

Методы 3D реконструкции

Александр Шишков

Itseez, 2015

Одометрия

- **Одо́метр** (греч. ὁδός — дорога + μέτρον — мера), в просторечии счётчик — прибор для измерения количества оборотов колеса. При помощи него может быть измерен пройденный транспортным средством путь.

<https://ru.wikipedia.org/wiki/Одометр>



Какие устройства мы можем использовать?

GPS/Глонасс

- Не работает в помещении
- Большая погрешность

Одометр

- Скольжение
- Большая погрешность



Какие устройства мы можем использовать?

**IMU (Акселерометер+Гироскоп
+Магнетометер)**

- большая погрешность
- неустойчив к помехам

Какие устройства мы можем использовать?

Что же остается?

Камера



Depth-сенсор

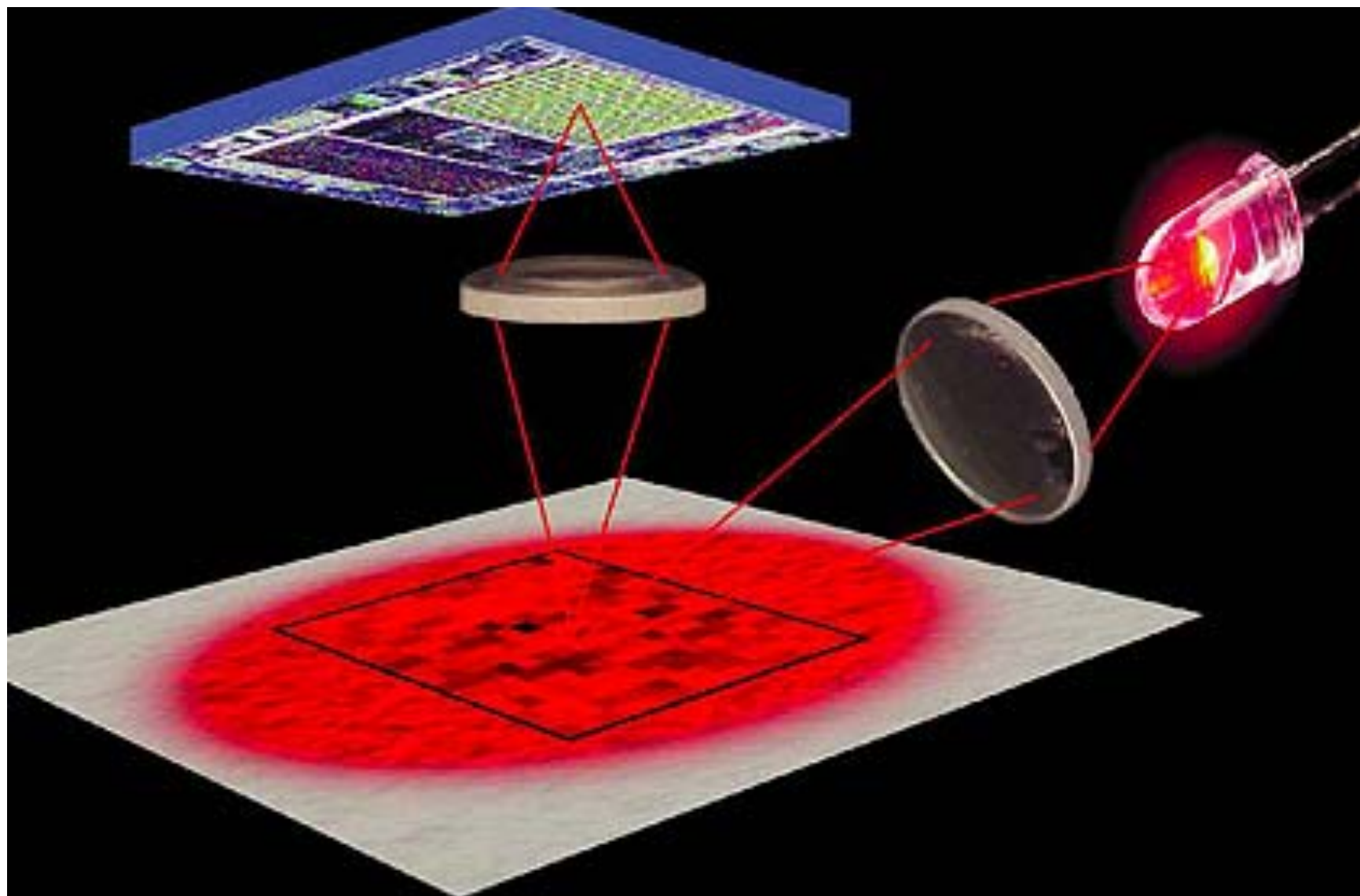


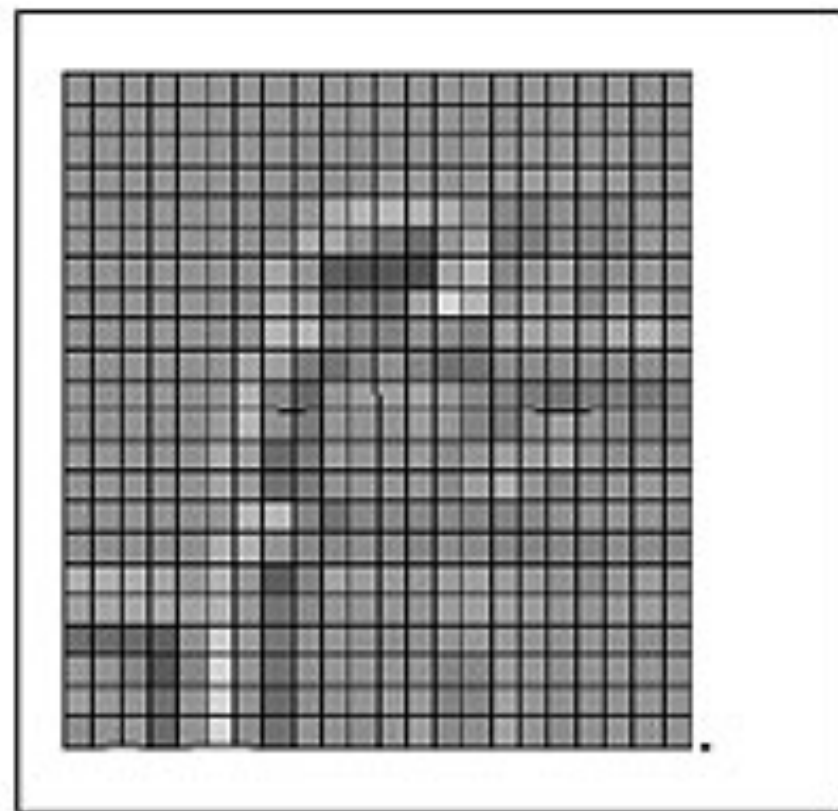
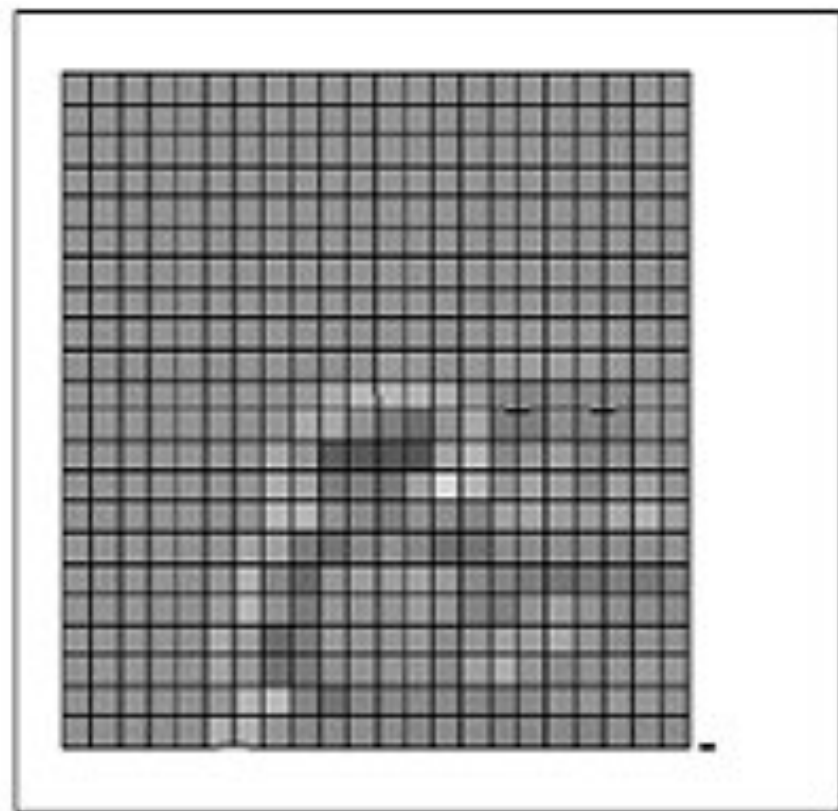
Лидар



- **Визуальная одометрия** — метод оценки положения и ориентации объекта на основе анализа последовательности изображений, снятых установленной на нем камерой

Какие примеры вы встречаете каждый день?



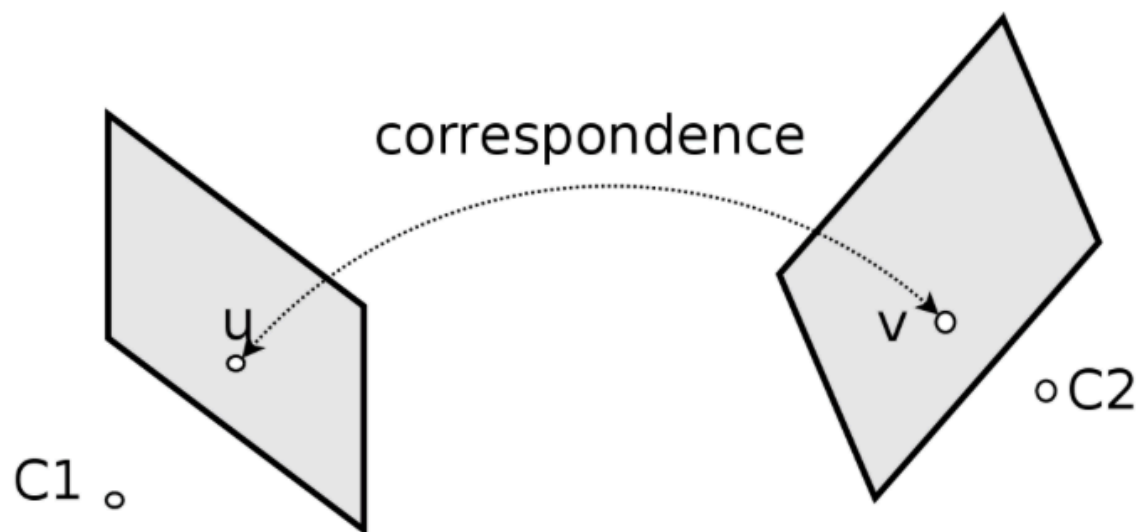


Постановка задачи

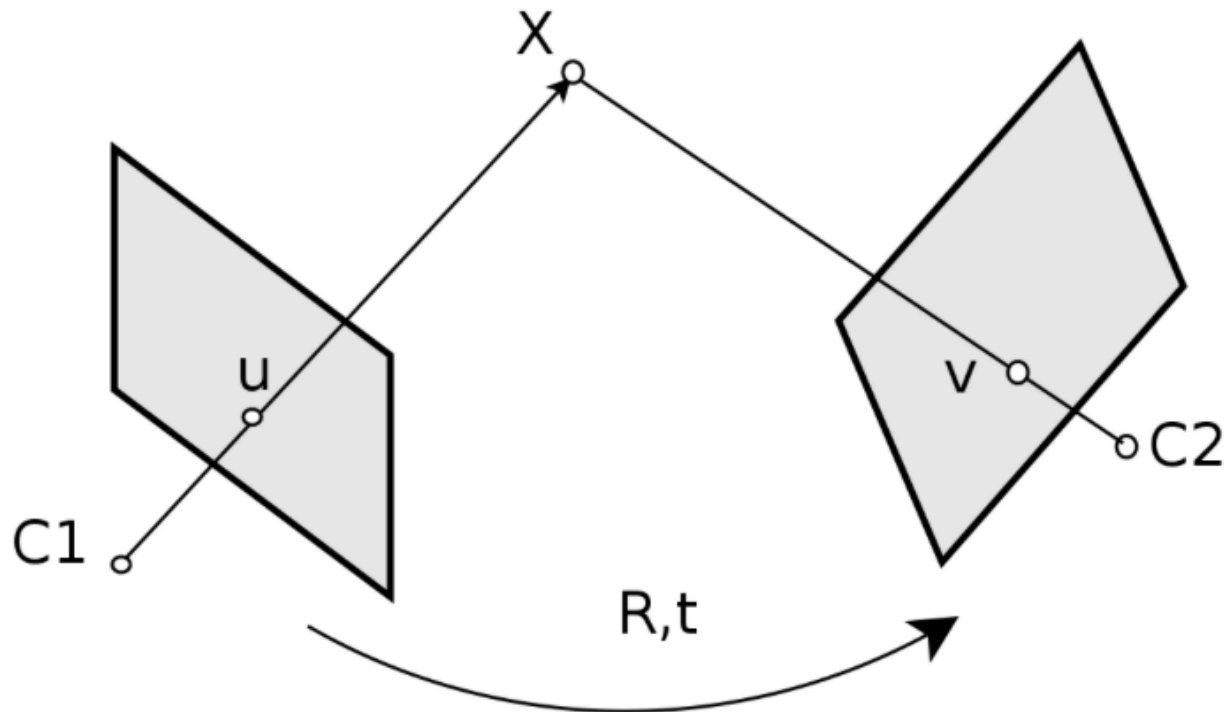
По заданной последовательности кадров определить положение камеры в некоторой системе координат. Обычно предполагается работа в реальном времени.



