

# МНОГОПОТОЧНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА ВИЗУАЛЬНОЙ ЛОКАЛИЗАЦИИ БПЛА НА ОСНОВЕ ИЗВЕСТНОЙ 3D МОДЕЛИ ОКРУЖЕНИЯ И ТЕХНОЛОГИИ CUDA

к.т.н., доц. Буйвал А. К.

асп., Гавриленков М.А.

Брянский Государственный Технический Университет

# Визуальная локализация



Локализация по маркерам

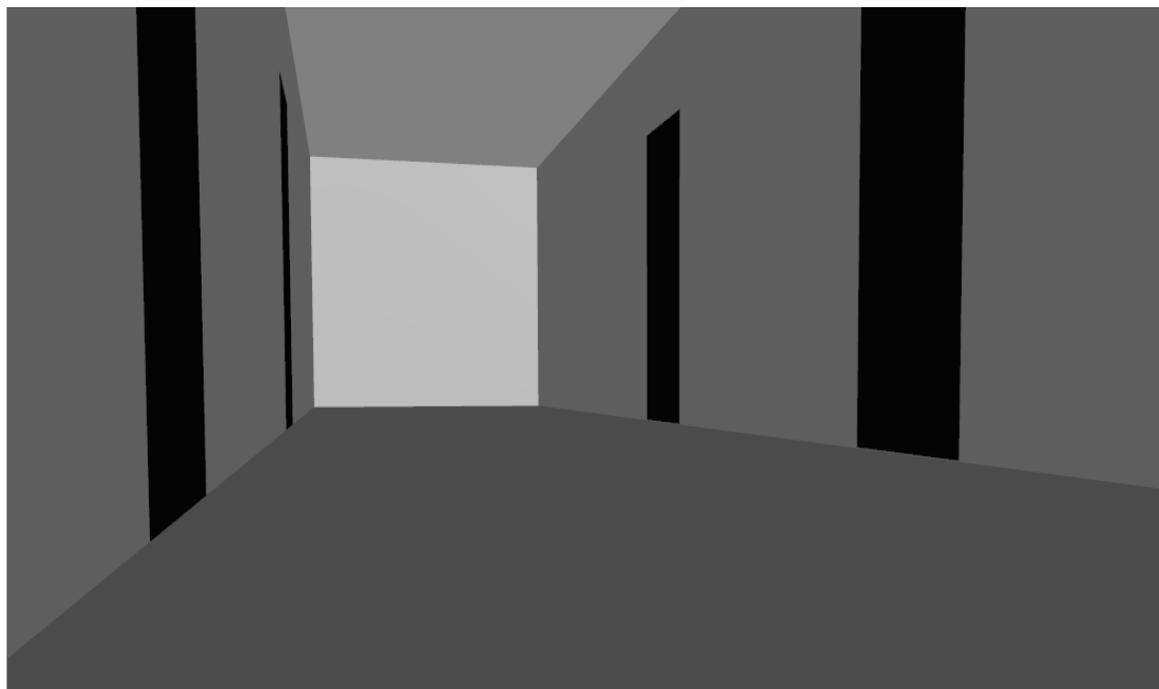


Локализация по имеющимся характерным элементам в окружающей среде

# Полет квадрокоптера, использующего внешние сообщения о локализации



# Оценка близости двух изображений



Модель



Изображение с камеры

# Методы оценки близости изображений

- **По совпадающим пикселям граней (Метод Klein и Murray)**

$$W_t^{(n)} = p(y_t \mid x_t^{(n)}) \propto \exp\left(k \frac{a}{v}\right),$$

где  $a$  – количество выравненных краев,

$v$  – общее количество видимых краев,

$k$  – const для определения функции наблюдения.

- **По совпадающим пикселям каждой грани отдельно**

$$W_t^{(n)} = p(y_t \mid x_t^{(n)}) \propto \exp\left(k \frac{a}{v} + \lambda \frac{\sum_{j=0}^m a_j / v_j}{m}\right),$$

где  $m$  – число краев.

- **Метод ближайшего грани**

# Метод ближайшей грани

- Оценка длины нормали в общем весе прямой по формуле

$$g(d) = \exp\left(-\frac{d^2}{2\sigma^2}\right)$$

$d$  – длина нормали,  $\sigma$  – параметр определяющий вес нормали в зависимости от длины нормали.

