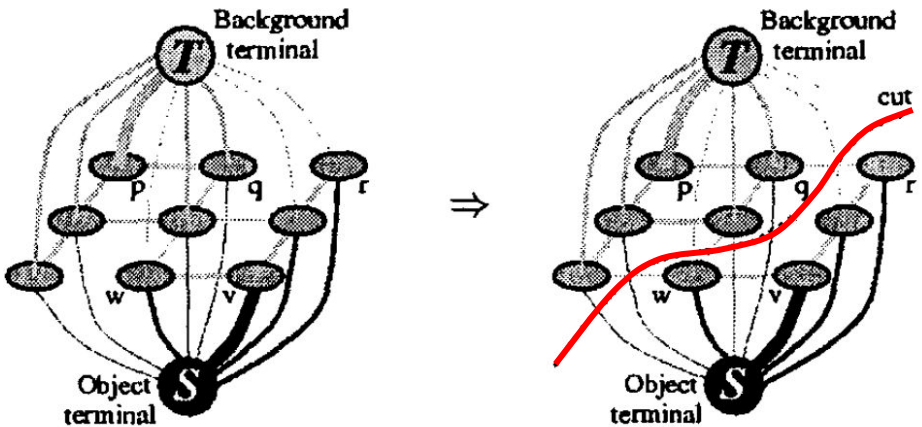


(a) Image with seeds.



(d) Segmentation results.

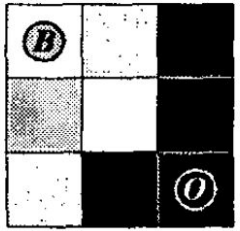
Figure 1. A simple 2D segmentation example for a 3×3 image. The seeds are $\mathcal{O} = \{v\}$ and $\mathcal{B} = \{p\}$. The cost of each edge is reflected by the edge's thickness. The regional term (2) and hard constraints (4,5) define the costs of t-links. The boundary term (3) defines the costs of n-links. Inexpensive edges are attractive choices for the minimum cost cut.



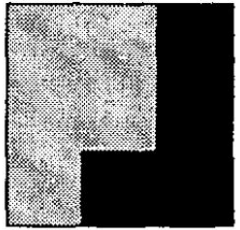
(b) Graph. (c) Cut.

Из чего состоит сумма такого минимального разреза?

Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Boykov et. al., 2001

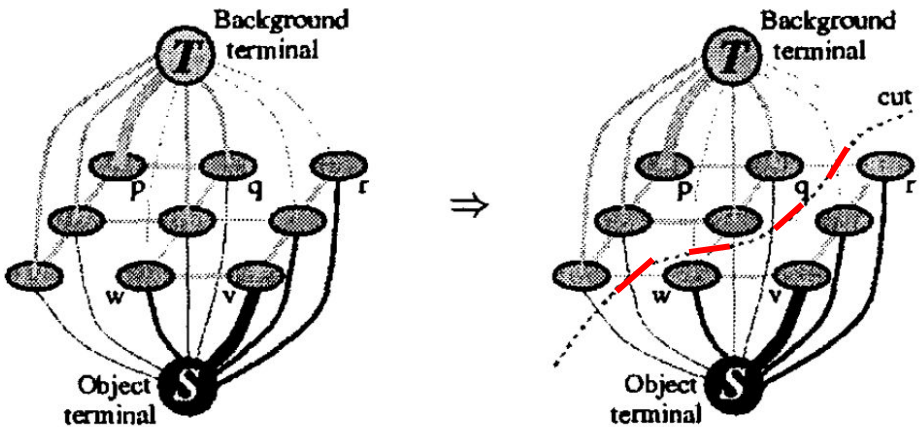


(a) Image with seeds.



(d) Segmentation results.

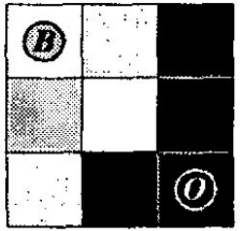
Figure 1. A simple 2D segmentation example for a 3×3 image. The seeds are $\mathcal{O} = \{v\}$ and $\mathcal{B} = \{p\}$. The cost of each edge is reflected by the edge's thickness. The regional term (2) and hard constraints (4,5) define the costs of t-links. The boundary term (3) defines the costs of n-links. Inexpensive edges are attractive choices for the minimum cost cut.



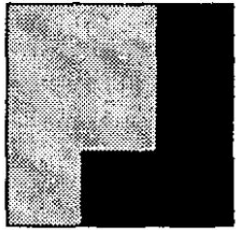
(b) Graph. (c) Cut.

Из чего состоит сумма такого минимального разреза?

Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Boykov et. al., 2001

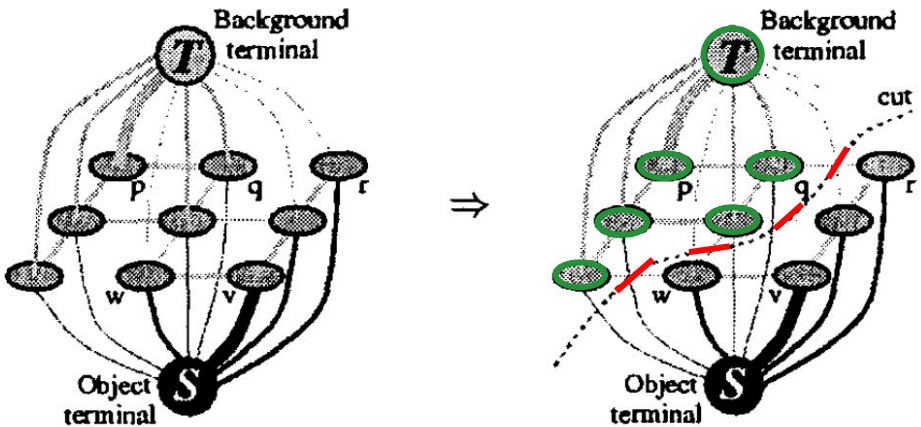


(a) Image with seeds.



(d) Segmentation results.

