

# Локальные ключевые точки - SIFT.

Фотограмметрия. Лекция 1

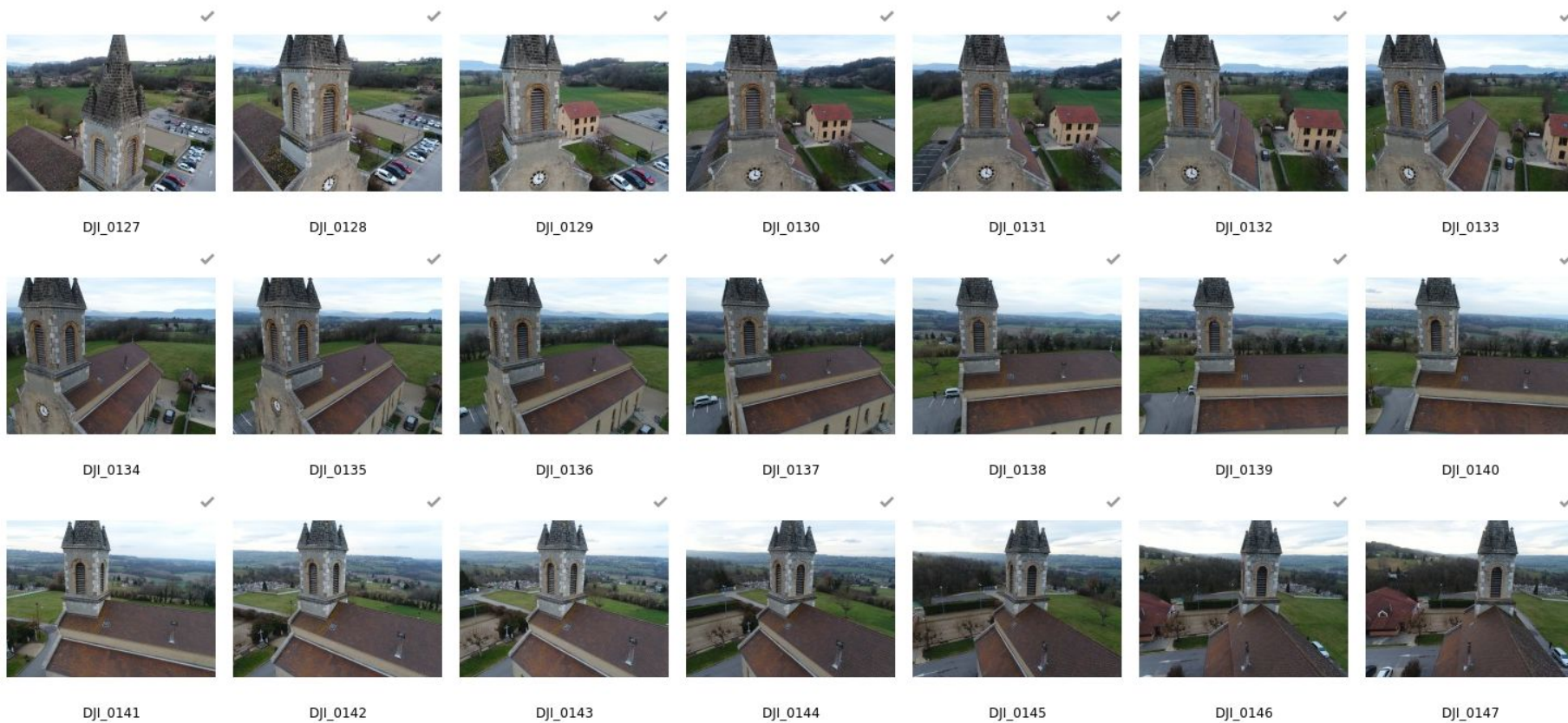


- план курса
- SIFT

Полярный Николай  
[polarnick239@gmail.com](mailto:polarnick239@gmail.com)

# Задача в целом

По множеству фотографий восстановить трехмерную модель.



# Задача в целом

По множеству фотографий восстановить трехмерную модель.

