GRAPHICS & MEDIA LAB
VIDEO GROUP

Получение глубины из движения камеры

Зачесов Антон

Video Group
CS MSU Graphics & Media Lab

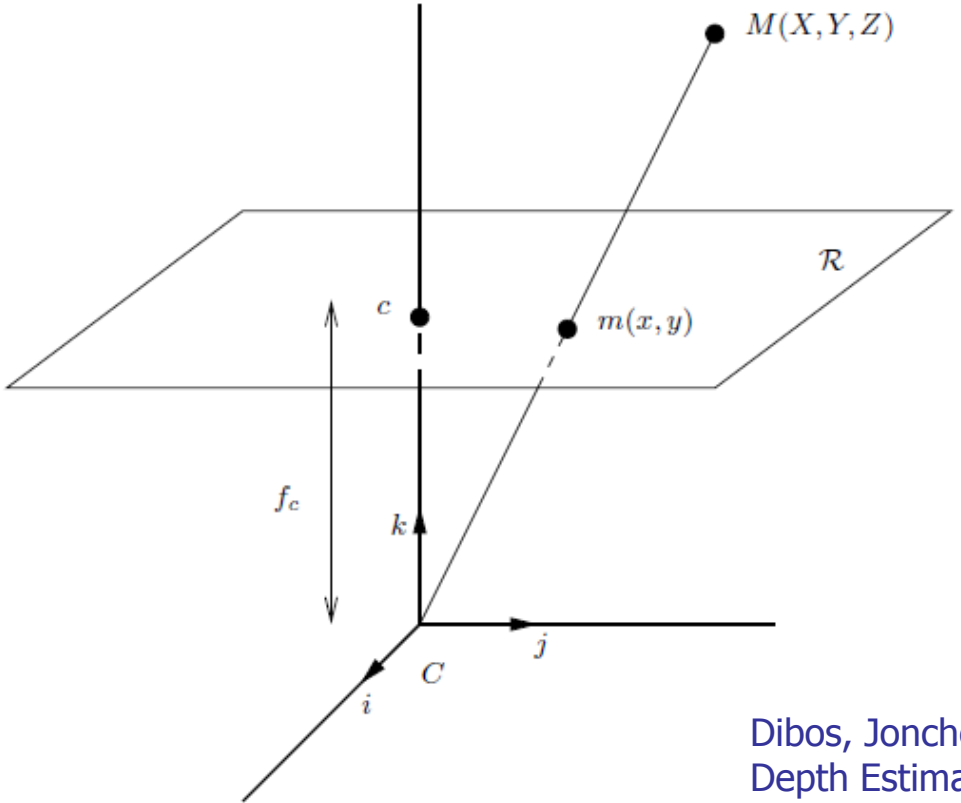
30.03.2015

Содержание

- **Введение**
- Iterative Depth Estimation
- Live Scene Reconstruction
- Real-time Depth Estimation
- Свой алгоритм

Введение
Постановка задачи

Каждая точка пространства при съемке проецируется на плоскость кадра



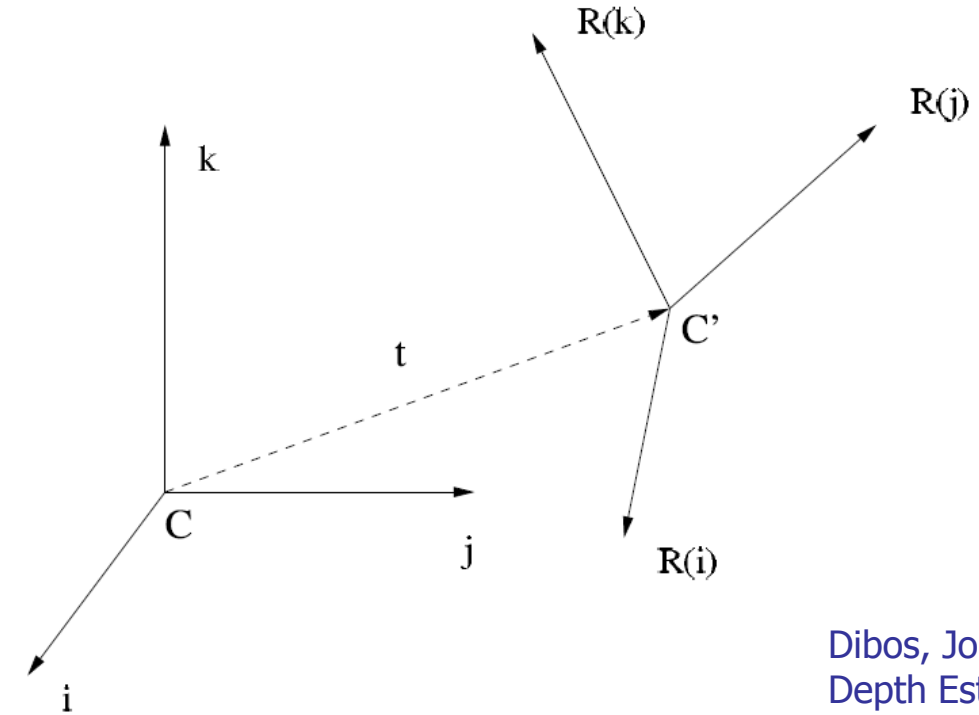
$$x = f_c \frac{X}{Z} \quad y = f_c \frac{Y}{Z}$$

f_c – фокусное расстояние камеры

Dibos, Jonchery, Koepfler, "Iterative Camera Motion and Depth Estimation in a Video Sequence", CAIP, 2009

Введение
Постановка задачи

Зная движение камеры, можно точно восстановить отношение глубины разных объектов в сцене

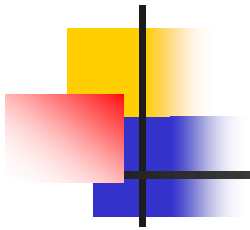


$$R = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \quad t = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}$$

R – матрица поворота

t – вектор сдвига

Dibos, Jonchery, Koepfler, "Iterative Camera Motion and Depth Estimation in a Video Sequence", CAIP, 2009

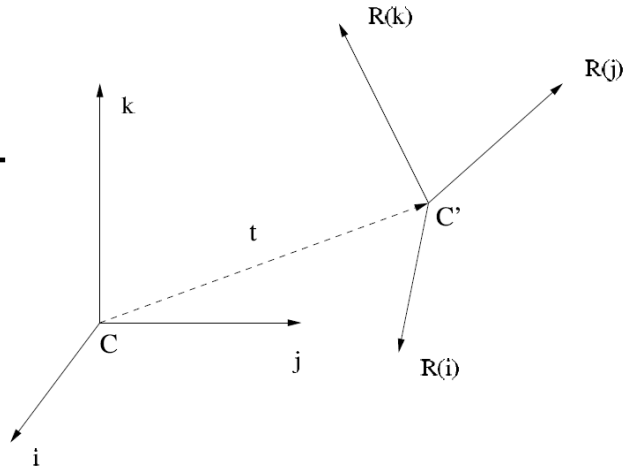


Содержание

- Введение
- **Iterative Depth Estimation**
- Live Scene Reconstruction
- Real-time Depth Estimation
- Свой алгоритм

Iterative Depth Estimation
Camera motion estimation

- Фокусное расстояние камеры $f_c = 1$
- Используется упрощение 12-параметрической модели:



$$\begin{cases} R = \begin{pmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{pmatrix} \\ t = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \end{cases}$$

$$R = R_{\theta,\alpha} R^k_\beta$$

$$t = (A, B, C)$$

R^k_β – поворот вокруг оси k на угол β

$R_{\theta,\alpha}$ – поворот на угол α вокруг оси Δ, угол $(i, \Delta) = \theta$

Iterative Depth Estimation

- Группа из шести параметров $(\theta, \alpha, \beta, A, B, C)$ однозначно описывает движение камеры
- Через эти параметры можно выразить смещение блоков изображений

$$\begin{cases} x' - x \approx -Cx + A - \beta y - \alpha x(y\cos\theta - x\sin\theta) + \alpha\sin\theta \\ y' - y \approx -Cy + B + \beta x - \alpha y(y\cos\theta - x\sin\theta) - \alpha\cos\theta \end{cases}$$

$(x',y'), (x,y)$ – координаты пикселей точки

Dibos, Jonchery, Koepfler, "Iterative Camera Motion and Depth Estimation in a Video Sequence", CAIP, 2009

Iterative Depth Estimation
Идея

Первый кадр

Iterative Depth Estimation
Идея

Второй кадр

Iterative Depth Estimation
Идея

GRAPHICS & MEDIA LAB
VIDEO GROUP

Искаженный первый кадр

Iterative Depth Estimation
Идея

Первый кадр

Iterative Depth Estimation
Идея

Модули смещений блоков между исходным и искаженным кадрами

Iterative Depth Estimation

- Для восстановления глубины сцены применяется belief propagation (100 итераций)
- Максимизируется вероятность

$$P(\mathbf{x}) = P(X = \mathbf{x} \,|Y = y)$$

$$y = \{f, g, \theta, \alpha, \beta, t_1, t_2, t_3\}$$

f, g – кадры

y – реализация random field Y

x – глубина сцены по отношению к кадру f

Dibos, Jonchery, Koepfler, "Iterative Camera Motion and Depth Estimation in a Video Sequence", CAIP, 2009

Iterative Depth Estimation

$$P(\mathbf{x}) \propto P(Y = y|X = \mathbf{x})\, P(X = \mathbf{x})$$

$$P(Y = y|X = \mathbf{x}) = \prod_{s \in V} \psi(\mathbf{x}_s)$$

V – множество всех пикселей

x_s – относительная глубина в пикселе s

$\psi(\mathrm{x}_s)$ – вероятность определения глубины для пикселя s

$$\psi(\mathbf{x}_s) = (1 - e_2)e^{-\frac{|f(s)-g(s')|}{\sigma_2}} + e_2$$

$s' = (x', y')$ – координаты пикселя (x,y) после применения к нему преобразования $\{\theta, \alpha, \beta, t_1, t_2, t_3\}$

Iterative Depth Estimation

$$P(\mathbf{x}) \propto P(Y = y | X = \mathbf{x})\, P(X = \mathbf{x})$$

$$P(X = \mathbf{x}) = \prod_{(s,t) \in E} \psi_{st}(\mathbf{x}_s, \mathbf{x}_t)$$

E – множество границ

$\psi_{st}(\mathrm{x}_s, \mathrm{x}_t)$ – отвечает за непрерывность глубины для соседних пикселей s и t

$$\psi_{st}(\mathbf{x}_s, \mathbf{x}_t) = (1 - e_1) e^{-\frac{|\mathbf{x}_s - \mathbf{x}_t|}{\sigma_1}} + e_1$$

e_1 , σ_1 $= const$

Dibos, Jonchery, Koepfler, "Iterative Camera Motion and Depth Estimation in a Video Sequence", CAIP, 2009