Figure 1. A simple 2D segmentation example for a 3×3 image. The seeds are $\mathcal{O} = \{v\}$ and $\mathcal{B} = \{p\}$. The cost of each edge is reflected by the edge's thickness. The regional term (2) and hard constraints (4,5) define the costs of t-links. The boundary term (3) defines the costs of n-links. Inexpensive edges are attractive choices for the minimum cost cut.

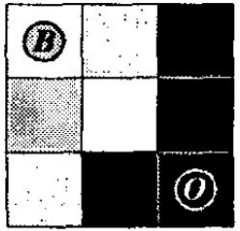


(b) Graph. (c) Cut.

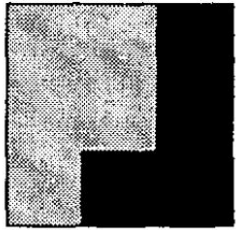
Из чего состоит сумма такого минимального разреза?

Минимальный разрез - разбиение вершин на **два множества** (мощность разреза - пропускной способности ребер между множествами).

Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Boykov et. al., 2001

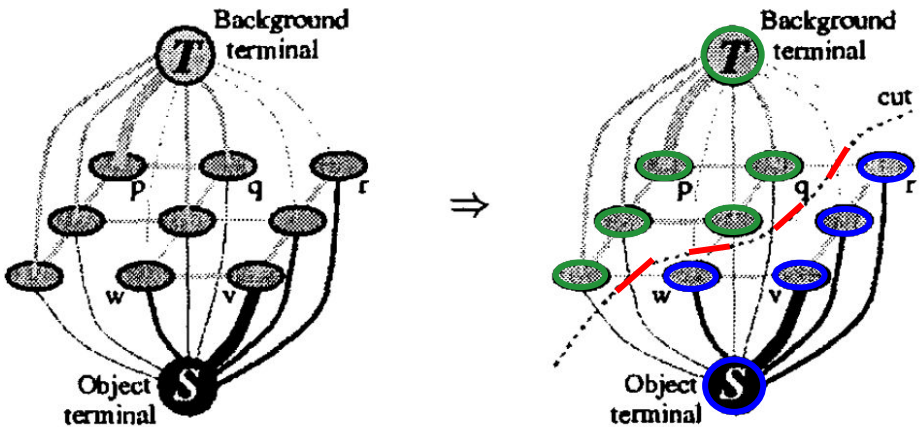


(a) Image with seeds.



(d) Segmentation results.

Figure 1. A simple 2D segmentation example for a 3×3 image. The seeds are $\mathcal{O} = \{v\}$ and $\mathcal{B} = \{p\}$. The cost of each edge is reflected by the edge's thickness. The regional term (2) and hard constraints (4,5) define the costs of t-links. The boundary term (3) defines the costs of n-links. Inexpensive edges are attractive choices for the minimum cost cut.

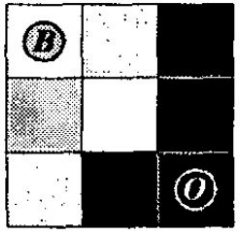


(b) Graph. (c) Cut.

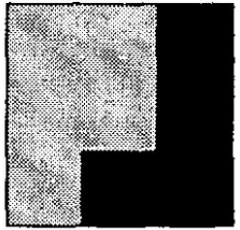
Из чего состоит сумма такого минимального разреза?

Минимальный разрез - разбиение вершин на **два множества** (мощность разреза - пропускной способности ребер между множествами).

Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Boykov et. al., 2001

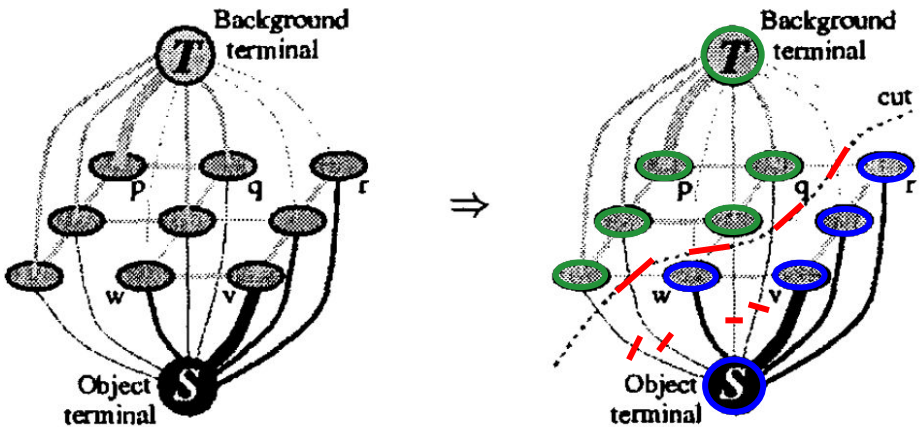


(a) Image with seeds.



(d) Segmentation results.

Figure 1. A simple 2D segmentation example for a 3×3 image. The seeds are $\mathcal{O} = \{v\}$ and $\mathcal{B} = \{p\}$. The cost of each edge is reflected by the edge's thickness. The regional term (2) and hard constraints (4,5) define the costs of t-links. The boundary term (3) defines the costs of n-links. Inexpensive edges are attractive choices for the minimum cost cut.

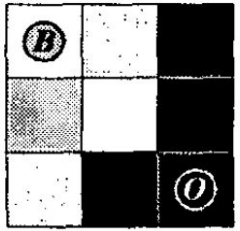


(b) Graph. (c) Cut.

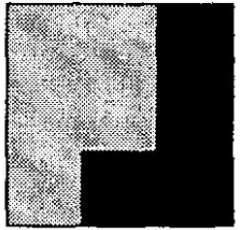
Из чего состоит сумма такого минимального разреза?

Минимальный разрез - разбиение вершин на **два множества** (мощность разреза - пропускной способности ребер между множествами).

Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Boykov et. al., 2001

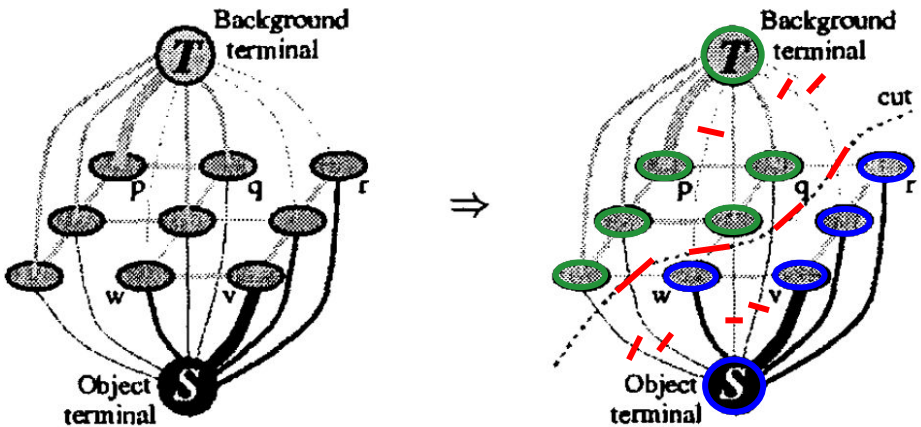


(a) Image with seeds.



(d) Segmentation results.

Figure 1. A simple 2D segmentation example for a 3×3 image. The seeds are $\mathcal{O} = \{v\}$ and $\mathcal{B} = \{p\}$. The cost of each edge is reflected by the edge's thickness. The regional term (2) and hard constraints (4,5) define the costs of t-links. The boundary term (3) defines the costs of n-links. Inexpensive edges are attractive choices for the minimum cost cut.

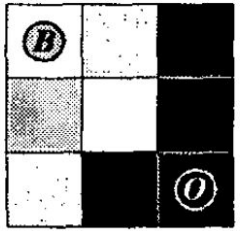


(b) Graph. (c) Cut.

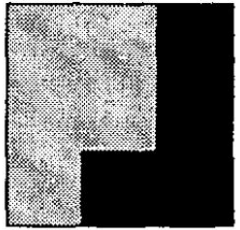
Из чего состоит сумма такого минимального разреза?

Минимальный разрез - разбиение вершин на **два множества** (мощность разреза - пропускной способности ребер между множествами).

Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Boykov et. al., 2001

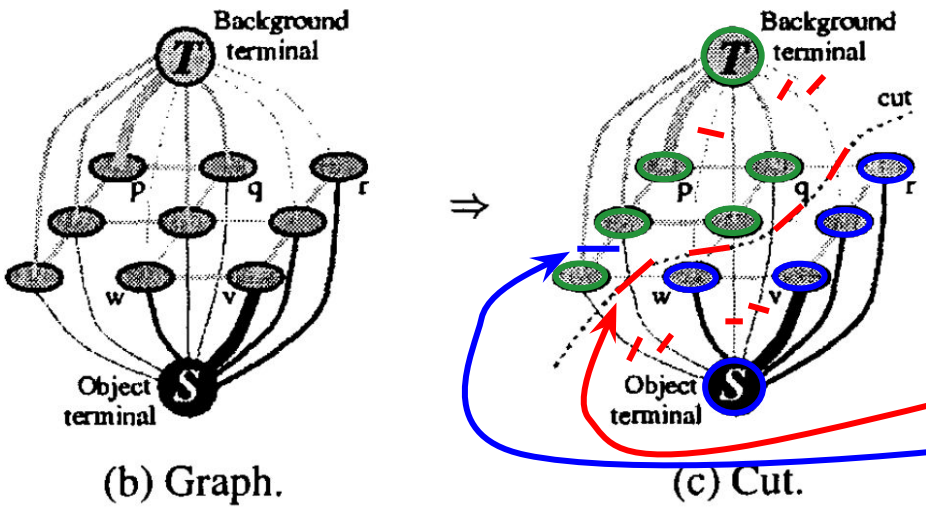


(a) Image with seeds.



(d) Segmentation results.

Figure 1. A simple 2D segmentation example for a 3×3 image. The seeds are $\mathcal{O} = \{v\}$ and $\mathcal{B} = \{p\}$. The cost of each edge is reflected by the edge's thickness. The regional term (2) and hard constraints (4,5) define the costs of t-links. The boundary term (3) defines the costs of n-links. Inexpensive edges are attractive choices for the minimum cost cut.

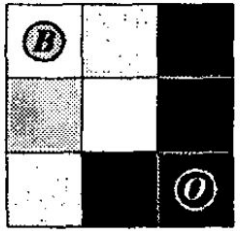


(b) Graph.

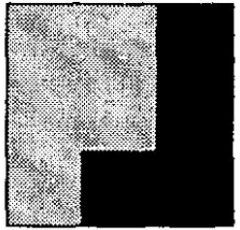
(c) Cut.

Что повлияло на проведение разреза здесь?
Почему победило это ребро а не соседнее?

Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Boykov et. al., 2001

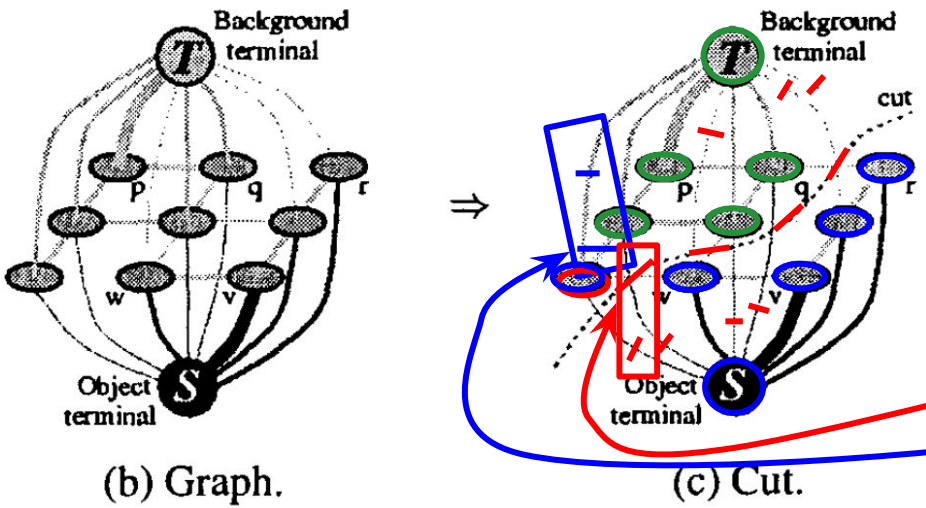


(a) Image with seeds.