Рассмотрим два последовательных изображения и найдем соответствия между изображениями



Оценим поворот и сдвиг камеры между кадрами, минимизирующие ошибку репроекции



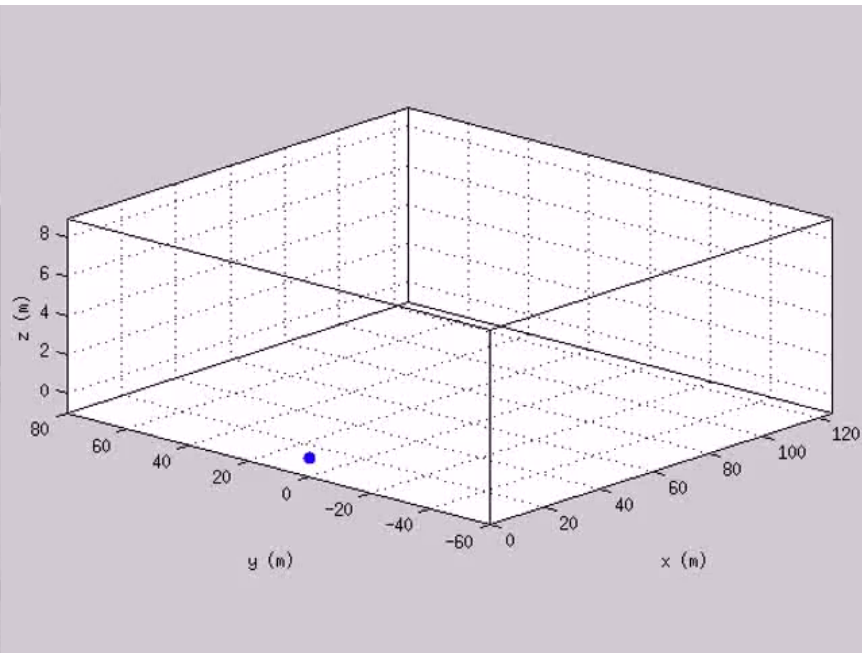
Пример алгоритма:

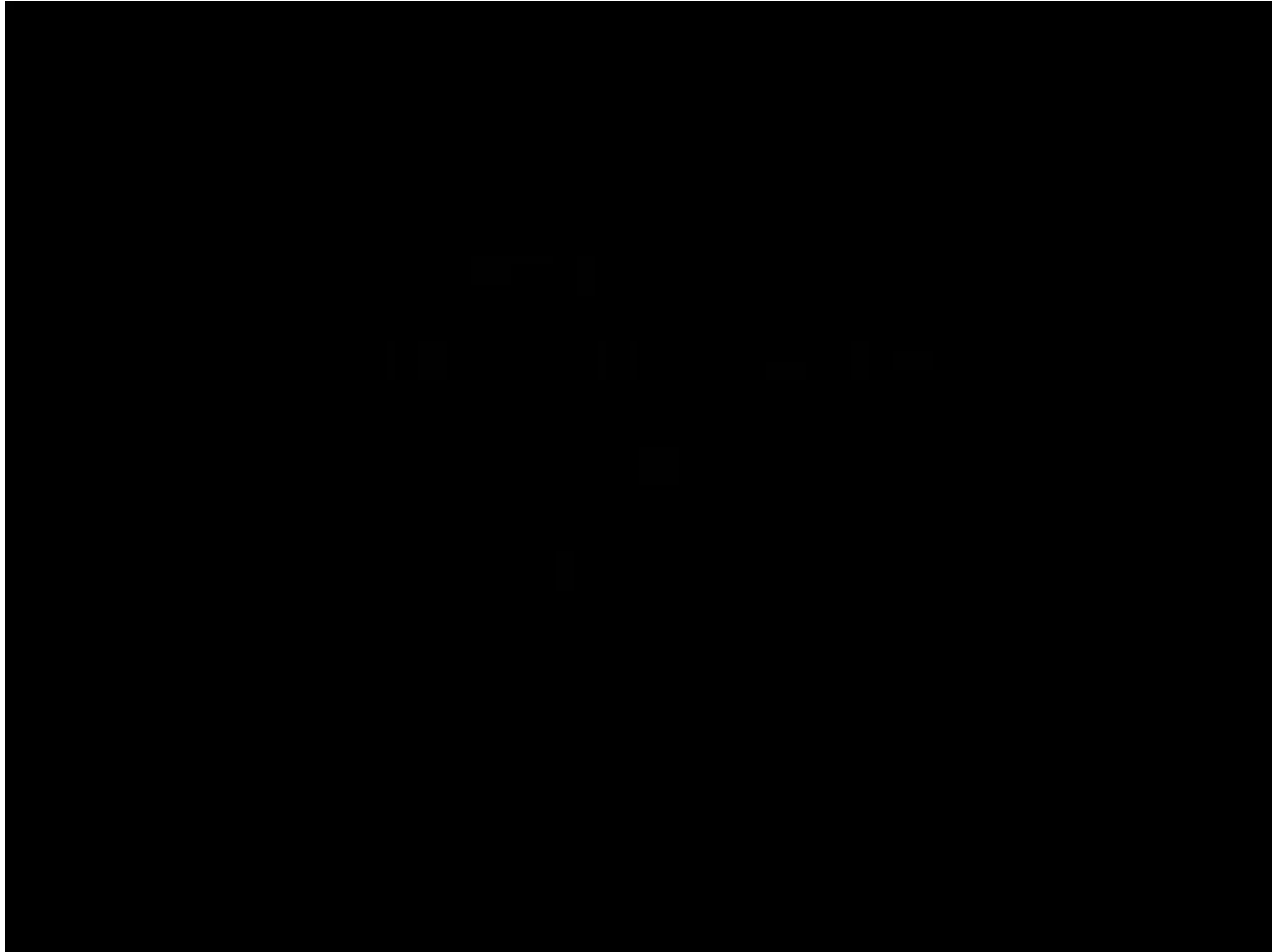
- Рассмотрим k -е изображение
- Найдем соответствия между k и $k-1$ изображениями (matching/tracking)
- Отфильтруем соответствия и найдем essential матрицу
- На основе найденной матрицы вычислим поворот и сдвиг
- Вычислим трехмерную модель особых точек
- Отнормируем масштаб точек
- ...