Iterative Depth Estimation
Метод уточнения глубины

- Идея – попеременно искать и уточнять параметры движения камеры и карту глубины
- Глубина сцены делится на H уровней $\Lambda = I_1 \cup I_2 \cup \cdots \cup I_H$
- На каждом шаге i:
  - Для каждой I_h считается смещение с глубиной $\hat{X}_h$
  - Для каждой I_h считается вес вклада в общее движение в кадре $p_h = \#\{\mathbf{X}_{i-1} \in I_h\}/\#K$
  - По полученным смещениям каждой области получаем новые параметры камеры $\varphi_i = \{\theta, \alpha, \beta, t_1, t_2, t_3\}$
  - Если $\|f \circ \varphi_i - g\|_1 < \|f \circ \tilde{\varphi} - g\|_1$, то $\tilde{\varphi} = \varphi_i$

Dibos, Jonchery, Koepfler, "Iterative Camera Motion and Depth Estimation in a Video Sequence", CAIP, 2009

Iterative Depth Estimation
Исходный кадр 1

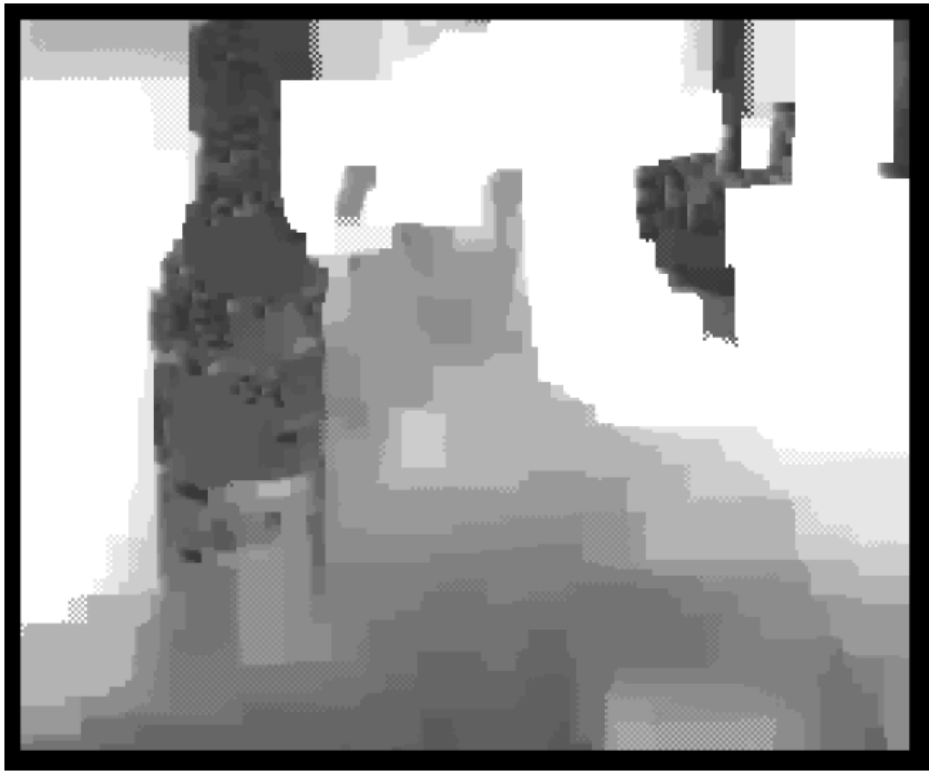
Dibos, Jonchery, Koepfler, "Iterative Camera Motion and Depth Estimation in a Video Sequence", CAIP, 2009

Iterative Depth Estimation

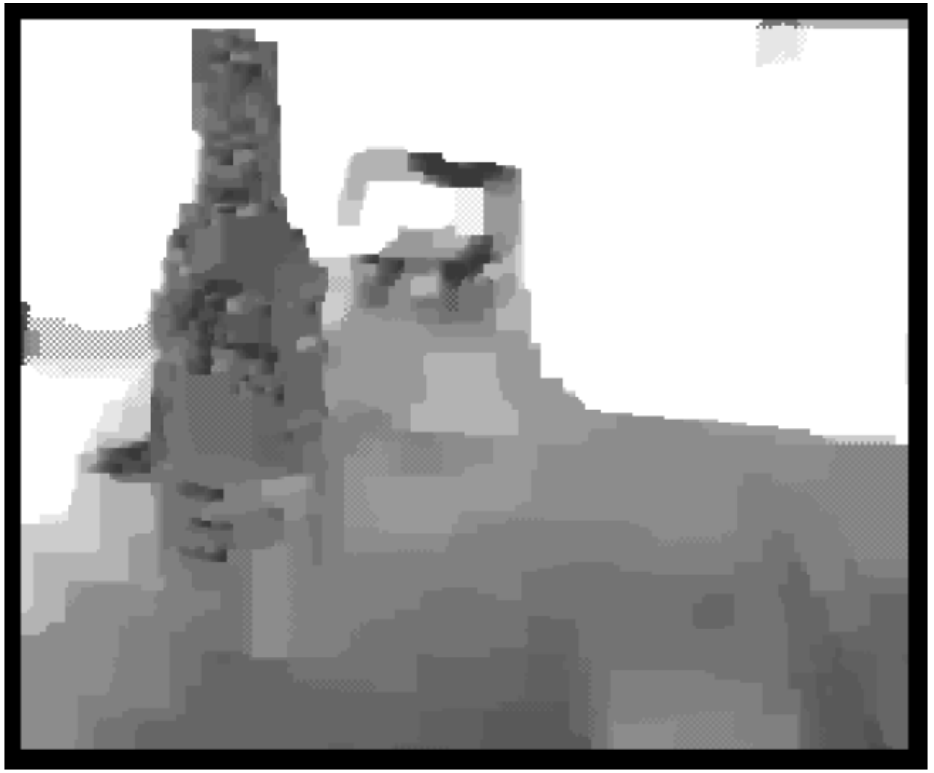
Исходный кадр 2

Dibos, Jonchery, Koepfler, "Iterative Camera Motion and Depth Estimation in a Video Sequence", CAIP, 2009

Iterative Depth Estimation
Результаты



100 итераций belief propagation



15 итераций алгоритма уточнения

Dibos, Jonchery, Koepfler, "Iterative Camera Motion and Depth Estimation in a Video Sequence", CAIP, 2009

Iterative Depth Estimation

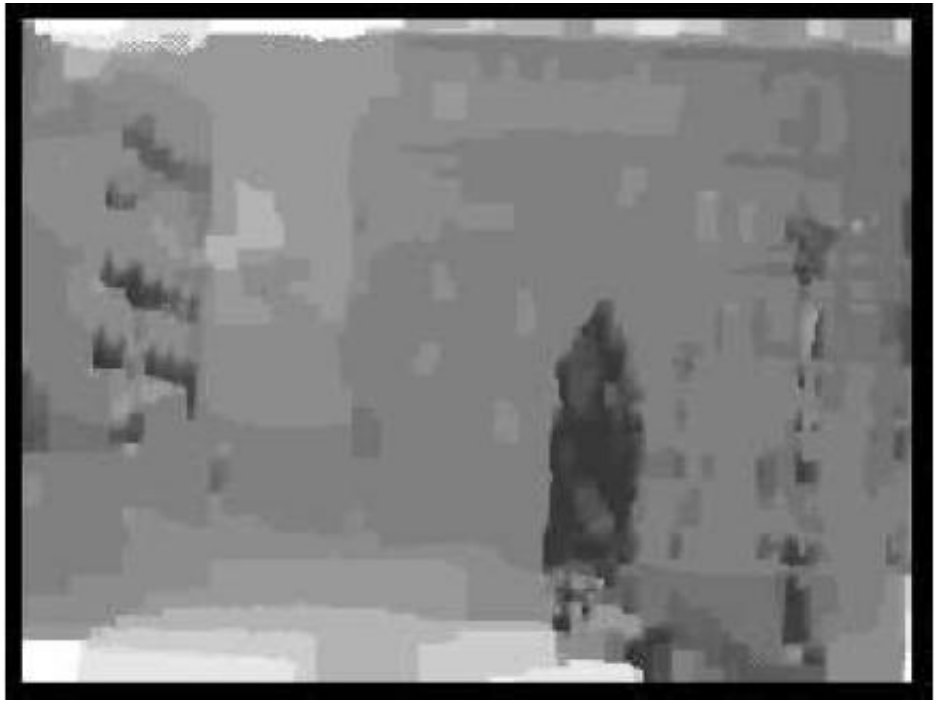
Исходный кадр 1

Dibos, Jonchery, Koepfler, "Iterative Camera Motion and Depth Estimation in a Video Sequence", CAIP, 2009

Iterative Depth Estimation

Исходный кадр 2

Iterative Depth Estimation
Результаты



100 итераций belief propagation

15 итераций алгоритма уточнения

Dibos, Jonchery, Koepfler, "Iterative Camera Motion and Depth Estimation in a Video Sequence", CAIP, 2009

Iterative Depth Estimation
Выводы

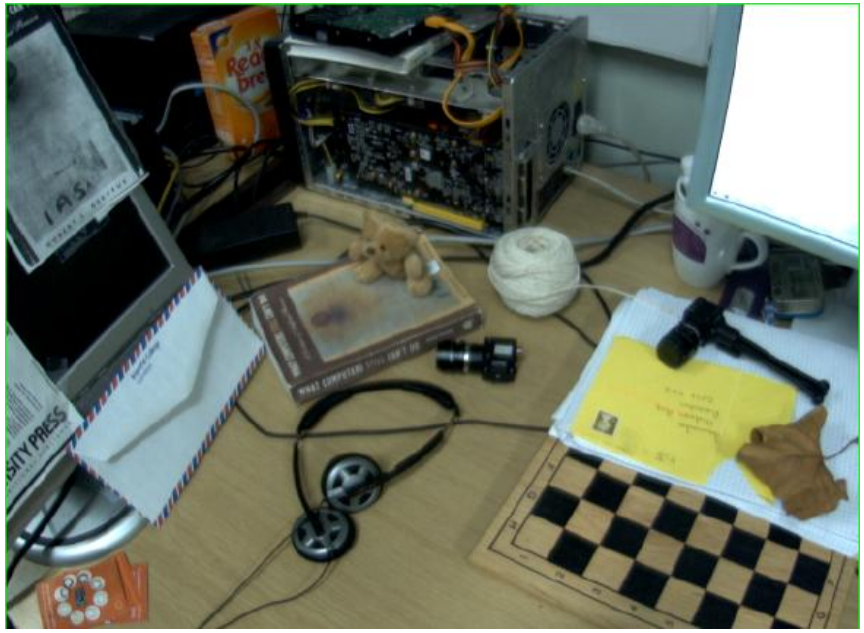
- Достоинства
  - Надежный алгоритм вычисления параметров движения камеры
  - Эффективный метод расчета глубины
- Недостатки
  - Вычислительная сложность
  - Изначальное деление глубины на фиксированное число уровней