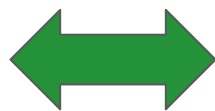


Данные предоставил Stéphane Prodent

# План курса

- 1) [ДЗ] **Ключевые точки (SIFT)**
- 2) [ДЗ] **Сопоставление ключевых точек (RANSAC, k-ratio test)**
- 3) [ДЗ] **Определение взаимного расположения пар камер**
- 4) [ДЗ] **Определение расположения всех камер (Bundle Adjustment)**
- 5) **Построение карт глубины (Semi-Global Matching)**
- 6) [ДЗ] **Построение карт глубины (Patch Match)**
- 7) **Обнаружение 3D проводов**
- 8) **Объединение карт глубины в облако точек и Poisson реконструкция**
- 9) [ДЗ] **Реконструкция 3D модели (Delaunay triangulation + graph min-cut)**
- 10) [ДЗ] **Реконструкция 3D модели (TVL1, TGV, out-of-core)**
- 11) **Текстурирование**
- 12) **Автоматическое планирование облета объекта (3D Mission Planner)**

# SIFT. Задача



Как алгоритму понять что это одна и та же поверхность?

Как вообще выбрать одни и те же точки слева и справа?

К каким искажениям выбор точек (детектирование) должен быть инвариантен?

Как сопоставить одни и те же точки? Как оценить похожесть? (описание точки)

# SIFT. Задача

- **Detect:** Инвариантно к искажениям выбрать ключевые точки
- **Describe:** Построить для каждой точки описание в виде вектора чисел

Статья: [Distinctive Image Features from Scale-Invariant Keypoints, G. Lowe, 2004](#)

# SIFT. Этапы

- 1) Построить пирамиды гауссова размытия  
(разбитые на октавы ради оптимизации)
- 2) Построить пирамиды DoG

# SIFT. Этапы

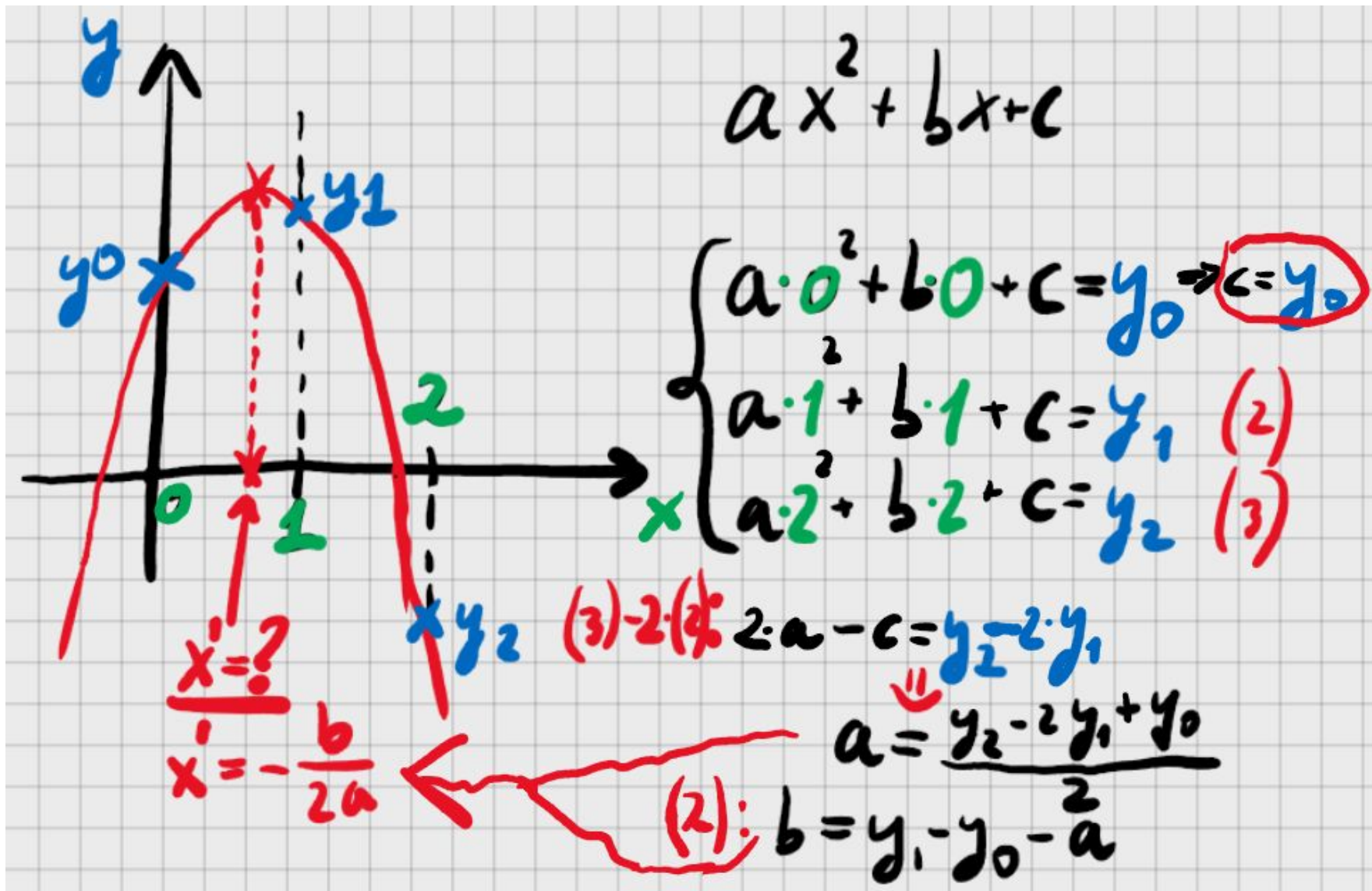
- 1) Построить пирамиды гауссова размытия  
(разбитые на октавы ради оптимизации)
- 2) Построить пирамиды DoG
- 3) Извлечь 3D экстремумы из пирамиды DoG