Возможно комбинировать устройства

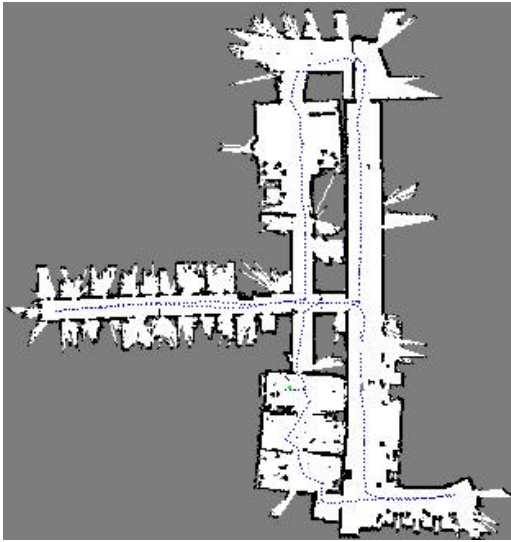
RGB-D

RGB-IMU

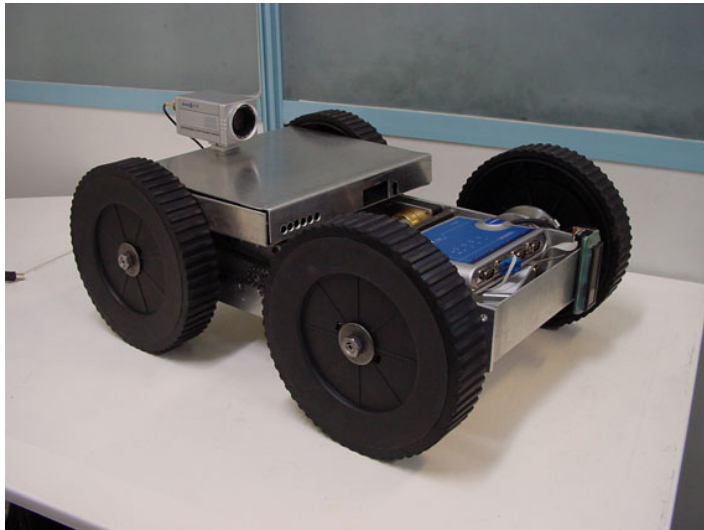




SLAM



SLAM: Simultaneous Localization and Mapping



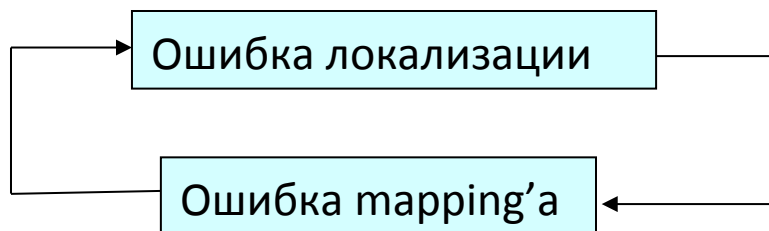
Autonomous Micro Air Vehicle Flight Indoors

Robust Robotics Group
CSAIL, MIT





Почему SLAM трудная задача?



- Ошибка быстро нарастает
- Ошибки возникают из-за неточных измерений фактического движения робота (шум в одометрии) и расстояний до препятствий (landmark) (шум в наблюдении)
- Loop-close problem



Введение: SLAM

SLAM: Simultaneous Localization and Mapping

Робот изучает незнакомое, статическое окружение.

Дано:

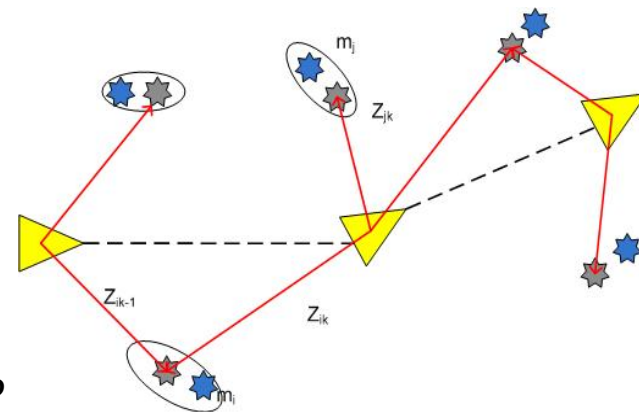
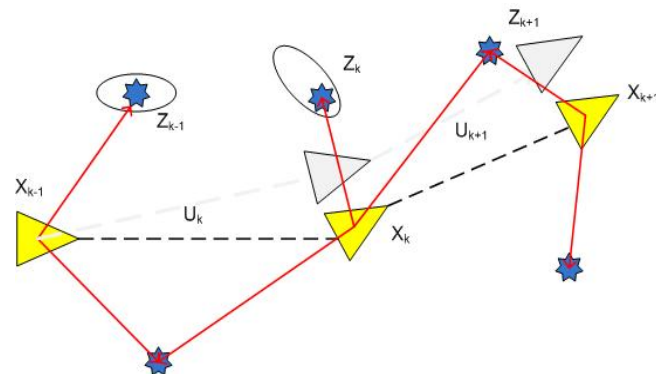
- Система одометрии
- Лазерный сонар

Оба источника данных зашумлены.