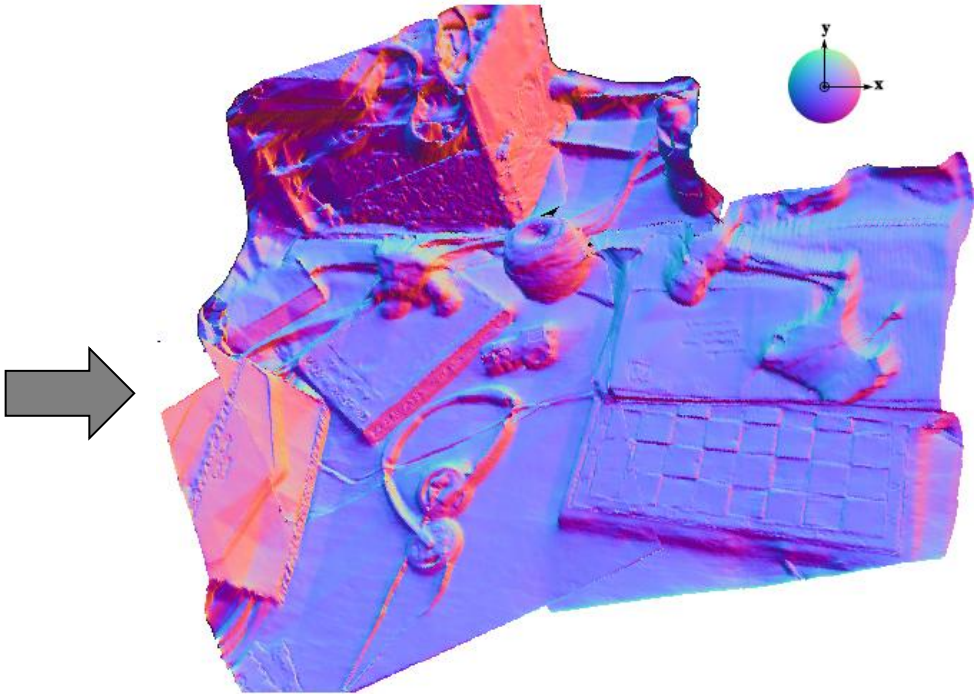
Содержание

- Введение
- Iterative Depth Estimation
- **Live Scene Reconstruction**
- Real-time Depth Estimation
- Свой алгоритм

Live Scene Reconstruction
Постановка задачи

По видео, снятому 2D-камерой, точно восстановить глубину и построить достоверную модель сцены



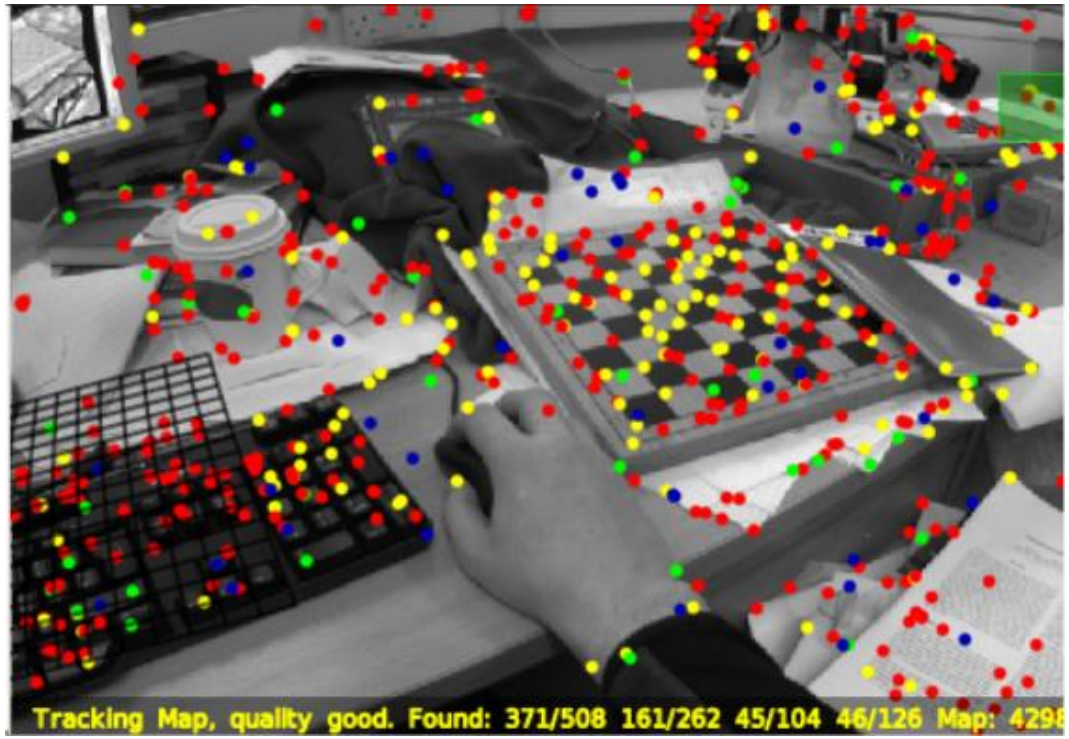


Newcombe, Davison, "Live Dense Reconstruction with a Single Moving Camera", CVPR, 2010

Live Scene Reconstruction
Camera motion estimation

GRAPHICS & MEDIA LAB
VIDEO GROUP

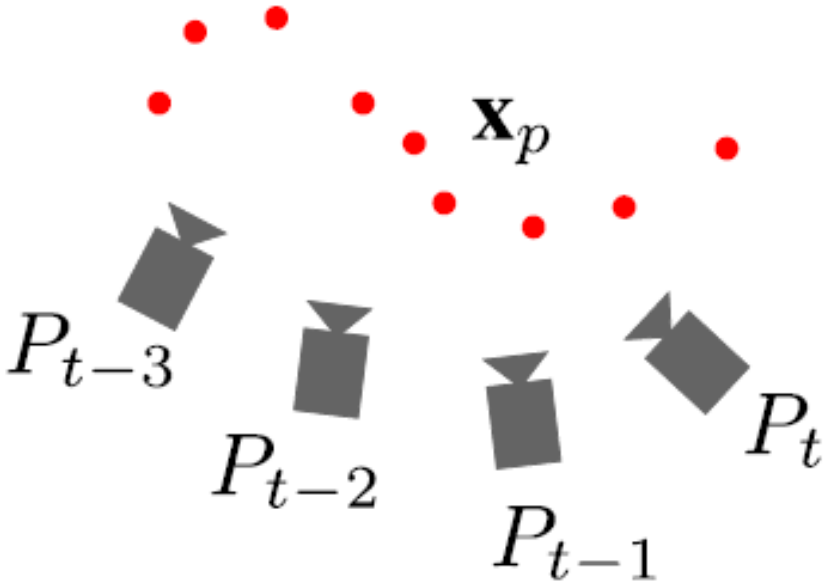
Анализ движения:
Parallel Tracking and Mapping system (PTAM)



Newcombe, Davison, "Live Dense Reconstruction with a Single Moving Camera", CVPR, 2010

Live Scene Reconstruction
Camera motion estimation

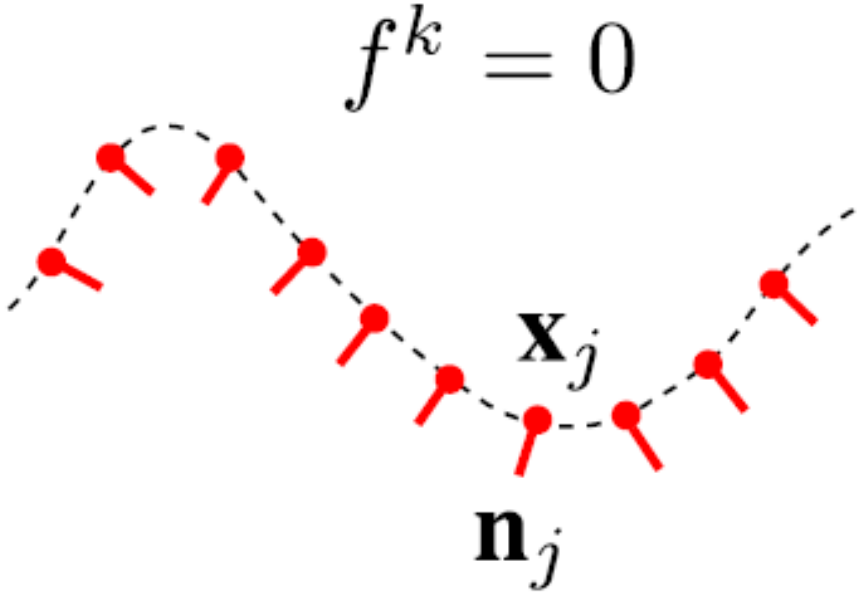
- PTAM позволяет расставлять в видео ключевые кадры
- Для каждого ключевого кадра определяется позиция камеры и видимость каждой точки в нем



Newcombe, Davison, "Live Dense Reconstruction with a Single Moving Camera", CVPR, 2010

Live Scene Reconstruction
Инициализация нормалей

В каждой точке x_j нормаль n_j принимается равной:
$n_j = 1/N \times \Sigma(V(P_T))$
$V(P_T)$ – направление на камеру в ключевом кадре T
N – число точек



Newcombe, Davison, "Live Dense Reconstruction with a Single Moving Camera", CVPR, 2010

Live Scene Reconstruction
Depth estimation

Для каждого ключевого кадра считается яркость $\mathbf{I}^{Ref}$
и матрица проекции точек пространства
на плоскость этого кадра

$$P^{Ref} = \mathrm{K}\left[\mathbf{R}_{cw}^{ref\,\mathrm{T}} \mid -\mathbf{R}_{cw}^{ref\,\mathrm{T}}\mathbf{t}_{cw}^{ref}\right]$$

K – матрица, получаемая при калибровке камеры
$\mathbf{R}_{cw}^{ref}$ – матрица поворота из системы координат камеры
в мировую систему координат

Live Scene Reconstruction
Depth estimation

- Каждой точке $\mathbf{x}_j = [x_j, y_j, z_j]^{\mathrm{T}}$ ставится в соответствие проекция $[u_j^i, v_j^i]^{\mathrm{T}} = \mathbf{u}_j^i = \mathbf{P}^i(\mathbf{x}_j)$ для каждой камеры i
- При малом смещении точки на расстояние $\Delta\mathbf{x}_j$ $\Delta\mathbf{u}_j^i = \mathbf{u}_j'^i - \mathbf{u}_j^i$ может быть выражено формулой

$$\Delta\mathbf{u}_j^i = \mathbf{J}_{\mathbf{x}_j}^i \Delta\mathbf{x}_j$$

$$\mathbf{J}_{\mathbf{x}_j}^i \equiv \left.\frac{\partial \mathbf{P}^i}{\partial \mathbf{x}}\right|_{\mathbf{x}_j} \equiv \left.\begin{bmatrix} \frac{\partial \mathbf{P}_u^i}{\partial x} & \frac{\partial \mathbf{P}_u^i}{\partial y} & \frac{\partial \mathbf{P}_u^i}{\partial z} \\ \frac{\partial \mathbf{P}_v^i}{\partial x} & \frac{\partial \mathbf{P}_v^i}{\partial y} & \frac{\partial \mathbf{P}_v^i}{\partial z} \end{bmatrix}\right|_{\mathbf{x}_j}$$