(d) Segmentation results.

Figure 1. A simple 2D segmentation example for a 3×3 image. The seeds are $\mathcal{O} = \{v\}$ and $\mathcal{B} = \{p\}$. The cost of each edge is reflected by the edge's thickness. The regional term (2) and hard constraints (4,5) define the costs of t-links. The boundary term (3) defines the costs of n-links. Inexpensive edges are attractive choices for the minimum cost cut.



(b) Graph.

(c) Cut.

Что повлияло на проведение разреза здесь?
Выбор между этими и этими двумя слагаемыми.

Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Boykov et. al., 2001

Ссылки (Graph Max-Flow/Min-Cut: алгоритмы)

“GrabCut” — Interactive Foreground Extraction using Iterated Graph Cuts, Rother et. al., 2004

Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Boykov et. al., 2001

Boykov/Kolmogorov:

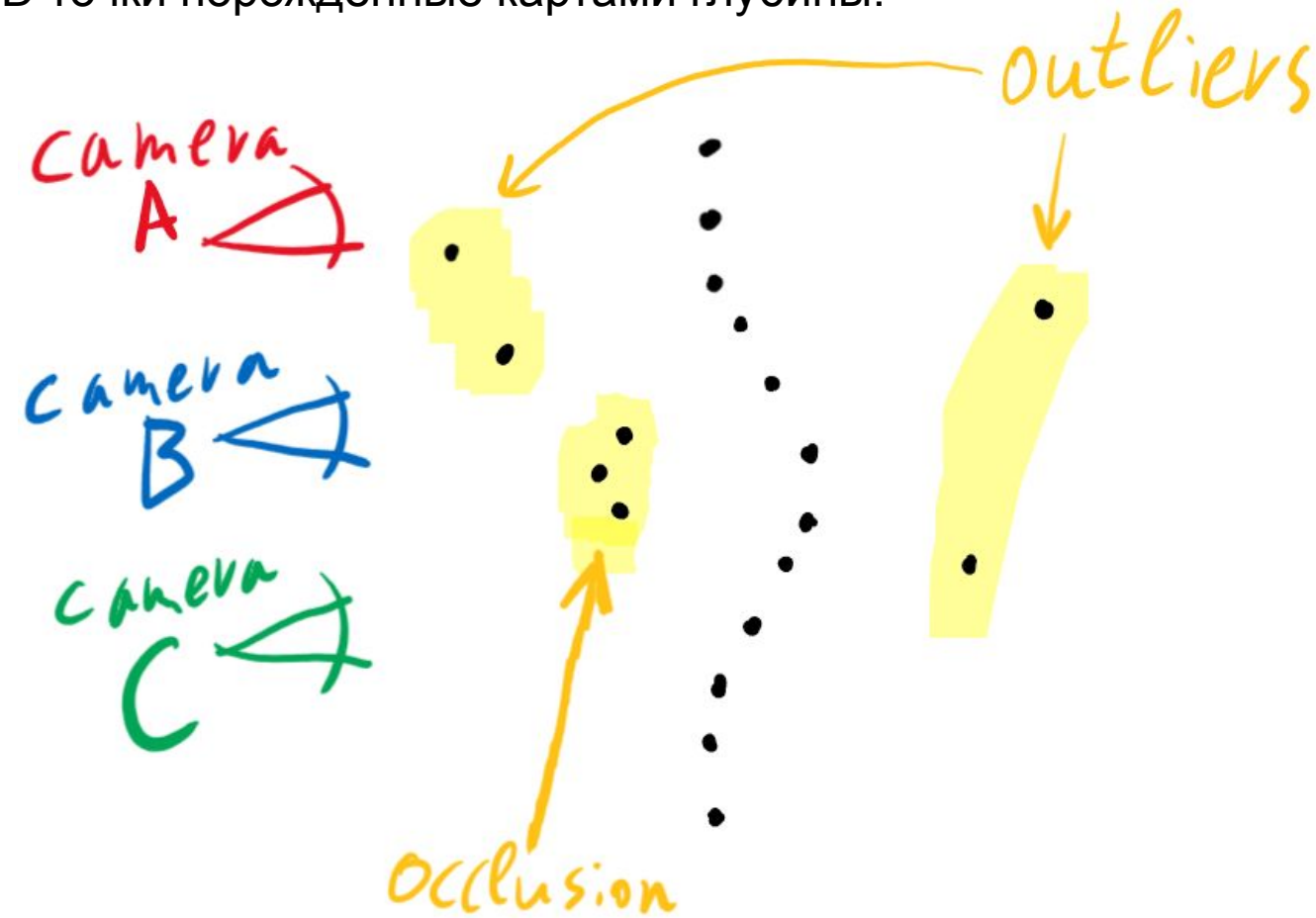
- An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision, Boykov and Kolmogorov, 2004
- http://pub.ist.ac.at/~vnk/software.html
- https://xip.uclb.com/i/software/maxflow_computervision.html

IBFS:

- Faster and More Dynamic Maximum Flow by Incremental Breadth-First Search, Goldberg et. al., 2015
- http://web.archive.org/web/20170703042834/http://www.cs.tau.ac.il/~sagihed/ibfs/
- https://github.com/PolarNick239/IBFS/tree/53a0a3a974b7b5f2c495b81d003c87cb5e87d42a