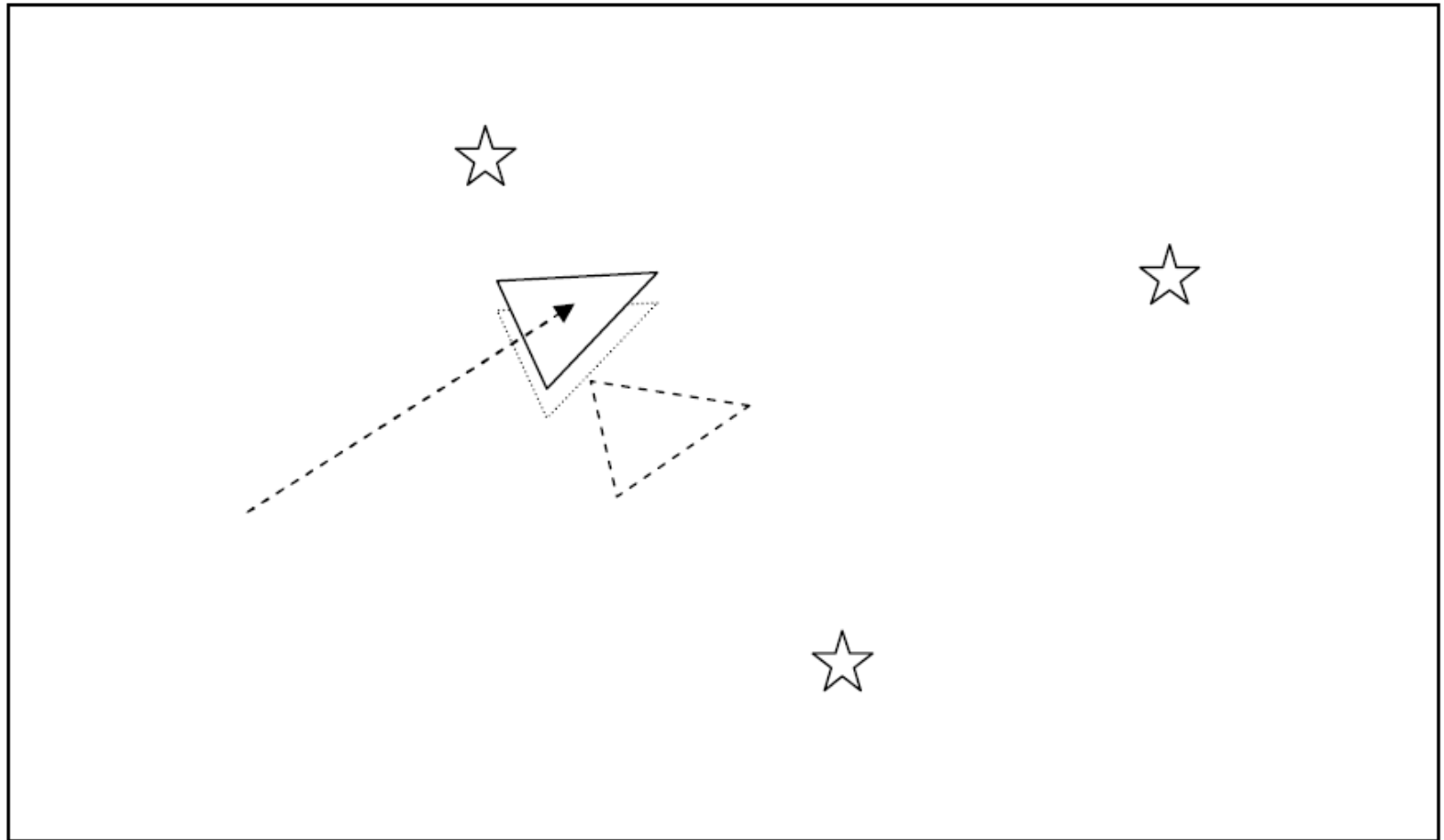
Оценка:

- Положения робота -- **localization**
где Я ?
- Детализация окружающего пространства
-- **mapping**

На что похоже то, что вокруг меня?



SLAM EKF



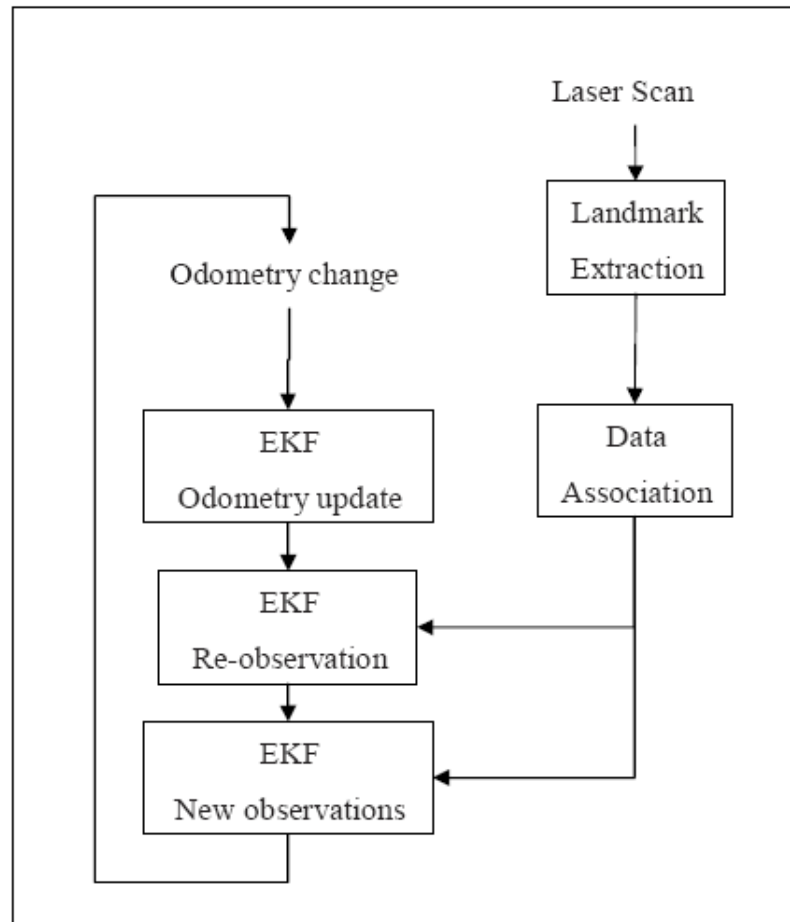
SLAM EKF

Постановка задачи

- $X_{0:k} = \{x_0, x_1, \dots, x_k\} = \{X_{0:k-1}, x_k\}$ – история положений робота
- $Z_{0:k} = \{z_1, z_2, \dots, z_k\} = \{Z_{0:k-1}, z_k\}$ - положения наблюдаемых особенностей
- $U_{0:k} = \{u_1, u_2, \dots, u_k\} = \{U_{0:k-1}, u_k\}$ – история управлений
- $m = \{m_1, m_2, \dots, m_n\}$ – множество особенностей

Задача - в каждый момент времени уметь вычислять $P(x_k, m | Z_{0:k}, U_{0:k}, x_0)$. Часто для решения задачи используют рекуррентные соотношения, в связи с чем вводят модель движения: $P(x_k | x_{k-1}, u_k)$ и модель наблюдения $P(z_k | x_k, m)$.

SLAM EKF



SLAM EKF

- Алгоритм состоит из трех частей:
 - Обновление текущего состояния системы на основе одометрии
 - Обновление полученного состояния на основе видимых особенностей
 - Добавление новых особенностей к текущему состоянию системы

SLAM EKF

- $P(x_k | x_{k-1}, u_k) \Leftrightarrow x_k = f(x_{k-1}, u_k) + w_k$, где f – кинематическая модель движения, а w_k – гауссов шум с заданной ковариационной матрицей Q_k .
- $P(z_k | x_k, m) \Leftrightarrow z_k = h(x_k, m) + v_k$, где h – геометрическая модель наблюдения, а v_k гауссов шум с ковариационной матрицей R_k .