Newcombe, Davison, "Live Dense Reconstruction with a Single Moving Camera", CVPR, 2010

Live Scene Reconstruction
Depth estimation

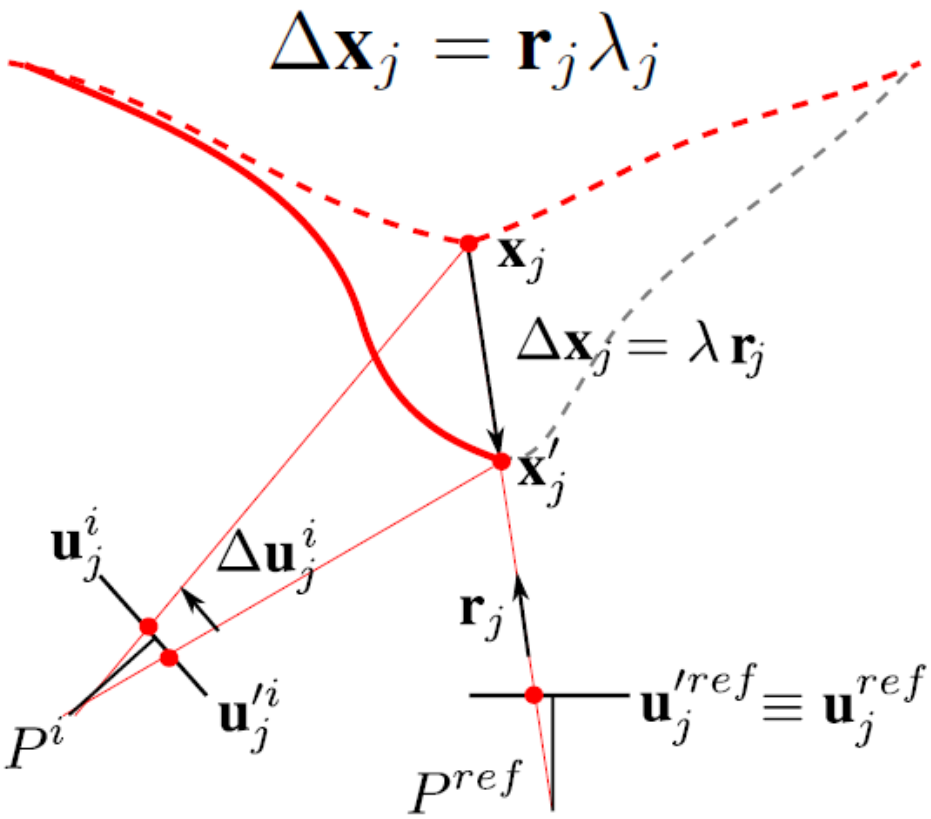
- Якобиан рассчитывается для матрицы проекции каждого ключевого кадра
- Из полученных линейных систем можно вычислить $\Delta \mathbf{x}_j$ для каждого пикселя

$$\Delta \mathbf{u}_j^i = \mathbf{J}_{\mathbf{x}_j}^i \Delta \mathbf{x}_j$$

$$\mathbf{J}_{\mathbf{x}_j}^i \equiv \left. \frac{\partial \mathbf{P}^i}{\partial \mathbf{x}} \right|_{\mathbf{x}_j} \equiv \left. \begin{bmatrix} \frac{\partial \mathbf{P}_u^i}{\partial x} & \frac{\partial \mathbf{P}_u^i}{\partial y} & \frac{\partial \mathbf{P}_u^i}{\partial z} \\ \frac{\partial \mathbf{P}_v^i}{\partial x} & \frac{\partial \mathbf{P}_v^i}{\partial y} & \frac{\partial \mathbf{P}_v^i}{\partial z} \end{bmatrix} \right|_{\mathbf{x}_j}$$

Newcombe, Davison, "Live Dense Reconstruction with a Single Moving Camera", CVPR, 2010

Live Scene Reconstruction
Уточнение глубины



- $\mathbf{r}_j$ – луч из позиции камеры к пикселю j
- λ_j – параметр для каждой вершины
- P^{ref} – ключевой кадр
- P^i – один из соседних кадров

Newcombe, Davison, "Live Dense Reconstruction with a Single Moving Camera", CVPR, 2010

Live Scene Reconstruction
Уточнение глубины

$$\begin{bmatrix} r_j^x \\ r_j^y \\ r_j^z \end{bmatrix} \equiv \mathbf{r}_j = \frac{\mathbf{R}_{cw}^{ref}[\mathbf{u}_j^{ref\,\mathrm{T}}|1]}{\|\mathbf{R}_{cw}^{ref}[\mathbf{u}_j^{ref\,\mathrm{T}}|1]\|}$$

$\mathbf{R}_{cw}^{ref}$ – матрица поворота из системы координат камеры в мировую систему координат

$\mathbf{K}_j^i$ – скалярное произведение

$$\Delta\mathbf{u}_j^i = \mathbf{K}_j^i \lambda_j \qquad \mathbf{K}_j^i \equiv \mathbf{J}_{\mathbf{x}_j}^i \mathbf{r}_j$$

$$\begin{bmatrix} u_j^1 \\ v_j^1 \\ \vdots \\ u_j^{2n} \\ v_j^{2n} \end{bmatrix} \equiv \Delta\mathbf{u}_j \qquad \begin{bmatrix} K[u]_j^1 \\ K[v]_j^1 \\ \vdots \\ K[u]_j^{2n} \\ K[v]_j^{2n} \end{bmatrix} \equiv \mathbf{K}_j$$

Newcombe, Davison, "Live Dense Reconstruction with a Single Moving Camera", CVPR, 2010

Live Scene Reconstruction
Уточнение глубины

GRAPHICS & MEDIA LAB
VIDEO GROUP

- Система сводится к поиску $\min_{\lambda_j} \|\mathbf{K}_j \lambda_j - \Delta \mathbf{u}_j\|$
- Решив систему, получаем параметр λ:

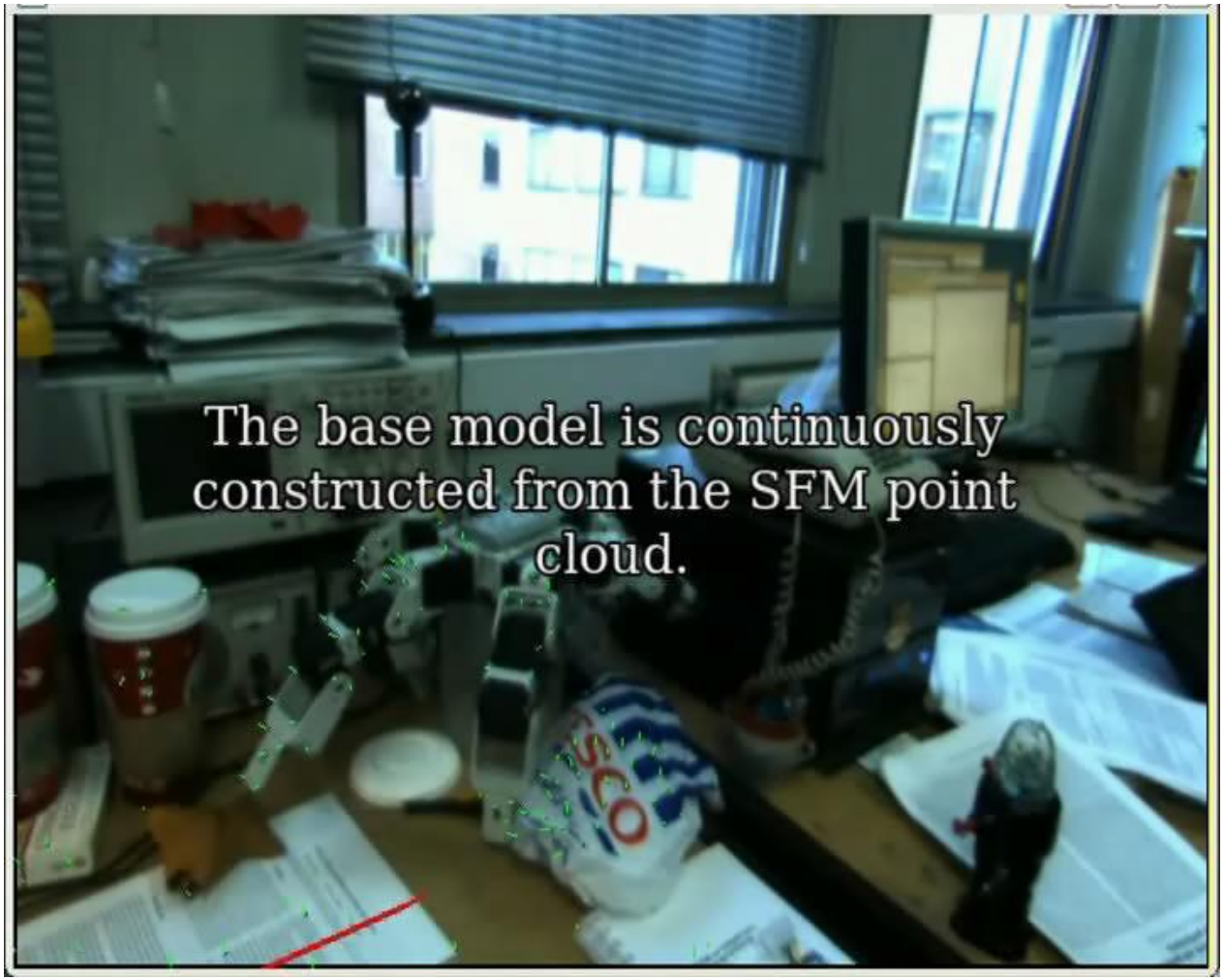
$$\lambda_j = (\mathbf{K}_j^{\mathrm{T}} \mathbf{K}_j)^{-1} \mathbf{K}_j \Delta \mathbf{u}_j \equiv \frac{\sum_{a=1}^{2n} k_a u_a}{\sum_{a=1}^{2n} k_a^2}$$

- Получаем точные координаты вершины:

$$\mathbf{x}'_j = \mathbf{x}_j + \mathbf{r}_j \lambda_j$$

Newcombe, Davison, "Live Dense Reconstruction with a Single Moving Camera", CVPR, 2010