# SIFT. Этапы

- 1) Построить пирамиды гауссова размытия  
(разбитые на октавы ради оптимизации)
- 2) Построить пирамиды DoG
- 3) Извлечь 3D экстремумы из пирамиды DoG
- 4) Суб-пиксельно уточнить их координаты

# Субпиксельное уточнение (школьная парабола)



# Субпиксельное уточнение (парабола через ряд Тейлора)

$$f(x) = a \cdot x^2 + b \cdot x + c$$

$$f(x') = \text{extremum}$$

↑???

$$f'(x') = 0$$

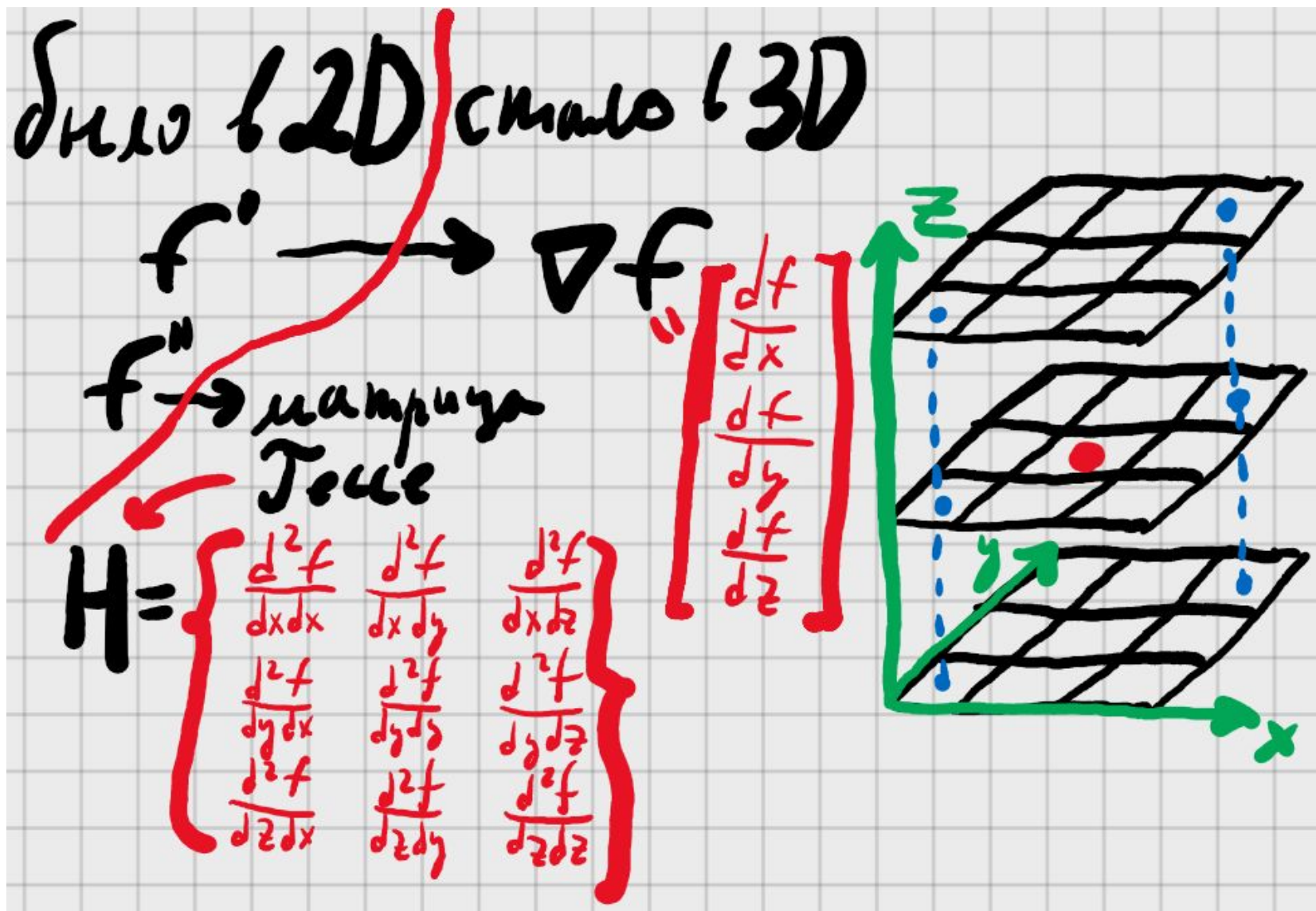
$$f(x) \sim f(x_0) + f'(x_0)dx + \frac{1}{2}f''(x_0)dx^2$$