SLAM EKF

Состояние модели:

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{m}}_k \end{bmatrix} = \mathbb{E} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{m} \end{bmatrix} | \mathbf{Z}_{0:k}$$

с матрицей ковариации

$$\begin{aligned} \mathbf{P}_{k|k} &= \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xm} \\ \mathbf{P}_{xm}^T & \mathbf{P}_{mm} \end{bmatrix}_{k|k} \\ &= \mathbb{E} \left[\begin{pmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_k \\ \mathbf{m} - \hat{\mathbf{m}}_k \end{pmatrix} \begin{pmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_k \\ \mathbf{m} - \hat{\mathbf{m}}_k \end{pmatrix}^T \mid \mathbf{Z}_{0:k} \right] \end{aligned}$$

SLAM EKF

Обновление текущего состояния системы на основе одометрии:

$$\begin{aligned}\hat{\mathbf{x}}_{k|k-1} &= \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k) \\ \mathbf{P}_{xx, k|k-1} &= \nabla \mathbf{f} \mathbf{P}_{xx, k-1|k-1} \nabla \mathbf{f}^T + \mathbf{Q}_k,\end{aligned}$$

где $\nabla \mathbf{f}$ – якобиан функции \mathbf{f} в точке \mathbf{x}_{k-1} .

SLAM EKF

Обновление полученного состояния на основе видимых особенностей и добавление новых особенностей к текущему состоянию системы

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{m}}_k \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} \\ \hat{\mathbf{m}}_{k-1} \end{bmatrix} + \mathbf{W}_k [\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}, \hat{\mathbf{m}}_{k-1})]$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{W}_k \mathbf{S}_k \mathbf{W}_k^T,$$

$$\mathbf{S}_k = \nabla \mathbf{h} \mathbf{P}_{k|k-1} \nabla \mathbf{h}^T + \mathbf{R}_k$$

$$\mathbf{W}_k = \mathbf{P}_{k|k-1} \nabla \mathbf{h}^T \mathbf{S}_k^{-1}$$

где $\nabla \mathbf{h}$ якобиан функции \mathbf{h} в точке \mathbf{x}_{k-1} .

SLAM EKF

Недостатки алгоритма:

- Некорректное ассоциирование меток
- Нелинейность
- Вычислительная сложность
 - Сложность EKF SLAM $O(N^2)$

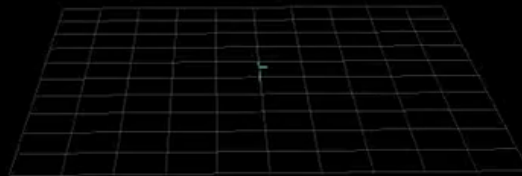
```
jlblanco@jlblanco-laptop2: ~/code/mrpt/src/share/mrpt/config_files/kf-slam
File Edit View Terminal Tabs Help

jlblanco@jlblanco-laptop2: ~/code/mrpt... x jlblanco@jlblanco-laptop2: ~/code/mrpt... x jlblanco@jlblanco-laptop2: ~/code/mrpt... x
jlblanco en [kf-slam] >> kf-slam EKF-SLAM_60_test.ini []
```


Loop closure

VSLAM on GPU

- GPU SURF feature detector
 - GPU BruteForceMatcher
 - GPU StereoBM
- x2.7 speedup



SFM

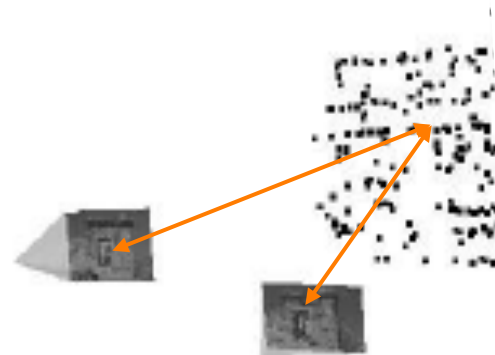


Восстановить трехмерную модель
одновременно с положениями и
параметрами камер на основе
неупорядоченного набора изображений.

