

Тестовое задание

SLAM team

Репроектор

1 Введение

В мире существуют разные модели камер, проецирующие трехмерный мир на двухмерную матрицу. Основными являются “идеальная” камера, в русской литературе камера-обскура, в англоязычной – pinhole camera 1. Такая камера не искажает прямые линии.

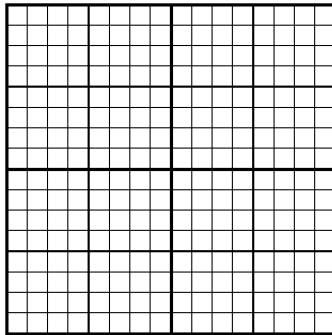


Рис. 1: Grid under Pinhole camera

Для описания работы pinhole модели используется следующие параметры и уравнения:

$$\alpha \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (1)$$

, где f_x, f_y – фокусные расстояния, c_x, c_y – координаты оптического центра. $\begin{bmatrix} u \\ v \end{bmatrix}$ –

координаты пикселя на картинке. $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$ – 3D координаты точки, α – неизвестная.

Вторым основным примером является камера с радиальной дисторсией. Такая камера искажает объекты тем сильнее, чем дальше они от оптического центра камеры 2.

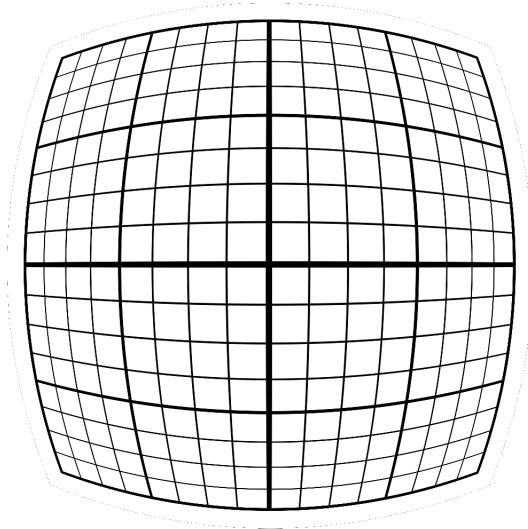


Рис. 2: Grid under Radially distorted camera

Для описания модели с радиальной дисторсией приведено уравнение, обратное проецирующее точку на картинке в луч:

$$\alpha \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{u-c_x}{f_x} \\ \frac{v-c_y}{f_y} \\ 1 + \lambda \left(\left(\frac{u-c_x}{f_x} \right)^2 + \left(\frac{v-c_y}{f_y} \right)^2 \right) \end{bmatrix} \quad (2)$$

От pinhole модели данная модель отличается дополнительным известным параметром λ , который в случае картинки выше был -0.2 .

2 Постановка задачи

Необходимо написать программу на C++, принимающую на вход:

- исходную модель камеры и её параметры
- исходное изображение
- модель камеры, в которую необходимо спроецировать изображение, и её параметры
- путь для сохранения финального изображения

На выходе должна быть одна картинка, которая является перепроецированием исходного изображения, описанного исходной моделью камеры на другую модель камеры.

Можно использовать любые библиотеки для чтения изображений и при желании распараллеливания, но вся математика должна быть написана на библиотеке Eigen. Для сборки кода лучше всего использовать CMake.

Код должен содержать:

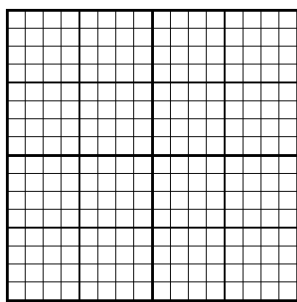
- для каждой модели камеры метод проецирования 3D точки на изображение (нужно будет вывести)
- для каждой модели камеры метод обратного проецирования точки с изображения в луч (нужно будет вывести)
- тесты, подумайте каким образом можно протестировать написанные вами функции, классы
- Dockerfile для сборки и запуска проекта

Опционально:

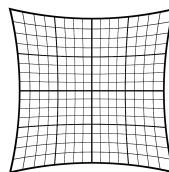
- Можно попробовать распараллелить, векторизовать ваше решение
- При выполнении предыдущего пункта сравнить скорость с первоначальным решением, через например google benchmark

Ожидается, что выполнение задачи должно занять сильно меньше 8 часов. Просьба оставить последние 30 минут на написание небольшого отчёта, своих мыслей касательно того, что ещё можно улучшить.

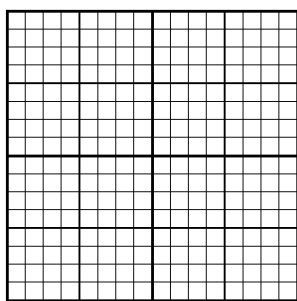
3 Примеры результатов перепроецирования



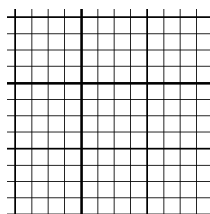
(a) Pinhole



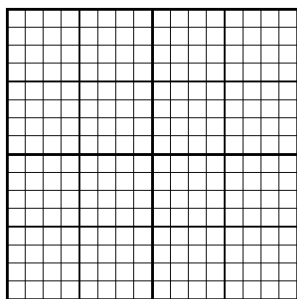
(b) Radially $\lambda = +0.1$



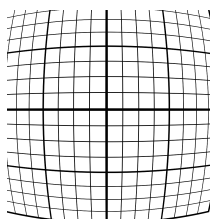
(a) Pinhole



(b) Pinhole with shifted center



(a) Pinhole



(b) Radially with shifted center