- Для каждой прямой вычисляется общий вес формуле

$$l = \frac{\sum_{i=0}^S g(d_i)}{S}$$

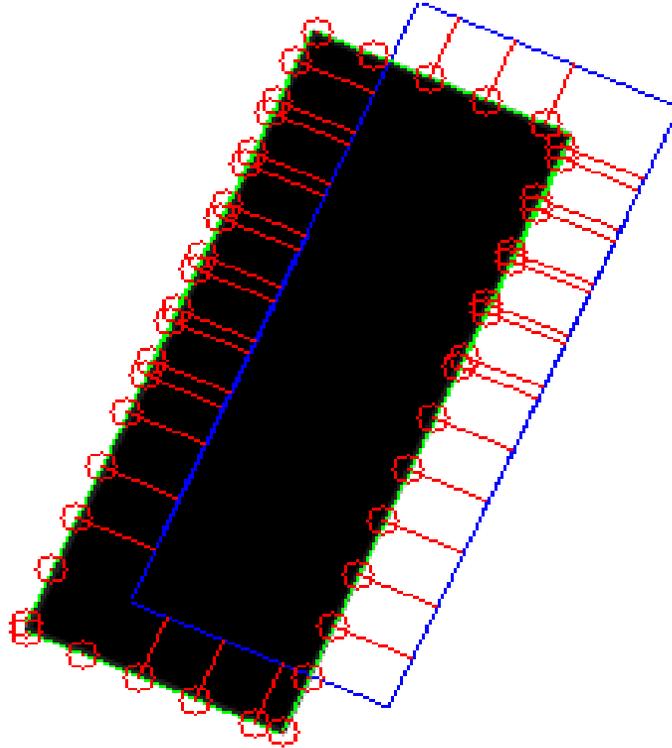
$S$  – общее количество прямых в смоделированном изображении.

- Вычисляется итоговая вероятность гипотезы по формуле

$$W = \alpha \cdot \exp\left(k \frac{\sum_{n=0}^m l_n}{m}\right)$$

$m$  – количество прямых,  $\alpha, k$  – параметры учета веса прямых в итоговом результате.

# Метод ближайшей грани. Пример



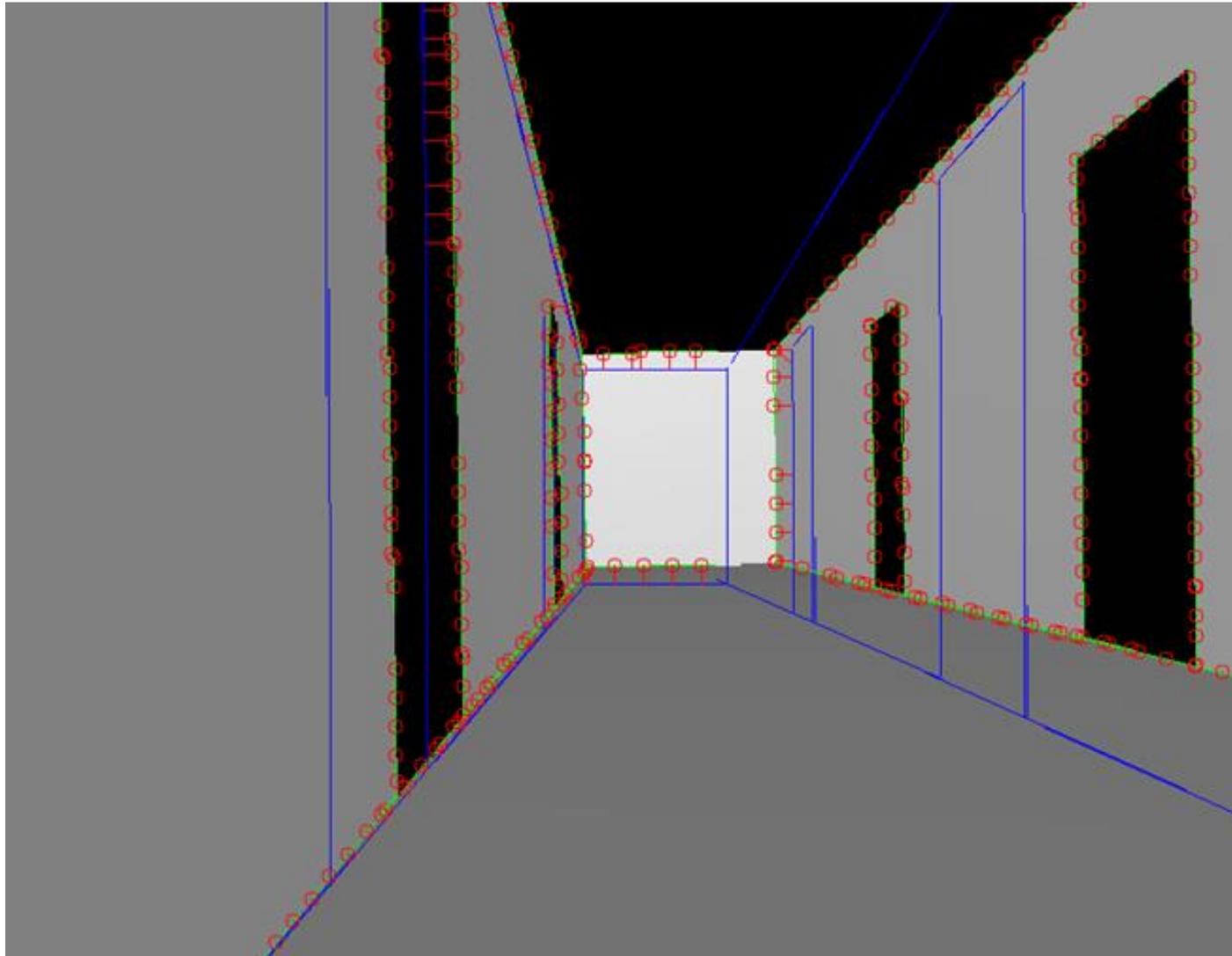
**Зеленым цветом** -  
границы, найденный на  
смоделированном изображении