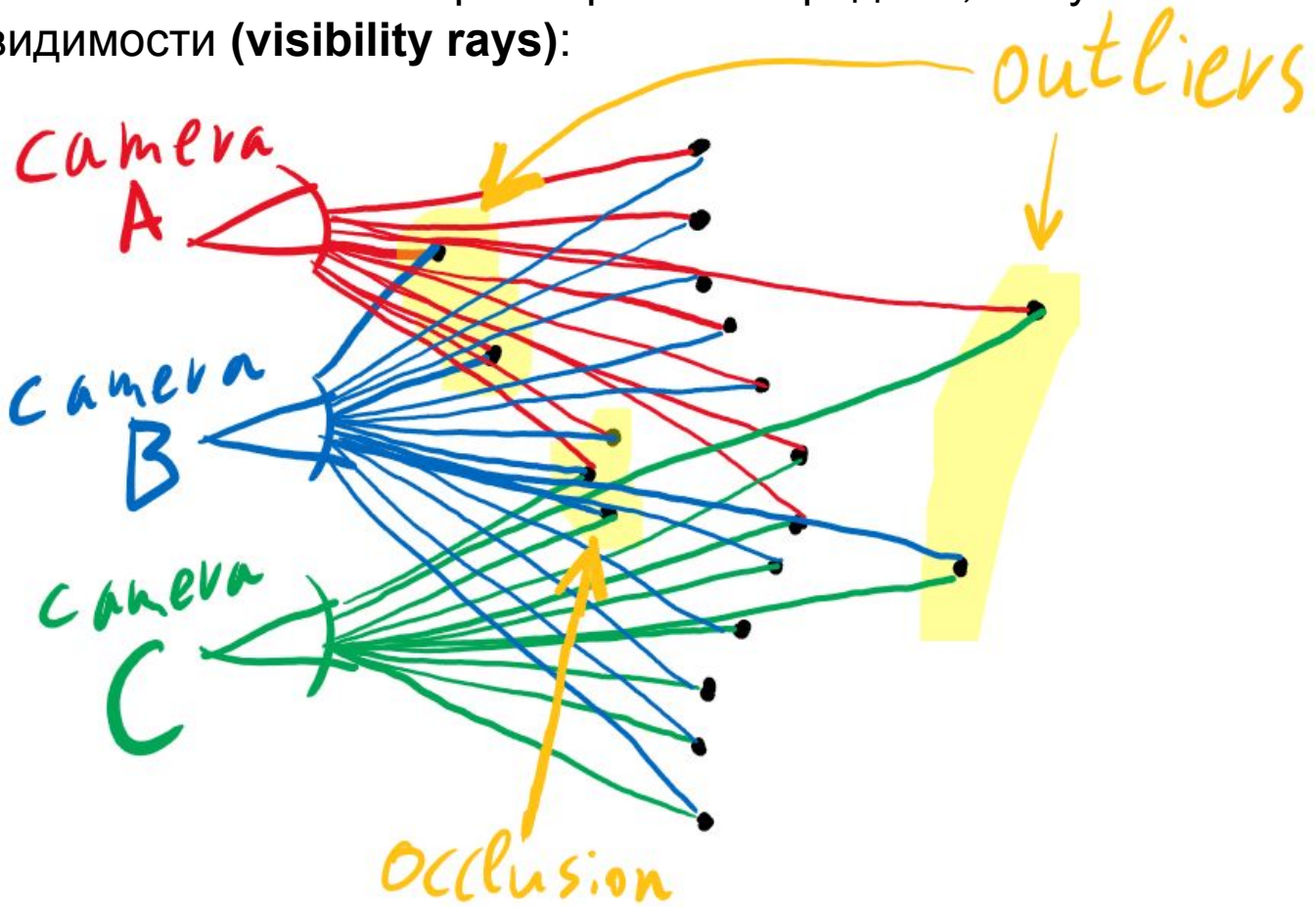
Алгоритм построения 3D модели

На вход даны 3D точки порожденные картами глубины:



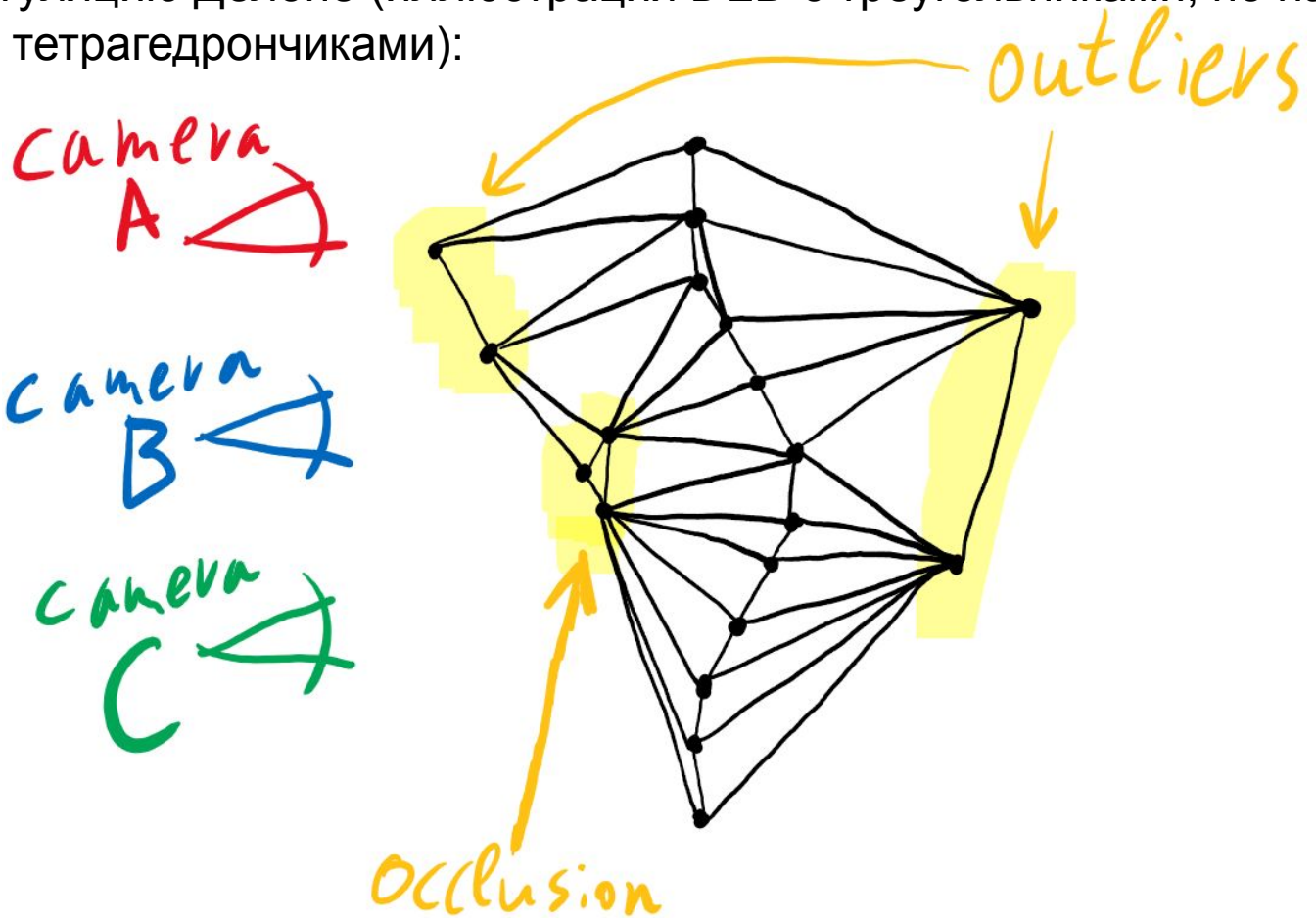
Алгоритм построения 3D модели

У каждой точки есть множество камер которые ее породили, т.е. у нас есть пучки отрезков видимости **(visibility rays)**:



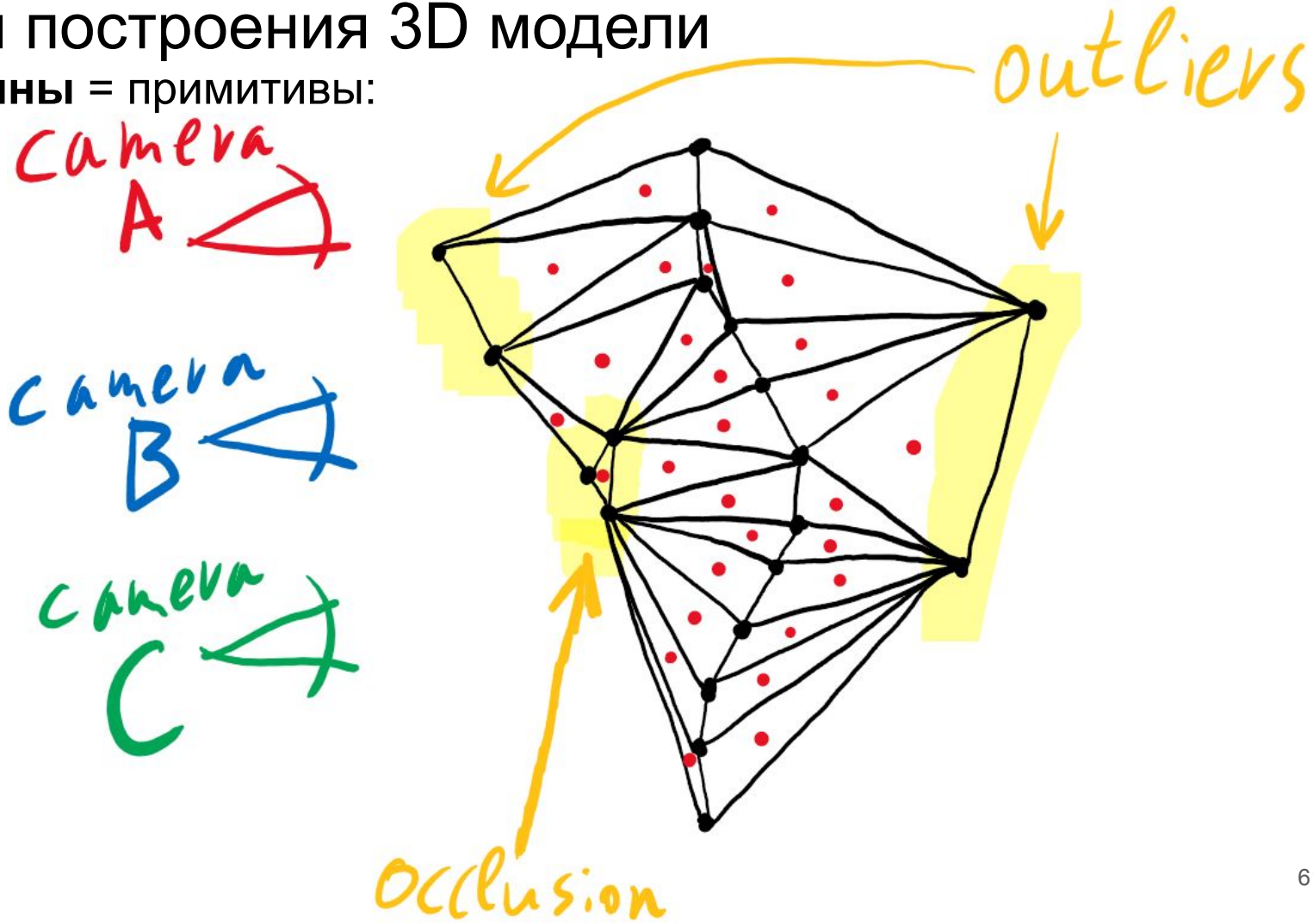
Алгоритм построения 3D модели

Построим триангуляцию Делоне (иллюстрация в 2D с треугольниками, но нас интересует 3D с тетрагедрончиками):