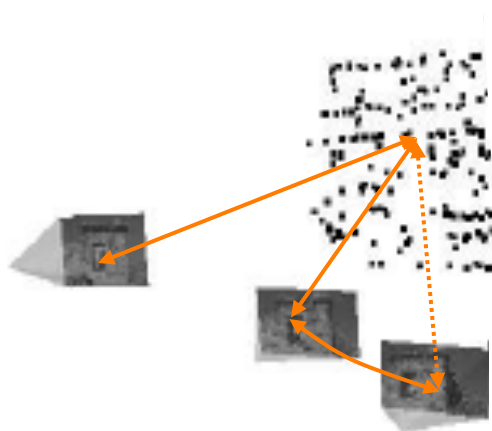
SFM



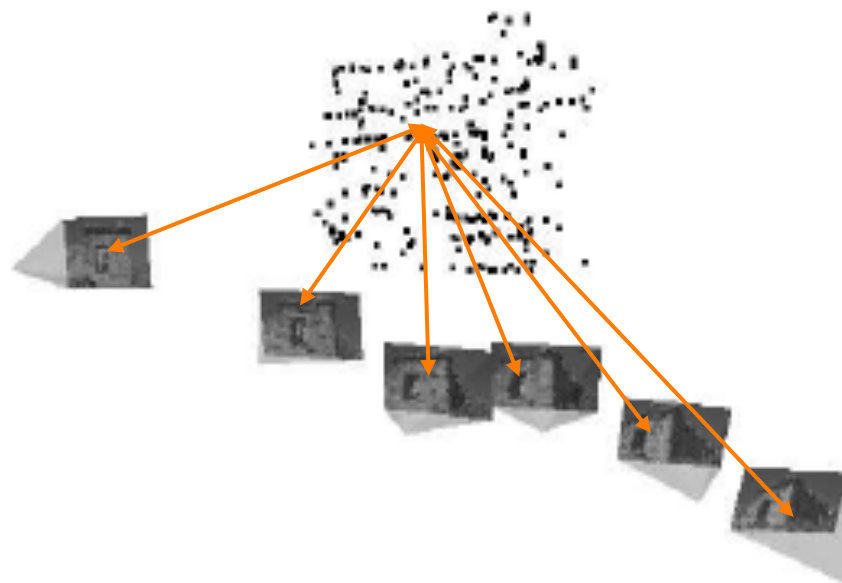
Инициализация структуры



Минимизация ошибки репроекции



Расширение модели

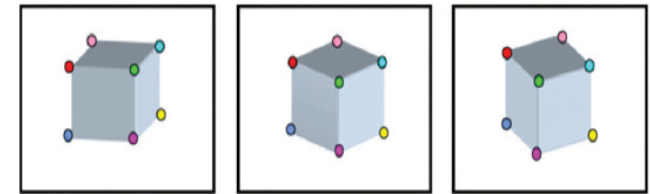


Алгоритм

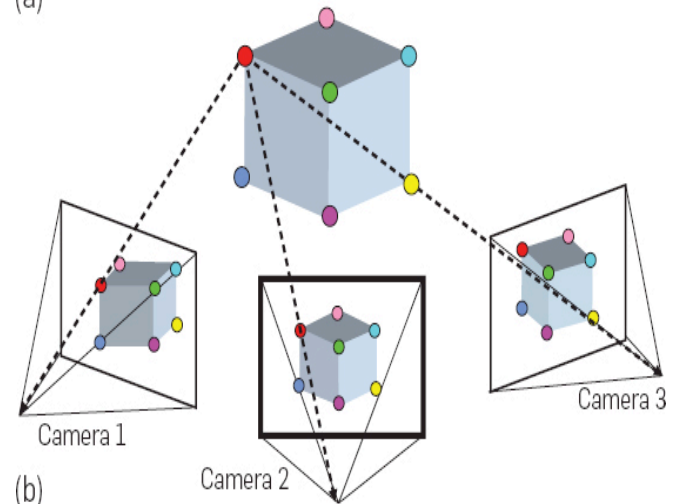
- Получить надежные соответствия (matching/tracking) на 2/3-изображениях
- Вычислить начальное приближение структуры
- Улучшить структуру и позы за счет новых изображений

Bundle adjustment problem

- ‘bundle’ – пучок лучей света исходящий из 3D точки и проходящий через центры камер

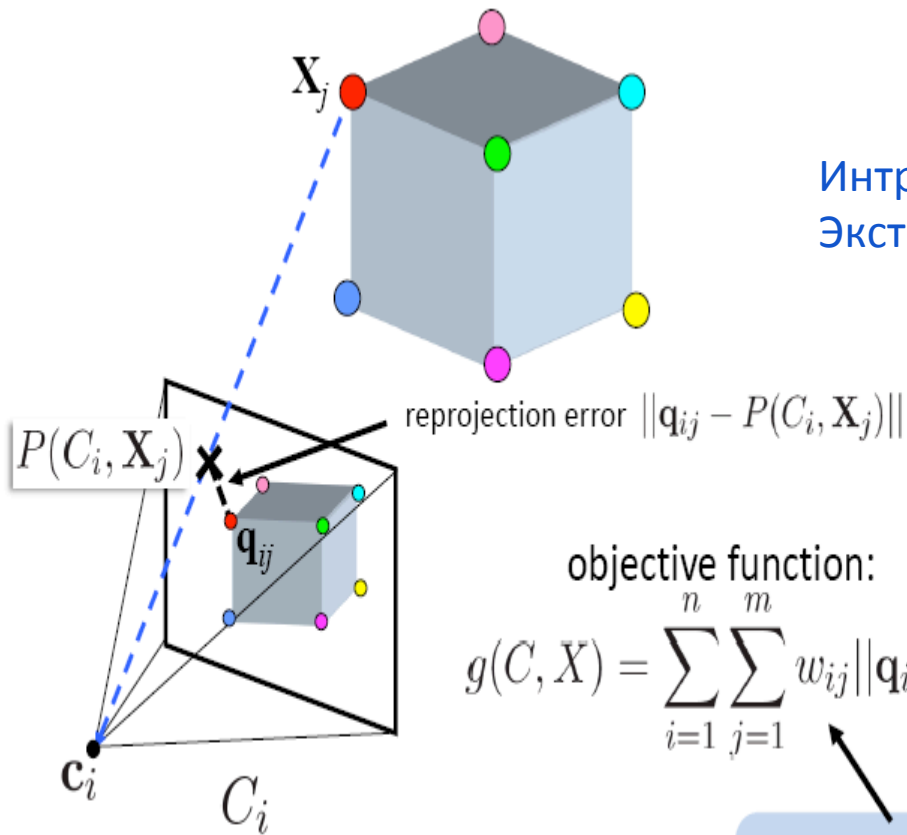


(a)



(b)

Bundle adjustment problem

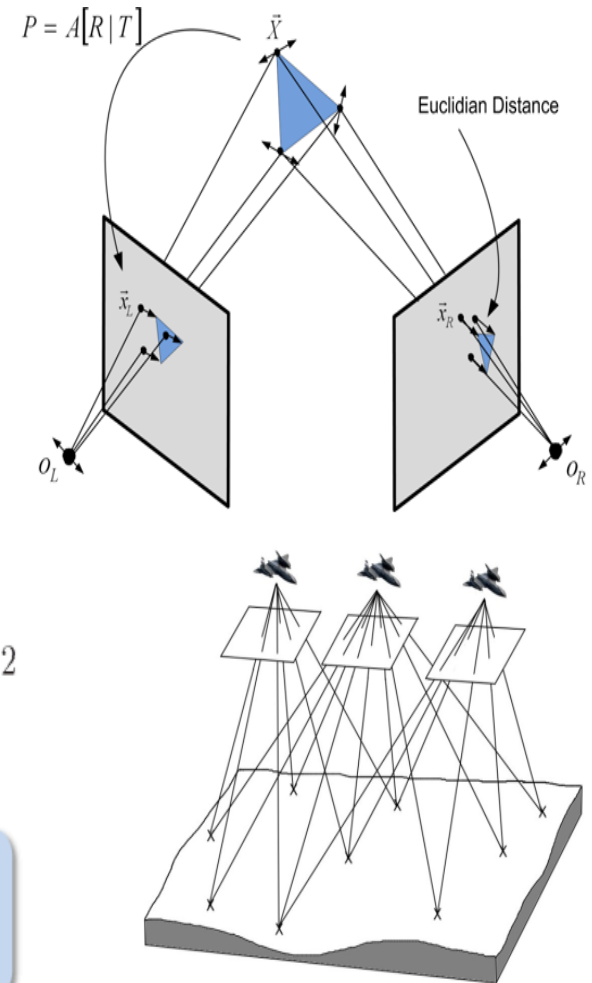


Интринсики (K, D)
Экстринсики (R | t)