**Красные окружности** –  
точки для формирования нормалей

**Синим цветом** –  
Прямые с изображения с камеры

**Красные отрезки** –  
Нормали, не превышающие  
Максимальной длины

# Метод ближайшей грани. Пример 2.



**Зеленым цветом** -  
границы, найденный на  
смоделированном изображении

**Красные окружности** –  
точки для формирования нормалей

**Синим цветом** –  
Прямые с изображения с камеры

**Красные отрезки** –  
Нормали, не превышающие  
Максимальной длины

# Particle filter

- Каждая частица – гипотеза о местоположении робота, со следующим вектор состояния:

$$X_t = (x_t, y_t, z_t, \alpha_t, \beta_t, \gamma_t)^T \in R^6$$

- Модель предсказания:

$$X_{t+\Delta t} = X_t + \Delta t (\dot{x}_t, \dot{y}_t, 0, 0, 0, 0)^T$$

- Измерительная функция: близость граней реального изображения и сгенерированного

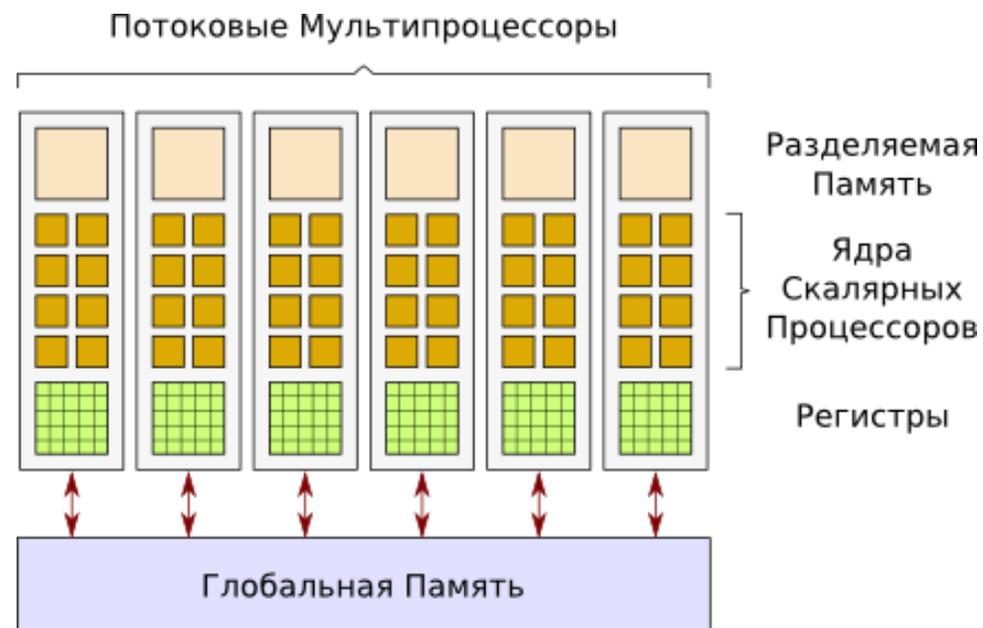
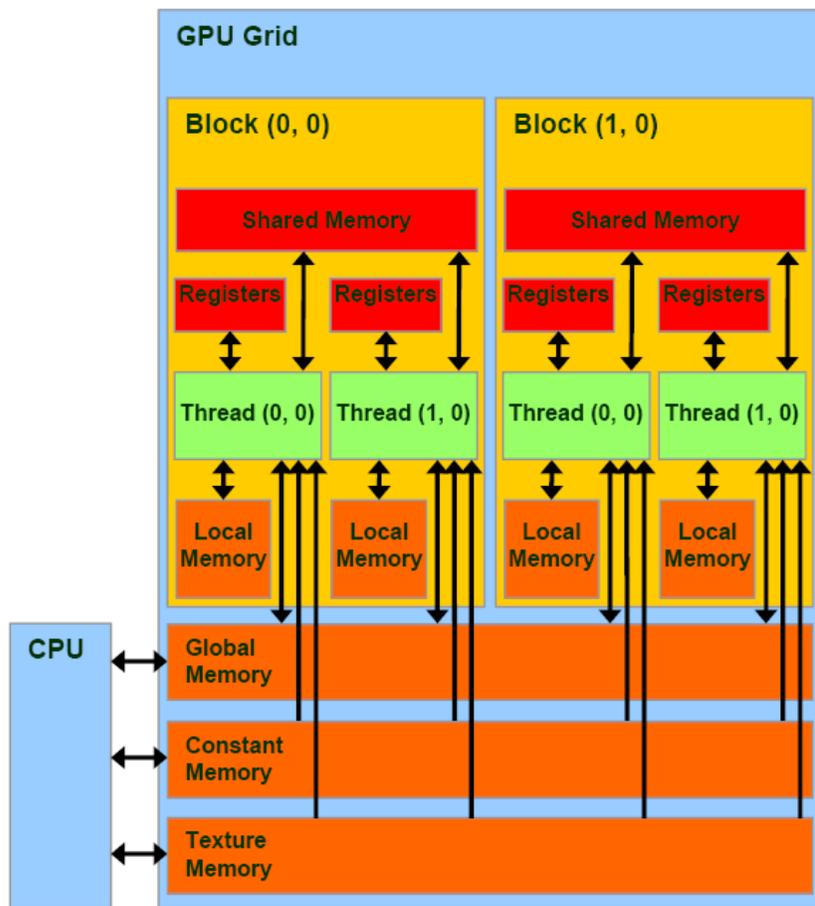
# Основные проблемы производительности

