

Изучение механизма работы существующих решений по определению уровня освещения

Введение

В задаче определения уровня освещения основной целью является измерение цвета, направления, интенсивности и других параметров источника света, падающего на сцену, без использования явного объекта с сенсорами этой информации.

Для решения данной задачи используются различные подходы, среди которых — извлечение сигналов и регрессия модели освещения и использование человеческого восприятия для грубого предположения.

ARCore[\[1\]](#)

API оценки освещения позволяет выделять с изображения некоторые характеристики, позволяющие подробно узнать важную информацию об источнике освещения и о самом уровне освещения на сцене, основываясь на сигналах, среди которых:

- Тени;
- Окружающий свет;
- Затенение;
- Блики;
- Отражающие свойства поверхностей.

Тени

По направлению теней можно с достаточной точностью узнать направление источника света (если он один).

Окружающий свет

Это малонаправленный, рассеянный свет, благодаря которому сцена в принципе видна. Как пример — солнце, лампы непрямого освещения (светящие в противоположном направлении от сцены). Рассеянным светом также называется свет, который не падает на сцену напрямую с источника, а попадает на неё после отражения от одной или многих других поверхностей; при этом процессе свет теряет интенсивность (в зависимости от отражательных свойств поверхности), и поэтому кажется "мягким".

Затенение

Это свойство любой поверхности (или объекта) отражать свет по-разному в зависимости от угла, под которым на поверхность (или объект) падает свет. Также по изменению цвета освещаемого объекта можно определить как уровень освещения, так и истинный цвет самого объекта.

Блики

Блики на изображении обычно выглядят как практически белые эллипсообразные фигуры. Блики обычно появляются, когда достаточно интенсивный свет падает прямо на линзу (или глаз). Если от какого-то объекта интенсивный отражаемый свет будет падать прямо на линзу камеры, на этом объекте на фотографии будет блик.

Механизм действия *ARCore*

HDR-режим

С помощью машинного обучения (модель, архитектура и использованные датасеты не указаны), используя вышеперечисленные сигналы, *ARCore* может достаточно точно определять направление и интенсивность основного источника света, а также общее окружающее освещение на сцене (используется для реалистичного отбрасывания теней на объектах дополненной реальности).

Режим *Ambient intensity*

Определяет среднюю интенсивность пикселей и вектор цветокоррекции для определения истинного цвета сцены. В *ARCore* используется для коррекции цвета и баланса белого виртуального объекта в соответствии со сценой[2].

Судя по тому, что данный режим помечен как "грубый", то есть неидеально точный, можно предположить, что используется алгоритм уровня *Grey-World* или *Grey-Edge* без использования машинного обучения.

Выводы

ARCore это мощный инструмент, позволяющий создавать реалистичные приложения с использованием дополненной реальности.

Удалось исследовать основные части механизмов работы алгоритмов по извлечению описанных во вступлении характеристик и сигналов, но, к сожалению, подробностей работы этих алгоритмов найдено не было в силу закрытости проекта.

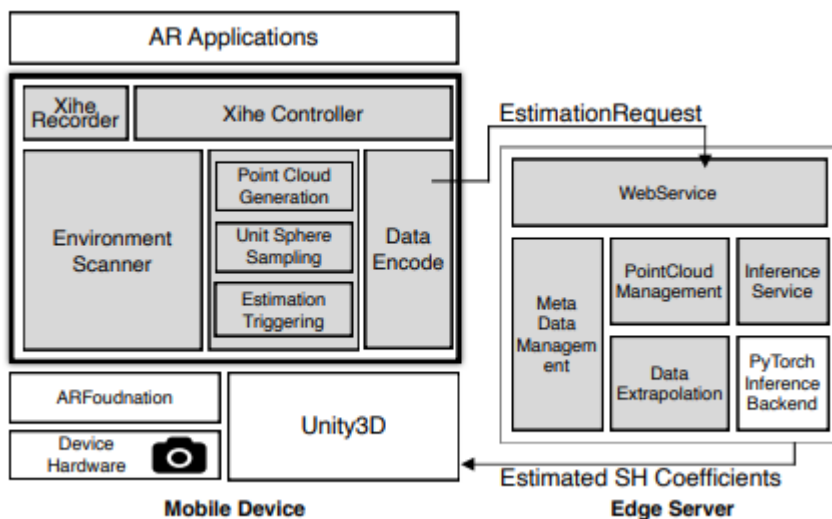
Основываясь на полученных данных, можно создать новое или дополнить уже существующее решение по оценке уровня освещения.