$$f'(x) \sim 0 + f'(x_0) + \frac{1}{2}f''(x_0)dx \cdot 2 = 0$$

$$dx = - \frac{f'(x_0)}{f''(x_0)}$$

$$x' = x_0 + dx$$

# Субпиксельное уточнение в 3D (через ряд Тейлора и матрицу Гессеана)



# Субпиксельное уточнение в 3D (через ряд Тейлора и матрицу Гессiana)

$$f(x) \sim f(x_0) + dx^T \nabla f(x_0) + \frac{1}{2} dx^T [H f(x_0)] dx$$
$$\nabla f(x) \sim 0 + \nabla f(x_0) + \frac{1}{2} [H f(x_0)] dx = 0$$
$$dx = [H f(x_0)]^{-1} (-\nabla f(x_0))$$

*(решить уравнение устойчивее, чем обращение и умножать)*

$$[H f(x_0)] dx = -\nabla f(x_0)$$

*напр. в OpenCV:*

```
Vec3f X = H.solve(dD, DECOMP_LU)
```

SIFT в OpenCV: [opencv/features2d/src/sift\\_simd.hpp#L334](https://github.com/opencv/opencv/blob/master/src/sift_simd.hpp#L334)

# SIFT. Этапы

- 1) Построить пирамиды гауссова размытия  
(разбитые на октавы ради оптимизации)
- 2) Построить пирамиды DoG
- 3) Извлечь 3D экстремумы из пирамиды DoG
- 4) Суб-пиксельно уточнить их координаты
- 5) Определить масштаб

# SIFT. Этапы

- 1) Построить пирамиды гауссова размытия  
(разбитые на октавы ради оптимизации)
- 2) Построить пирамиды DoG
- 3) Извлечь 3D экстремумы из пирамиды DoG
- 4) Суб-пиксельно уточнить их координаты
- 5) Определить масштаб
- 6) Определить ориентацию (с учетом масштаба)

# SIFT. Этапы

- 1) Построить пирамиды гауссова размытия (разбитые на октавы ради оптимизации)
- 2) Построить пирамиды DoG
- 3) Извлечь 3D экстремумы из пирамиды DoG
- 4) Суб-пиксельно уточнить их координаты
- 5) Определить масштаб
- 6) Определить ориентацию (с учетом масштаба)
- 7) Построить HoG дескриптор (с учетом ориентации и масштаба)



# SIFT. Ссылки

**SIFT:** [Distinctive Image Features from Scale-Invariant Keypoints, G. Lowe, 2004](#)

Примеры реализации: [OpenSIFT](#) и его адаптации - [by lxc-xx](#) и в OpenCV ([sift.dispatch.cpp](#), [sift.simd.hpp](#)).

Сравнения разных детекторов и дескрипторов:

- 1) [Local Invariant Feature Detectors: A Survey, T. Tuytelaars, K. Mikolajczyk, 2008](#)
- 2) [A survey of recent advances in visual feature detection, Y. Li et al., 2015](#)
- 2) [Recent Advances in Features Extraction and Description Algorithms: A Comprehensive Survey, E. Salahat, M. Qasaimeh, 2017](#)

**Очень советую** <https://www.semanticscholar.org/> для того чтобы найдя одну хорошую статью - идти по цепочкам цитат как в статье от которых эта статья отталкивалась, так и в статьи которые ссылаются на эту статью.

# Вопросы?



Полярный Николай  
[polarnick239@gmail.com](mailto:polarnick239@gmail.com)