- Время выполнения алгоритма
- Ограничение по размерам и массе компьютера на борту БПЛА
- Потребность в синхронизации реального положения БПЛА в пространстве с его вычисляемыми модельными координатами

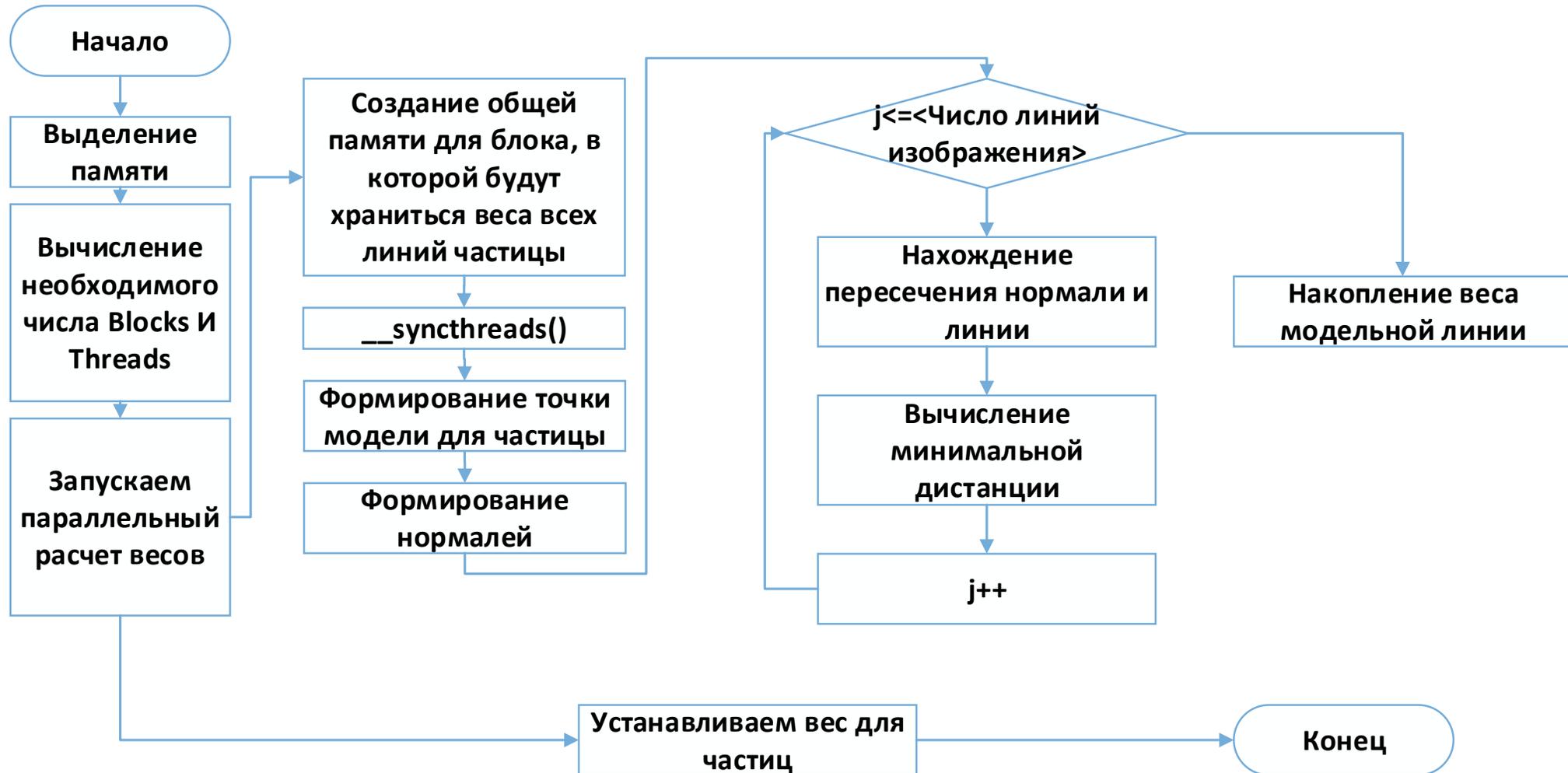
# Параллельные элементы алгоритма

- Вес каждой частицы можно рассчитывать независимо от других
- Рендеринг каждой частицы можно произвести параллельно.
- Каждый рендеринг содержит набор прямых вес каждой из которых можно посчитать параллельно.
- Каждая прямая содержит набор точек, вклад которых в общей вес можно рассчитать параллельно.

# Архитектура CUDA



# GPU реализация алгоритма локализации



SimpleScreenRecorder

tum\_drone GUI

**Send Commands**

**Node Communication Status**  
 Drone Navdata: 0 Hz  
 Drone Control: 5 Hz  
 Pose Estimates: 0 Hz  
 Joy Input: 0 Hz  
 Pings (RTT): 999 (500B), 999 (20kB)  
 Motors: -

**Autopilot Status:**  
 Idle (Queue: 0)  
 Current: NULL  
 Next: NULL  
 Target: (0.00, 0.00, 0.00), 0.0  
 Error: (0.00, 0.00, 0.00), 0.0 (|,| 0.00)  
 Cont.: r 0.00, p 0.00, g 0.00, y 0.00

**Stateestimation Status:**

**Control Source:**  
 Keyboard  Joystick  
 Autopilot  None  