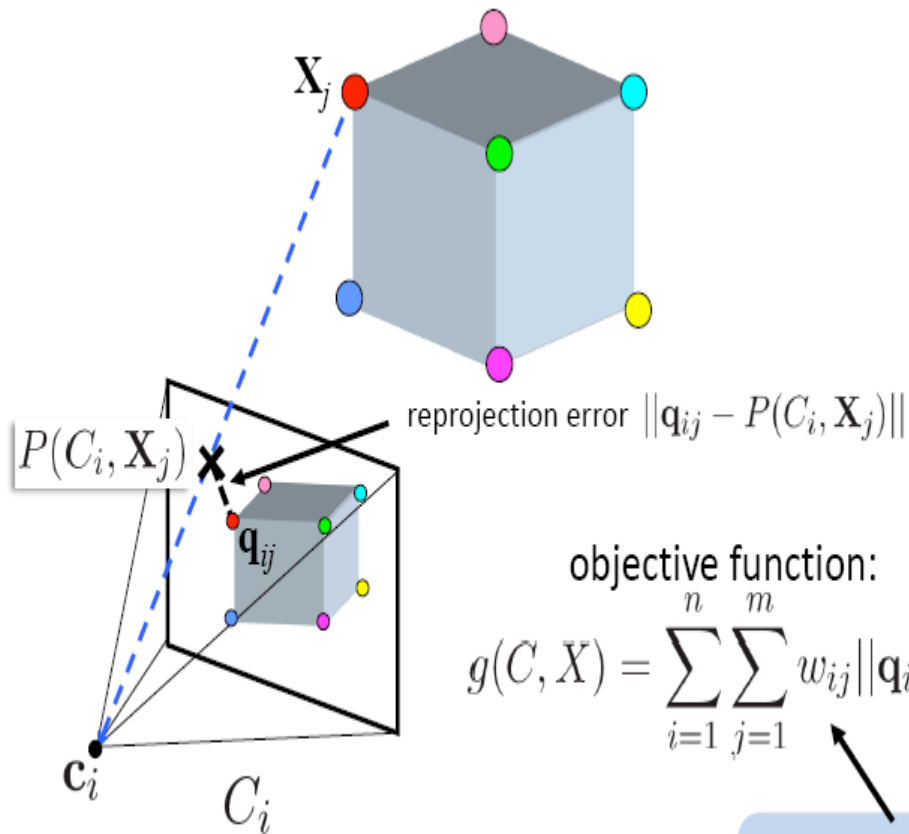
objective function:

$$g(C, X) = \sum_{i=1}^n \sum_{j=1}^m w_{ij} \|q_{ij} - P(C_i, X_j)\|^2$$

indicator variable:
1 if point j is visible in camera i
0 otherwise



Bundle adjustment problem



$P(R, t, f_x, f_y, c_x, c_y, k_1..k_6)$ -
проекция

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$

$$y' = y/z$$

$$x'' = x' \frac{1+k_1 r^2 + k_2 r^4 + k_3 r^6}{1+k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 x' y' + p_2 (r^2 + 2x'^2)$$

$$y'' = y' \frac{1+k_1 r^2 + k_2 r^4 + k_3 r^6}{1+k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1 (r^2 + 2y'^2) + 2p_2 x' y'$$

$$\text{where } r^2 = x'^2 + y'^2$$

$$u = f_x * x'' + c_x$$

$$v = f_y * y'' + c_y$$

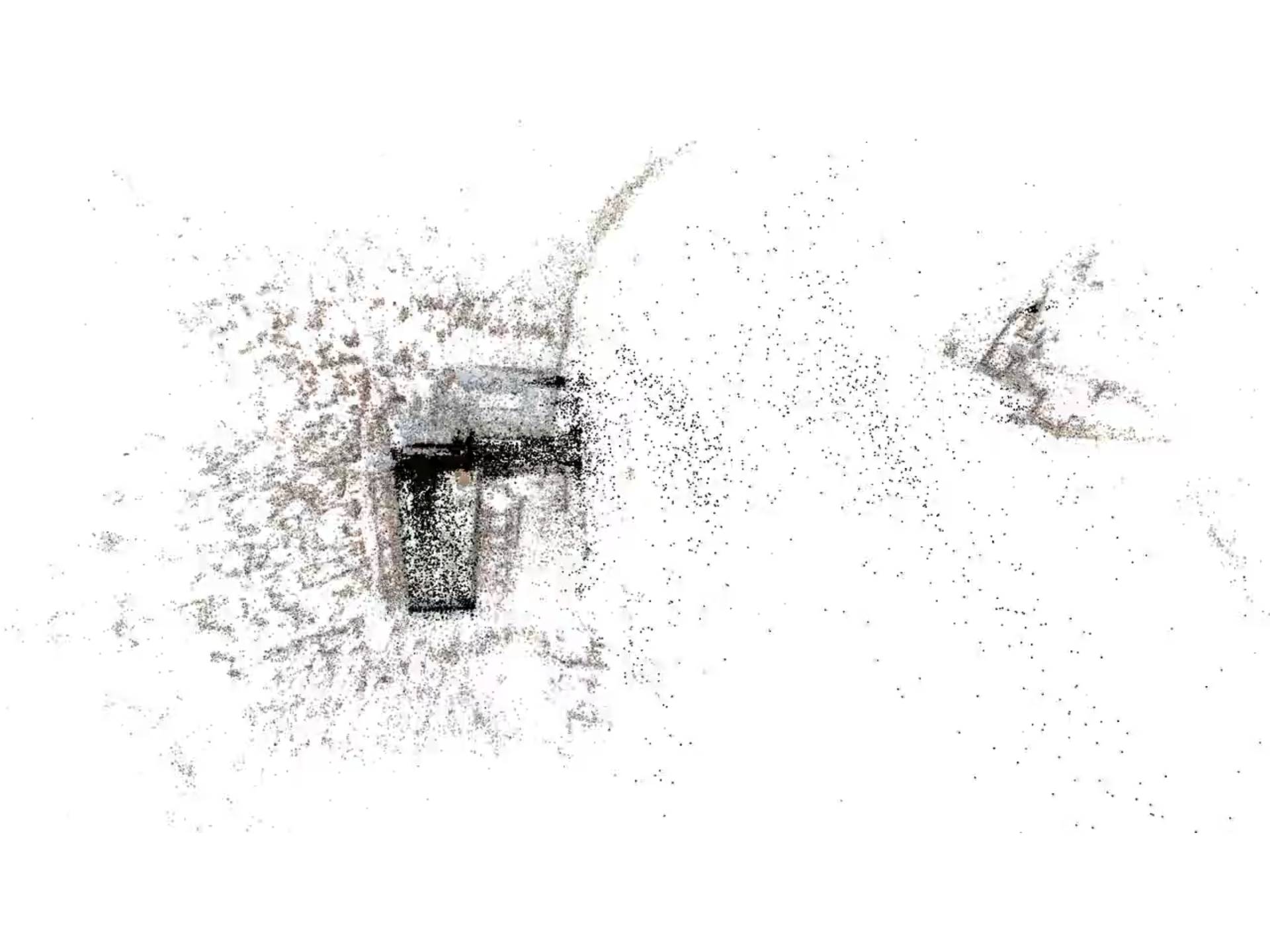
objective function:

$$g(C, X) = \sum_{i=1}^n \sum_{j=1}^m w_{ij} \|q_{ij} - P(C_i, X_j)\|^2$$

indicator variable:

1 if point j is visible in camera i
0 otherwise

Нелинейная!



Вопросы???