Live Scene Reconstruction
Результаты

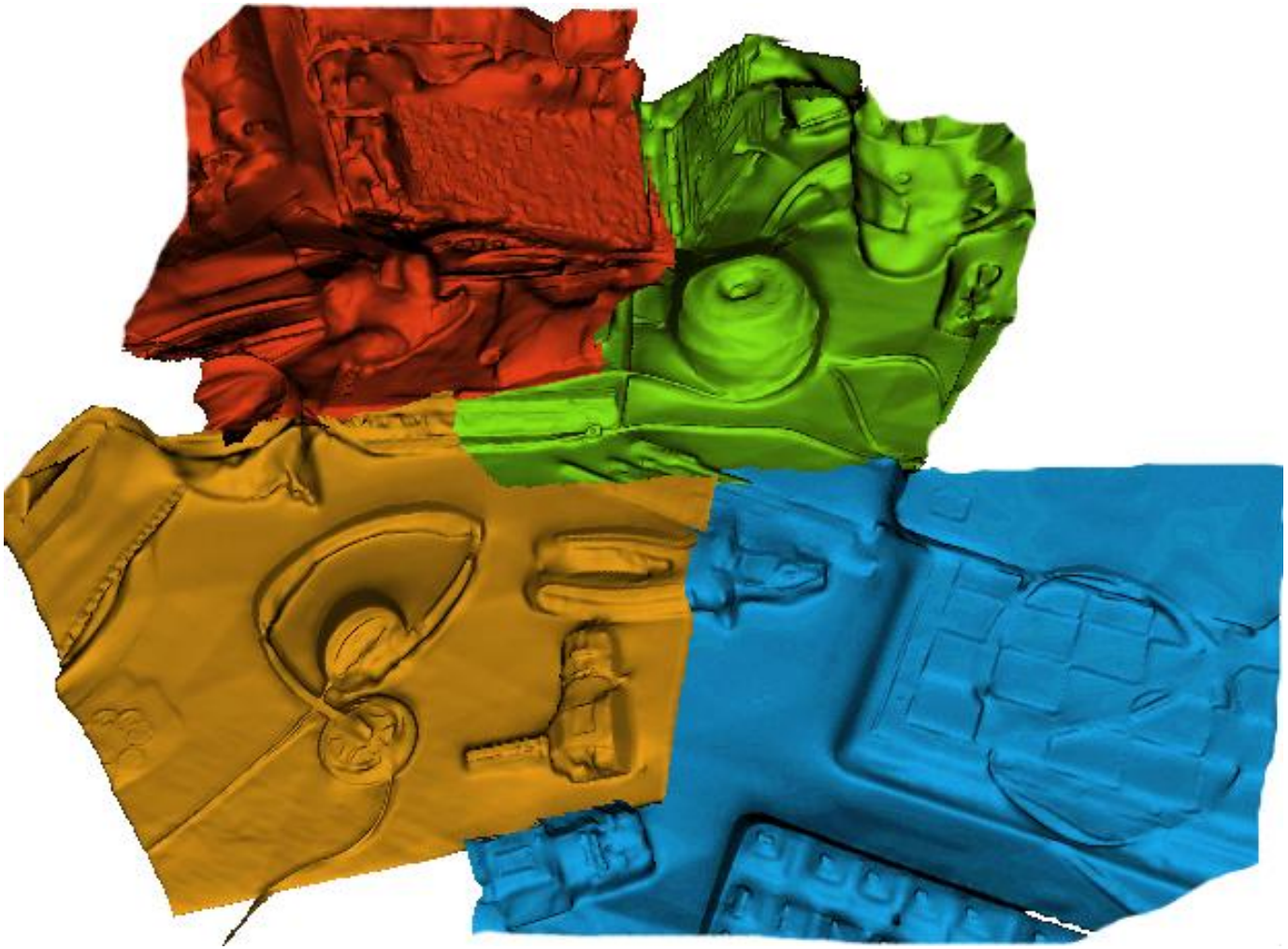


Newcombe, Davison, "Live Dense Reconstruction with a Single Moving Camera", CVPR, 2010

Live Scene Reconstruction
Результаты

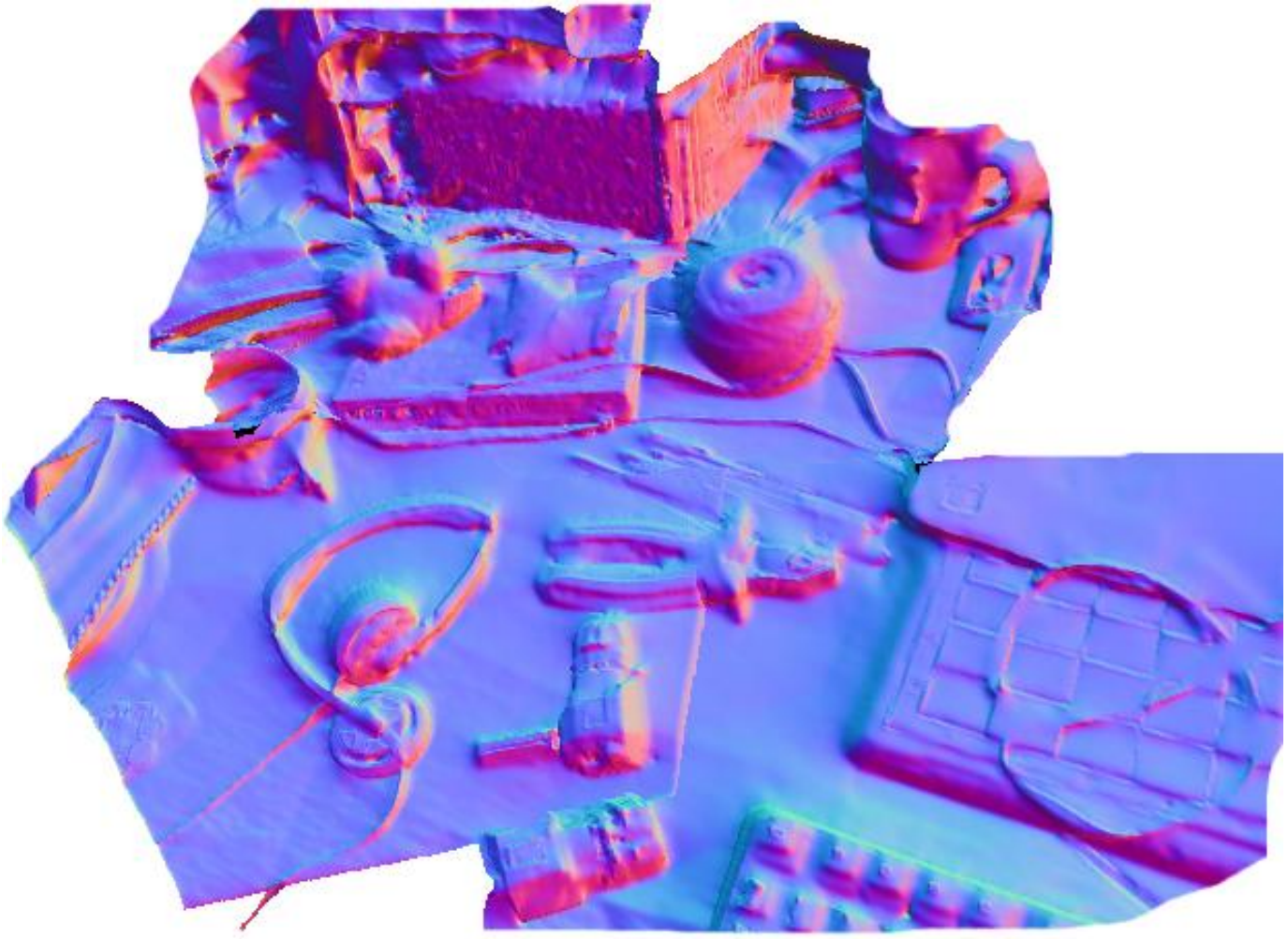
Newcombe, Davison, "Live Dense Reconstruction with a Single Moving Camera", CVPR, 2010

Live Scene Reconstruction
Результаты



Newcombe, Davison, "Live Dense Reconstruction with a Single Moving Camera", CVPR, 2010

Live Scene Reconstruction
Результаты



Newcombe, Davison, "Live Dense Reconstruction with a Single Moving Camera", CVPR, 2010

Live Scene Reconstruction
Итоги

- Достоинства
  - Скорость (засчет использования GPU)
    - Модель сцены строится в реальном времени
    - На QuadCore Xeon+2GPU – обновление модели каждые 2-5 секунд
  - Качество и точность получаемой карты глубины
- Недостатки
  - Необходимость калибровки камеры перед началом работы алгоритма

Содержание

- Введение
- Iterative Depth Estimation
- Live Scene Reconstruction
- **Real-time Depth Estimation**
- Свой алгоритм

Real-time Depth Estimation

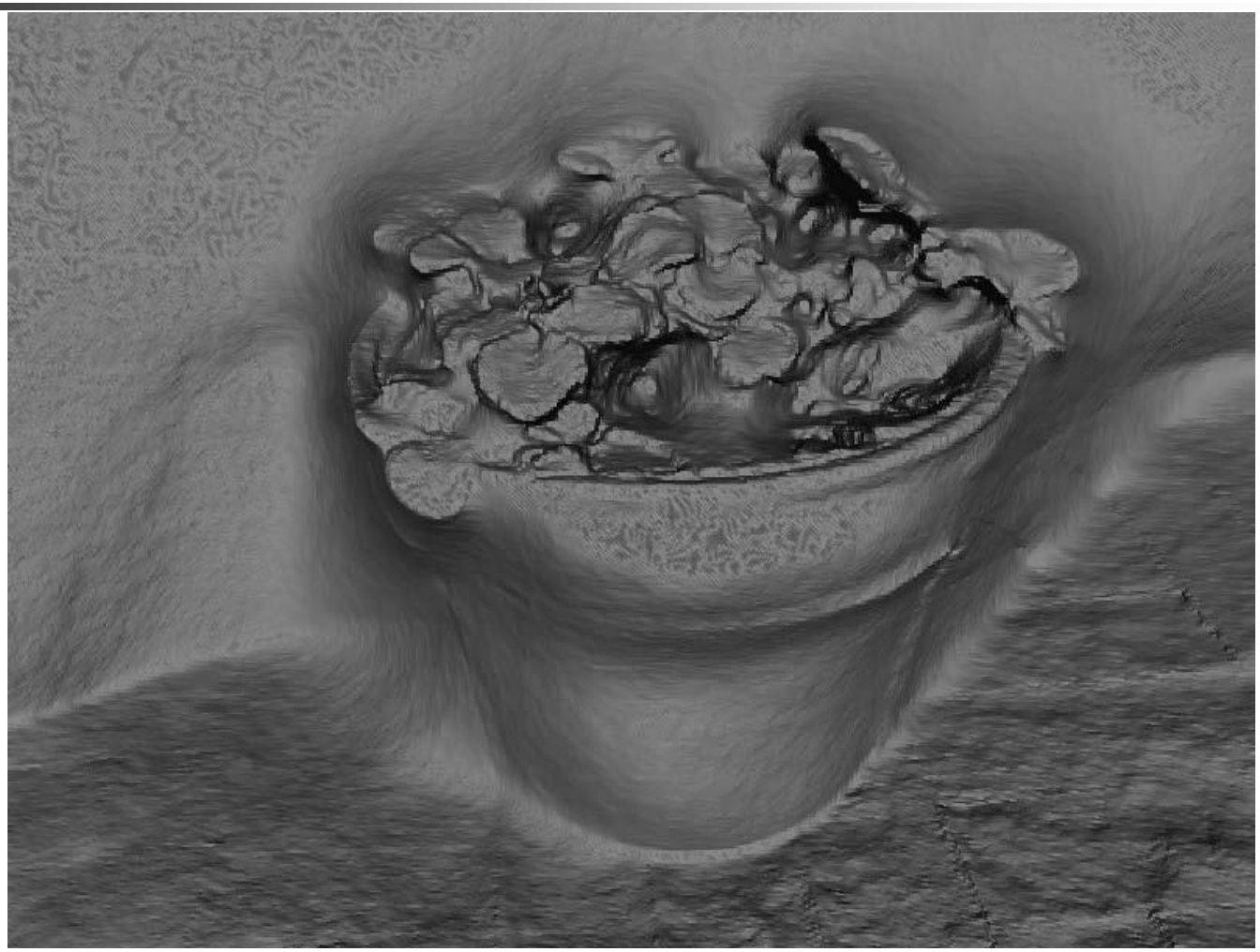
- Та же задача реконструкции сцены по движению одной камеры
- Оптимизация – нет расчета 2D-motion векторов для каждой точки кадра

Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

Real-time Depth Estimation
Пример

Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

Real-time Depth Estimation
Пример



Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

Real-time Depth Estimation
Пример

Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

Real-time Depth Estimation
2 изображения

Для простого случая пары изображений минимизируется энергия вида:

$$E(h) = \lambda \int_{\Omega_0} \left| I_1\left(\pi\left(\exp(\hat{\xi}_1)\,\mathbf{X}(\mathbf{x}, h)\right)\right) - I_0\left(\pi(\mathbf{x})\right) \right| d^2\mathbf{x} + \int_{\Omega_0} |\nabla h| \; d^2\mathbf{x}$$

I_0, I_1 – изображения

$exp(\xi_1)$ – 6 параметров камеры

π – проекция точки из мировых координат на плоскость кадра

Ω_0 – множество всех возможных глубин сцены

h – глубина

Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

Real-time Depth Estimation
2 изображения

$$E(h) = \lambda \int_{\Omega_0} \left| I_1\left(\pi\left(\exp(\hat{\xi}_1)\,\mathbf{X}(\mathbf{x}, h)\right)\right) - I_0\left(\pi(\mathbf{x})\right) \right| d^2\mathbf{x} + \int_{\Omega_0} |\nabla h|\, d^2\mathbf{x}$$

Разница между двумя изображениями, приведенными к одной системе координат

Позволяет избежать разрывов глубины на границах объектов

Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

Real-time Depth Estimation
N изображений

$$E(h) = \lambda \int_{\Omega} \sum_{i \in \mathcal{I}(\mathbf{x})} |\rho_i(\mathbf{x}, h)| \, d^2\mathbf{x} + \int_{\Omega} |\nabla h| \, d^2\mathbf{x}$$

Для n последовательных изображений можно суммировать энергии каждой из пар

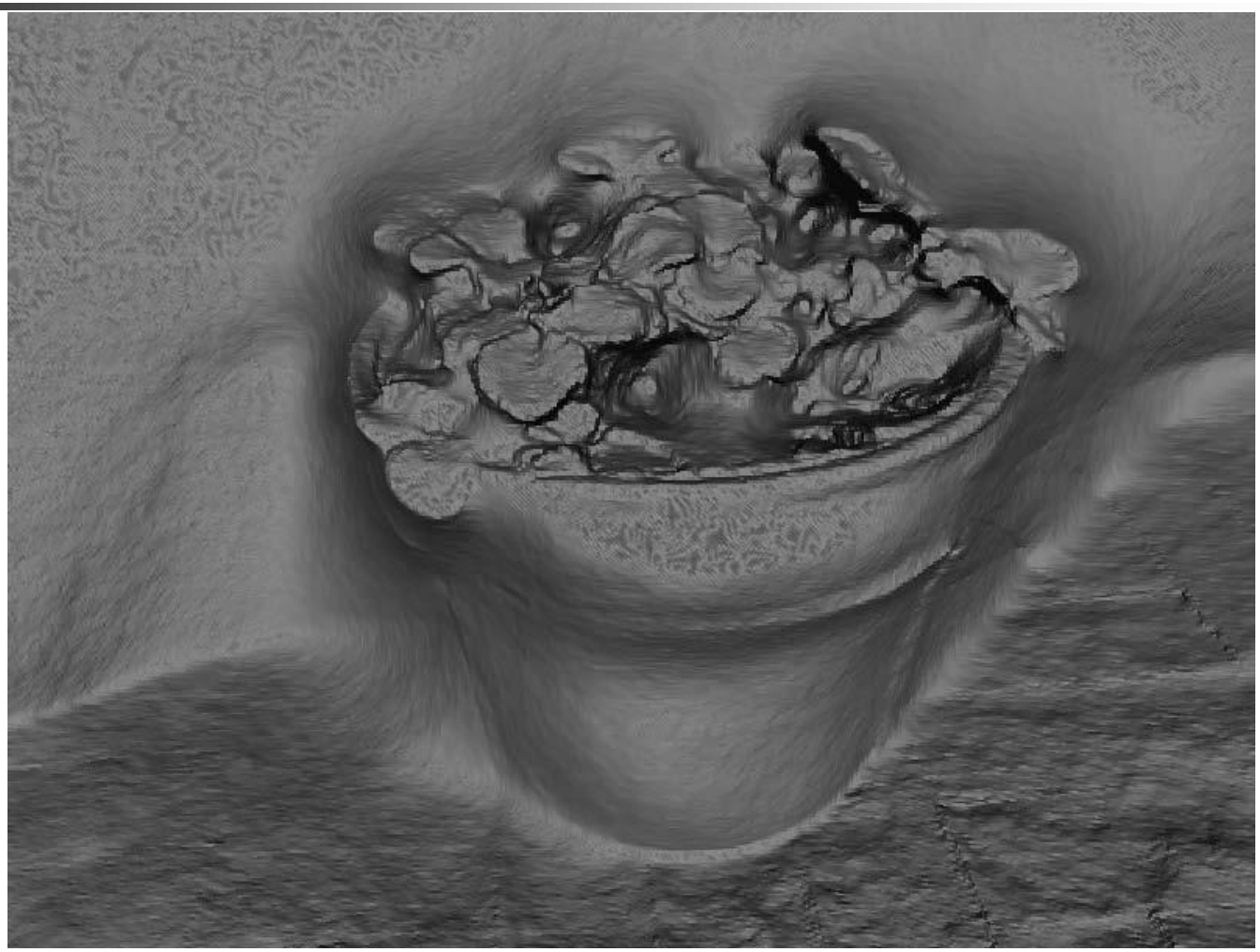
$$\rho_i(\mathbf{x}, h) = I_i(\mathbf{x}, h_0) + (h - h_0) \, I_i^h(\mathbf{x}) - I_0(\mathbf{x})$$

$I_i^h(\mathbf{x})$ – производная по h в точке h_0

$\mathcal{I}(\mathbf{x})$ – множество кадров, для которых проекция точки x находится внутри кадра

Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

Real-time Depth Estimation
Результат (без текстуры)



Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

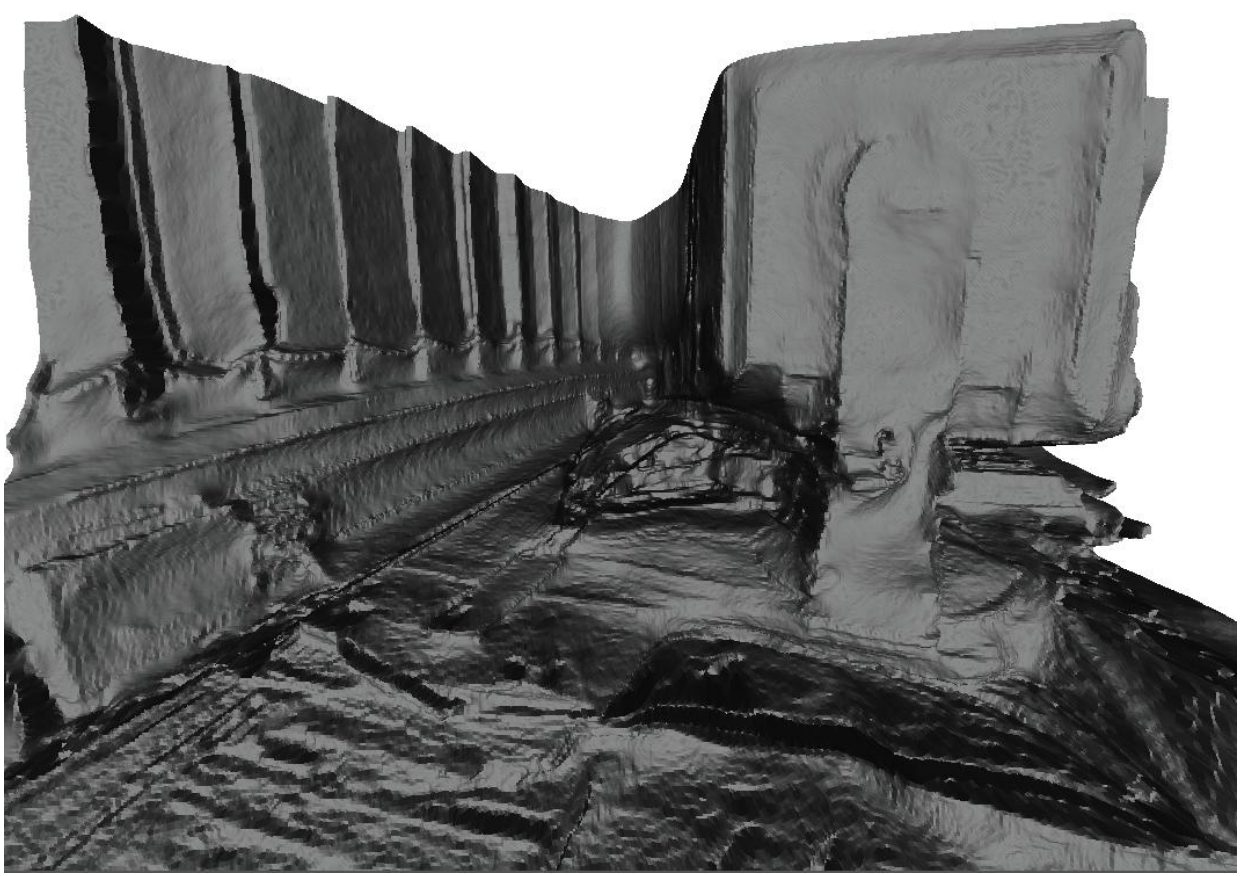
Real-time Depth Estimation
Результат (с текстурой)