 Use Onboard Hovering  
 Ping Drone (every 1s)

Load File:   
 Clear and Send Clear Send Reset  
 Land Takeoff Emergency Toggle Cam FlatTrim

**Messages**  
 Autopilot: Stop Controlling  
 sent: Takeoff  
 sent: LAND

**TutorialApplication Render Window**

Steps: 1 Real Time Factor: 0.90 Sim Time: 00 00:02:1

**SimpleScreenRecorder**

**Запись**

Начать запись

Включить горячие клавиши  Включить звуковое оповещение

Горячая клавиша:  Ctrl +  Shift +  Alt +  Super + R

**Информация**

Всего прошло: 0:00:00  
 FPS (вход): 0.00  
 FPS (выход): 0.00  
 Размер (вход): 1920x1080  
 Размер (выход): ?  
 Имя файла: 1.mkv  
 Размер файла: 0 B  
 Битрейт: 0 bps

**Предпросмотр**

Битрейт предпросмотра: 10

Примечание: Предпросмотр создаст дополнительную нагрузку на процессор (особенно при высоком битрейте).

**Журнал**

[PageRecord::StartPage] Запускаю...  
 [PageRecord::StartPage] Запущено.

Time  
 ROS Time: 131.47

# Сравнение производительности (200 частиц)

Номер опыта	1	2	3	4	5	6	7
CPU time, мс	123,60	102,83	150,12	131,84	146,74	113,83	163,70
GPU time, мс	7,39	6,54	8,44	7,45	8,46	7,03	8,28
CPU/GPU, раз	16,7	15,7	17,7	17,6	17,3	16,1	19,7