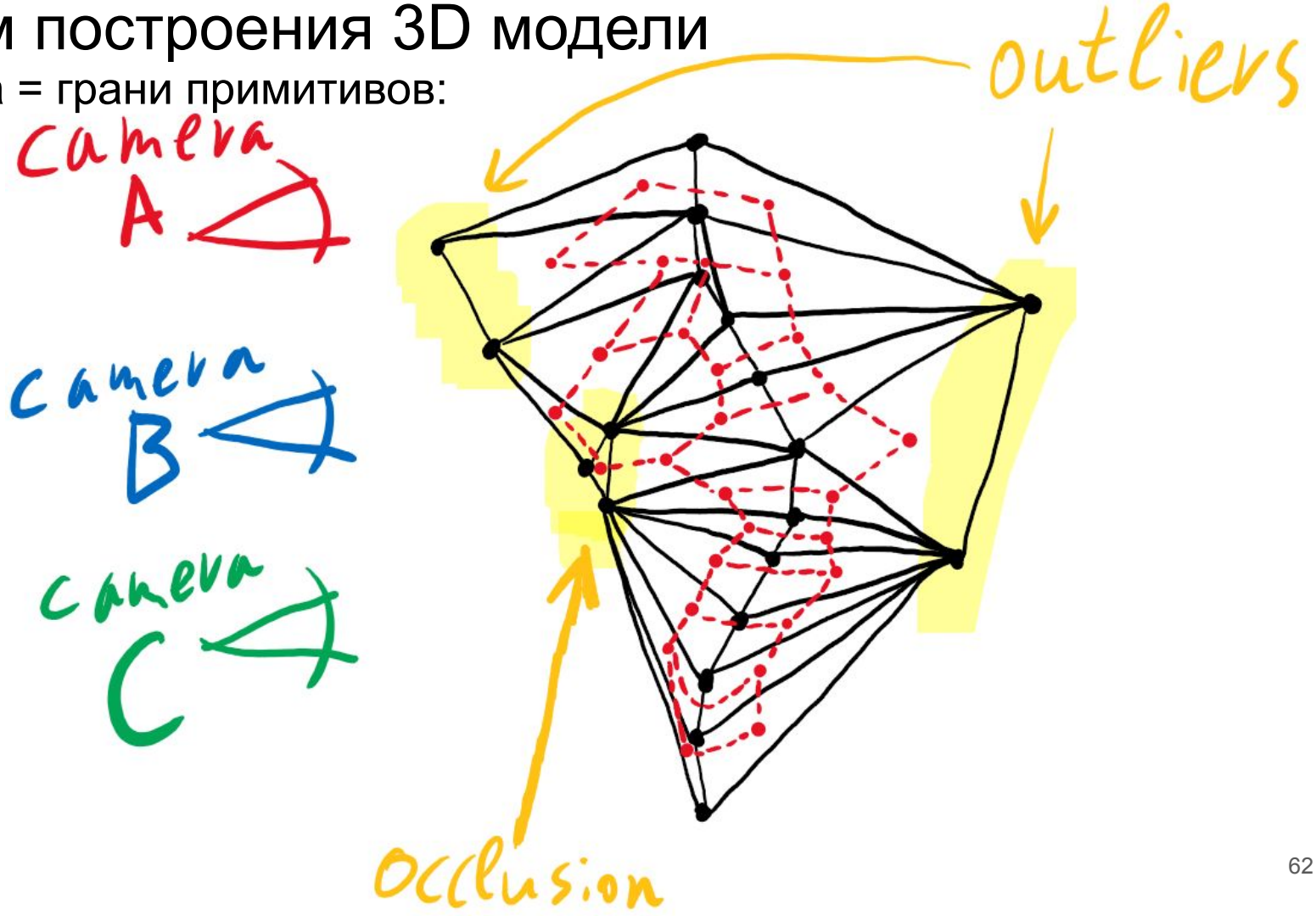
Алгоритм построения 3D модели

Граф: **вершины** = примитивы:



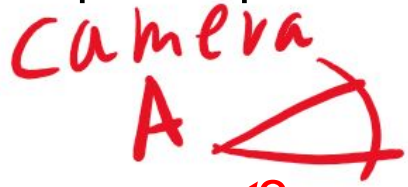
Алгоритм построения 3D модели

Граф: **ребра** = грани примитивов:

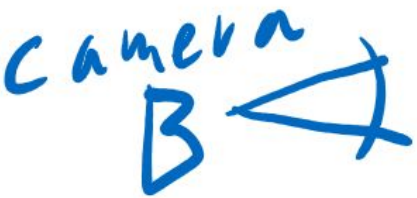


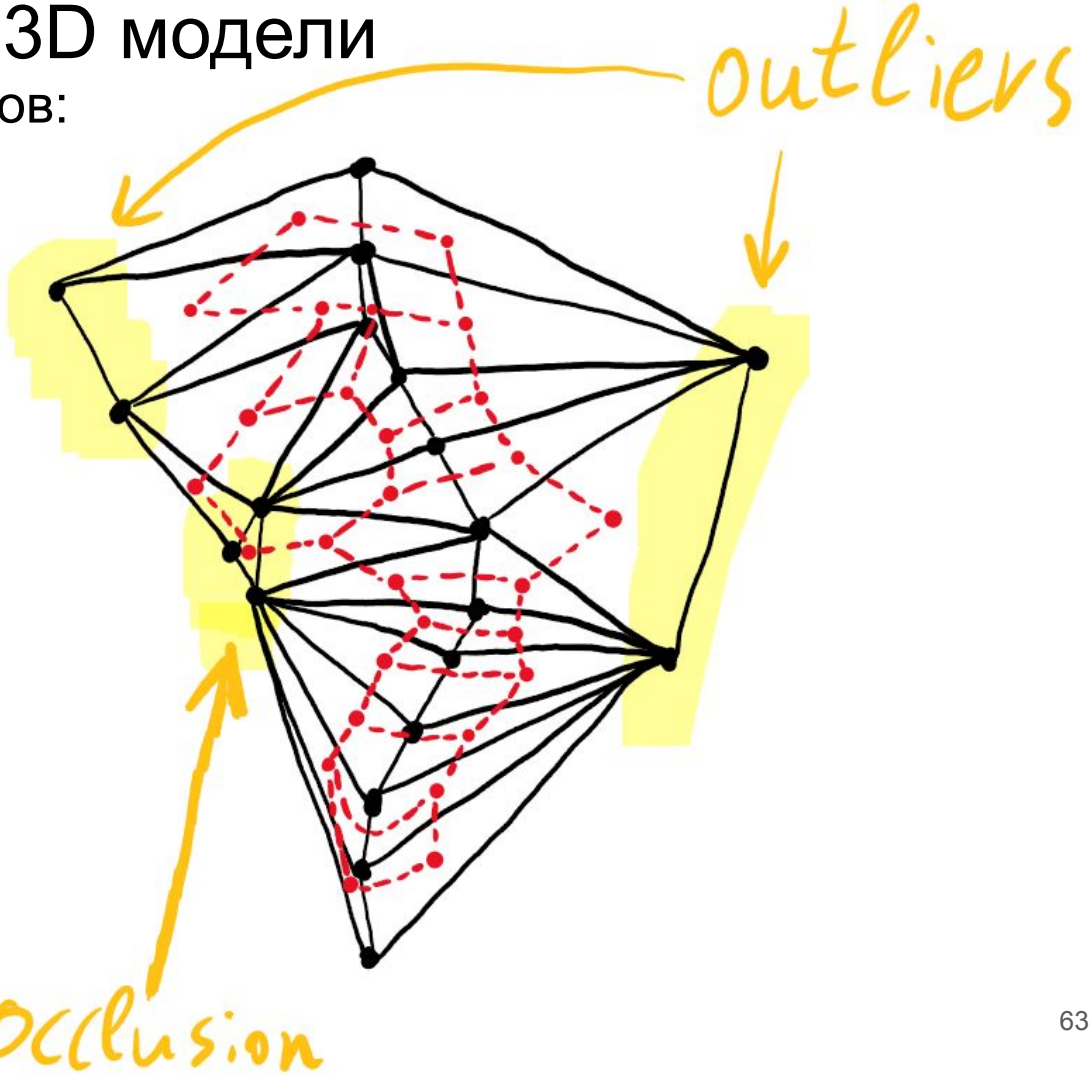
Алгоритм построения 3D модели

Граф: **ребра** = грани примитивов:



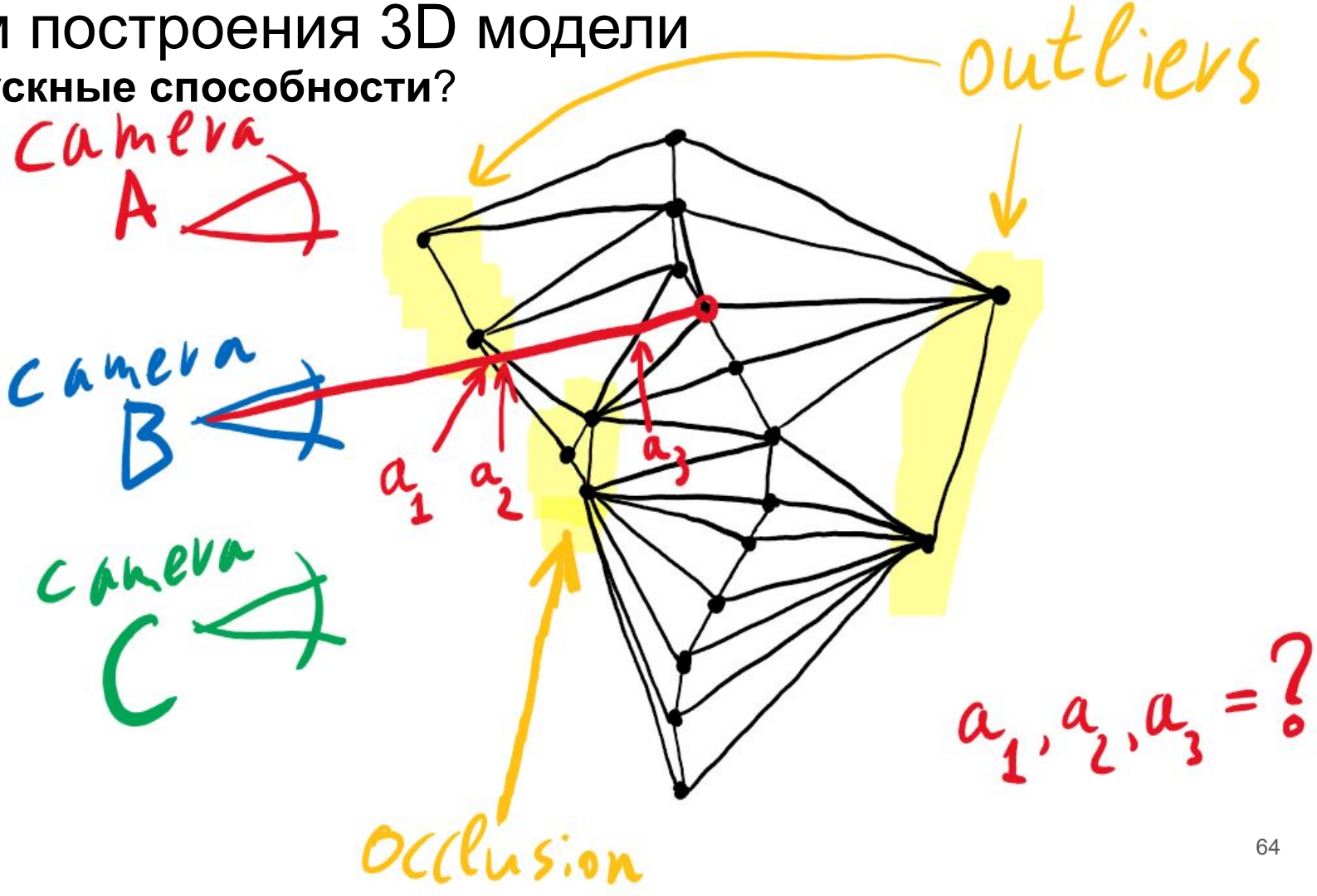
Но где исток ***s*** и сток ***t***?





Алгоритм построения 3D модели

Какие **пропускные способности**?



Полярный Николай
polarnick239@gmail.com