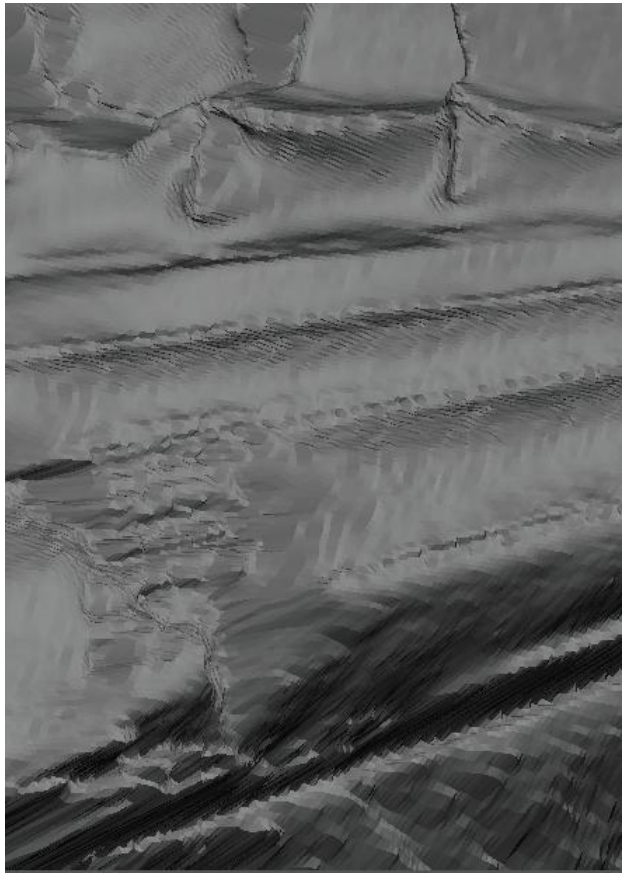
Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

Real-time Depth Estimation
Исходные данные

Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

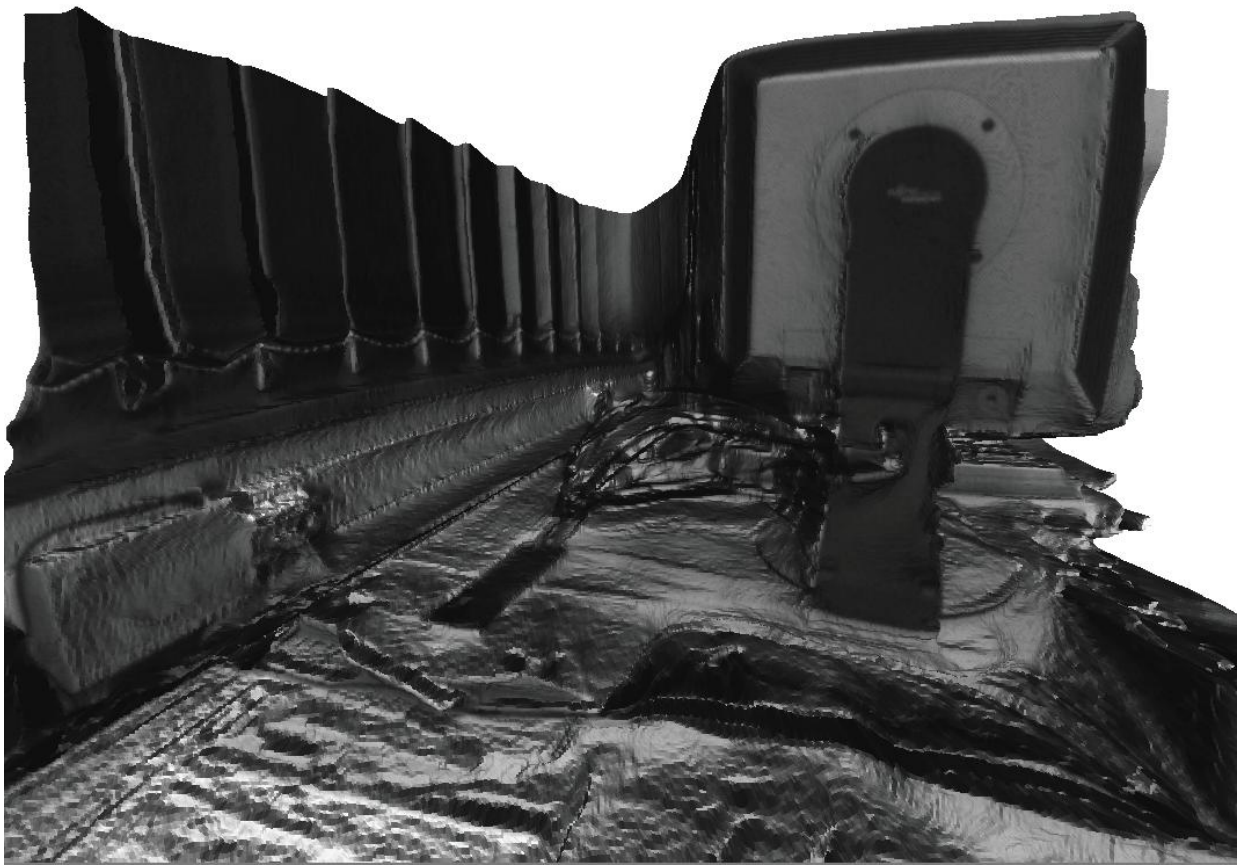
Real-time Depth Estimation
Результат (без текстуры)

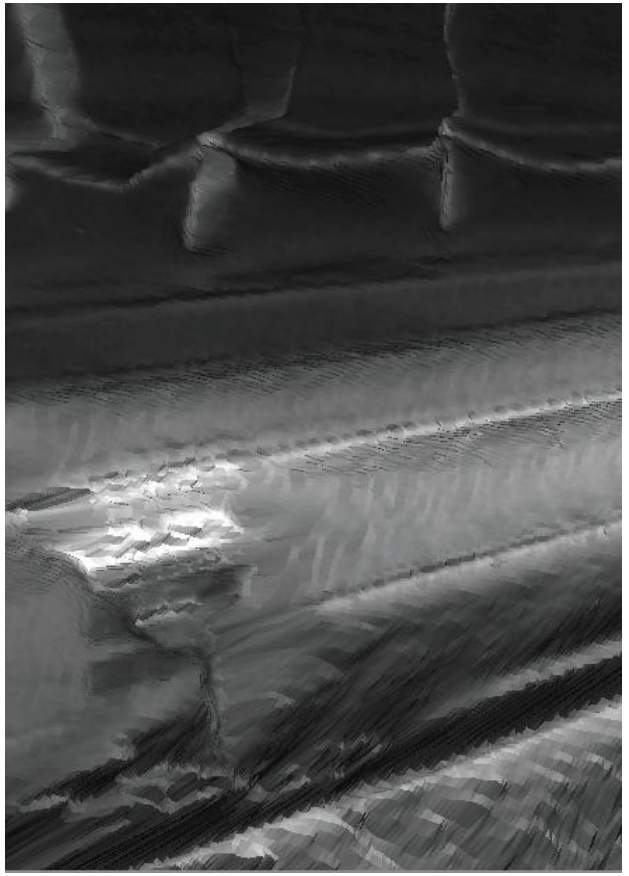




Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

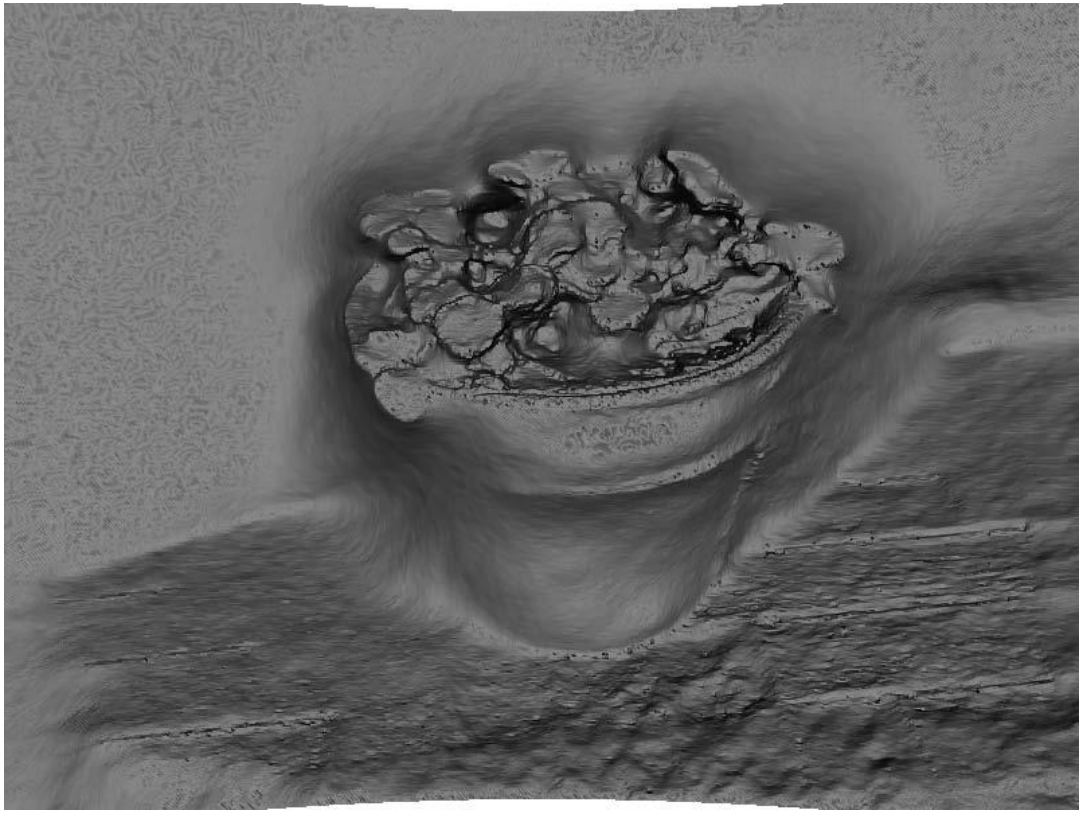
Real-time Depth Estimation
Результат (с текстурой)





Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

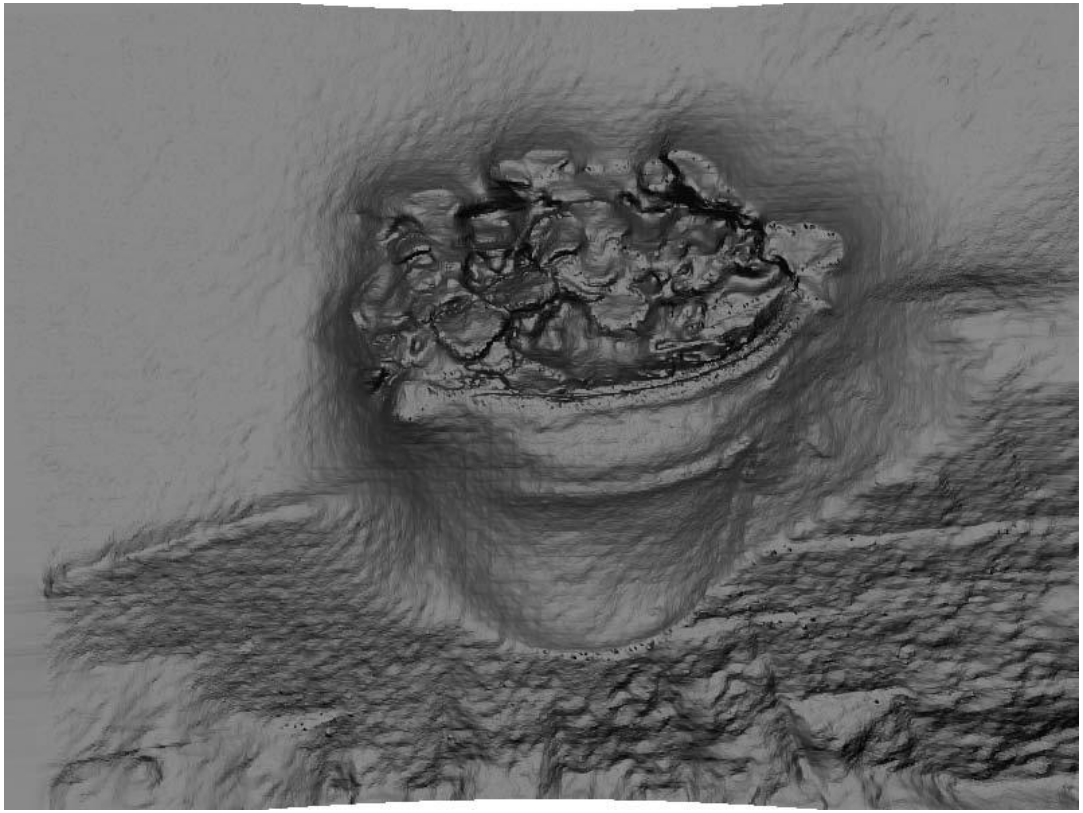
Real-time Depth Estimation
Скорость



High quality: 640x480, 1.8 fps

Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

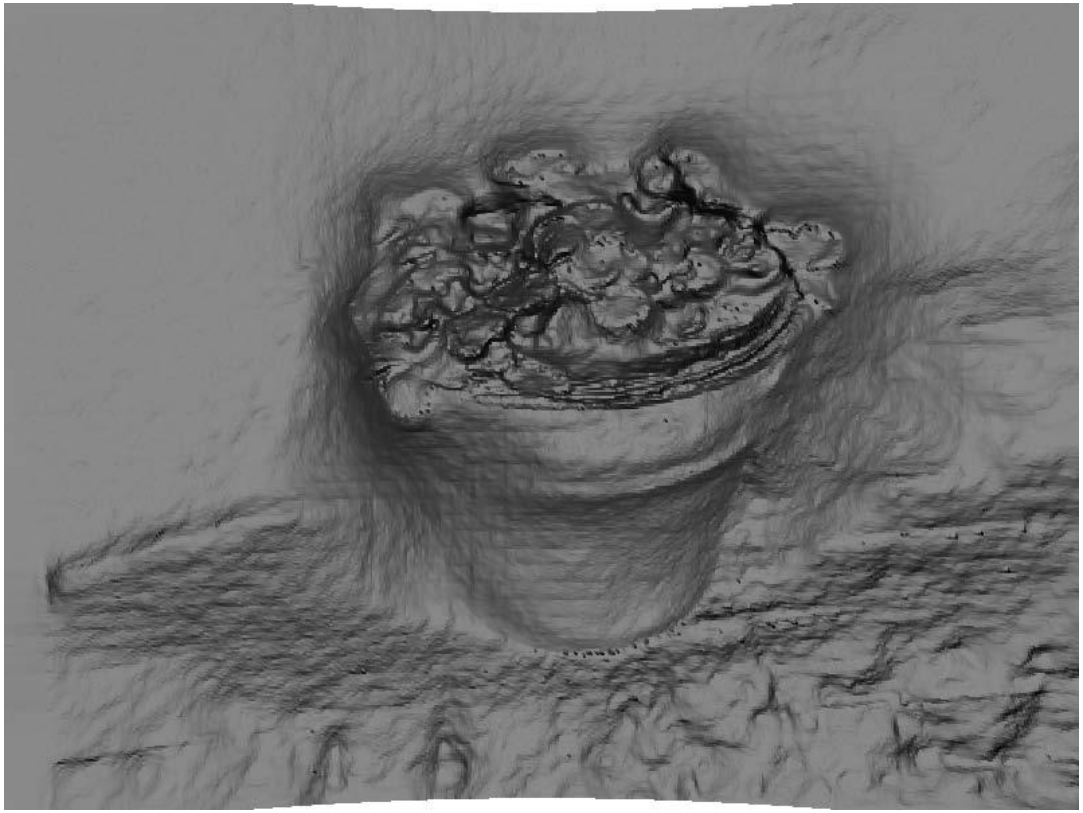
Real-time Depth Estimation
Скорость



Medium quality: 640x480, 11.4 fps

Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

Real-time Depth Estimation
Скорость



Low quality: 480x360, 24 fps

Stuhmer, Gumhold, Daniel Cremers, "Real-Time Dense Geometry from a Handheld Camera", DAGM, 2010

Real-time Depth Estimation
Итоги

- Достоинства
  - Высокая скорость и качество расчета (на NVidia GTX 480)
  - Не требуется калибровка камеры
- Недостатки
  - Не опубликованы результаты работы с реальным видео, а не с последовательностью из небольшого числа изображений

Содержание

- Введение
- Iterative Depth Estimation
- Live Scene Reconstruction
- Real-time Depth Estimation
- **Свой алгоритм**

Свой алгоритм

- Поиск матрицы проективного преобразования между соседними кадрами по данным ME
- Получение параметров движения камеры: перемещение и углы поворота
- Получение относительной глубины блока

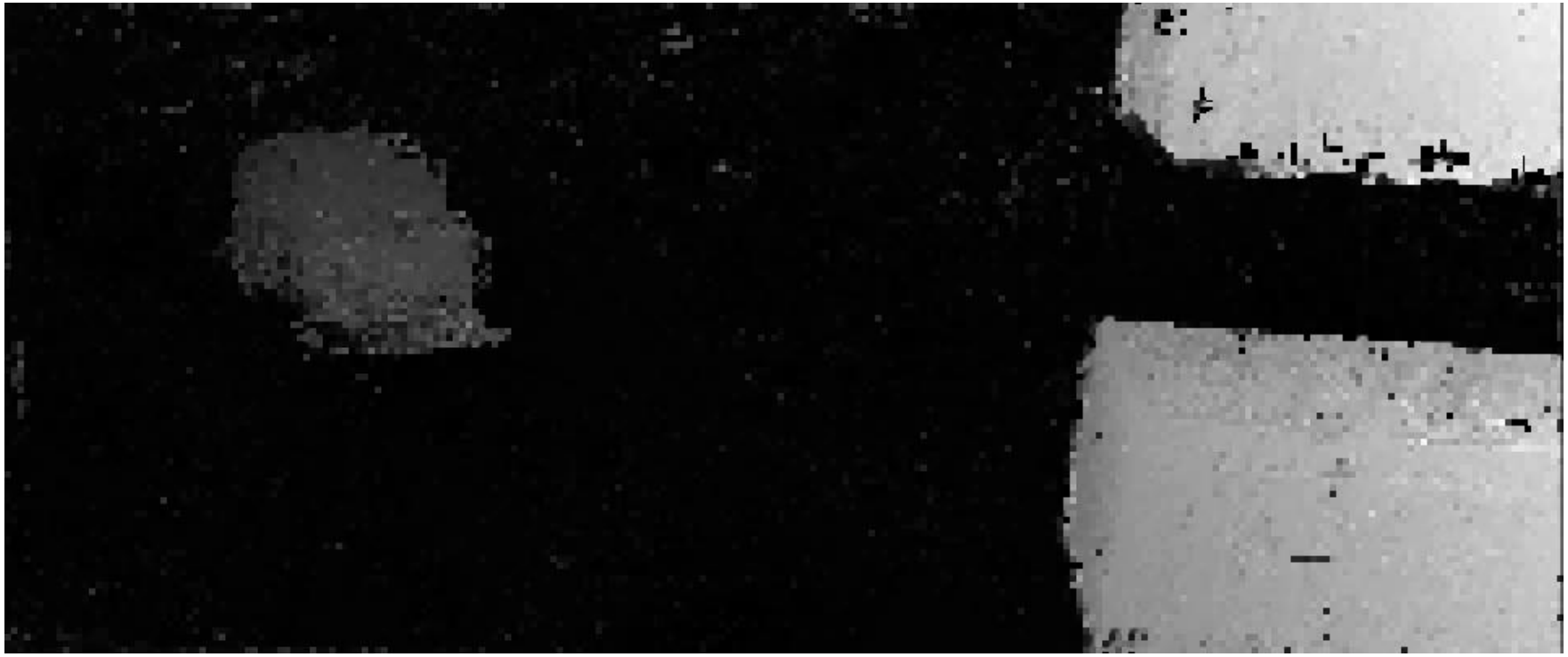
$$Z = \text{magnitude}\,(V_{real} - V_{camera\_motion})$$

V_{real} – motion vector текущего блока

$V_{camera\_motion} = pixel\_pos \times H$, где H – исходная матрица преобразования

Свой алгоритм
Исходное видео

Свой алгоритм
Первоначальная карта глубины



Свой алгоритм
Карта глубины после фильтрации

Свой алгоритм
Проблемы

- Неточная маска фона
- Неверный выбор точек при построении преобразования
- Ошибочное построение глубины для областей с низкой дисперсией и в областях открытия
- Слишком сильное поднятие движущихся объектов по глубине

Свой алгоритм
Дальнейшие планы

- Учет нескольких кадров при получении глубины
- Уточнение преобразования таким образом, чтобы параметры камеры не сильно изменялись от кадра к кадру
- Улучшение алгоритма заполнения областей с низким доверием к motion векторам

Литература

1. Franëcoise Dibos, Claire Jonchery, and Georges Koepfler, “Iterative Camera Motion and Depth Estimation in a Video Sequence”, Computer Analysis of Images and Pattern, 2009
2. Richard A. Newcombe and Andrew J. Davison, “Live Dense Reconstruction with a Single Moving Camera”, Int. Conf. on Computer Vision and Pattern Recognition 2010
3. Jan StËuhmer, Stefan Gumhold, and Daniel Cremers, “Real-Time Dense Geometry from a Handheld Camera”, The German Association for Pattern Recognition DAGM, 2010