Xihe[3]

Xihe это фреймворк для определения цвета и интенсивности освещения с целью улучшения качества работы мобильных приложений с дополненной реальностью. Доступен на *Android* и *iOS*.

Xihe призван решать проблему корректной репрезентации пространственно-вариантного освещения (при изменении ракурса на одной и той же сцене, уровень освещения и цвета объектов на сцене могут меняться; это и есть вариантность).

Архитектура *Xihe*:

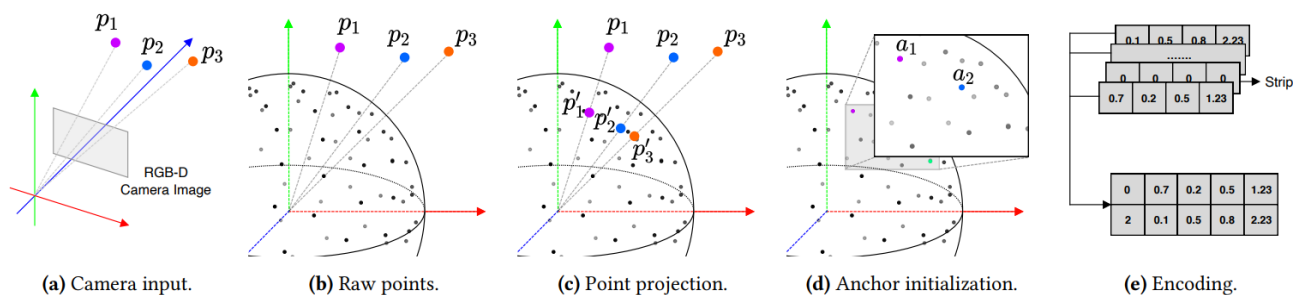


Unit-Sphere based Point Cloud Sampling

Разработчики данного фреймворка предлагают новую технику семплирования под названием *unit sphere-based point cloud*.

Данный метод основан на эмпирическом анализе, показавшем корреляцию между неполными данными (некоторой частью панорамного фото с углом обзора 360 градусов) и точностью предположения интенсивности и цвета освещения. Подробнее см. пункт 3.1 в [3-1]

Пример алгоритма работы данной техники:



PointAR

Мобильное 3д зрение на сегодняшний день всё ещё находится на начальных этапах развития, поэтому разработчики данного решения выбрали модель *PointAR* как основу для создания новой нейронной сети определения освещения.

Вкратце, *PointAR* это двухэтапная конвеерная нейронная сеть, которая состоит из преобразования облака точек для симуляции движения камеры с позиции наблюдения в позицию обработки, и из компактной модели глубокого обучения, основанной на облаке точек.

XiheNET

Это тот же самый *PointAR*, но в первый шаг интегрирована техника *Unit-Sphere based Point Cloud Sampling*.

Датасеты

Про генерирование и разбиение датасетов подробнее смотреть пункт 3.2 в [\[3-2\]](#)

Xihe предоставляет датасеты для тестов по запросу (см. раздел *Datasets* в *README* в гитхабе [\[4\]](#) проекта).

Нужно учитывать, что одной из основополагающих проблем этой статьи было создание модели, позволяющей производить предположения об освещении с очень большой производительностью (в итоге производительность вышла до 24мс на кадр в режиме производительности и до 200мс на кадр в режиме качества).

GLEAM[\[5\]](#)

Статья, приведённая к этой технологии до мельчайших подробностей объясняет всё о *GLEAM*. Поэтому, при необходимости узнать какие-то подробности или дополнительную информацию, стоит обратиться к приложенной статье.

Механизм действия

Система *GLEAM* определяет освещение, основываясь на сэмплировании освещения с помощью рефлексивной геометрии и с помощью генерации якорей пространства через интерполяцию. Эти

операции позволяют динамически обновлять пространственные якоря при их применении на каждый входной кадр.

Пространственные якоря

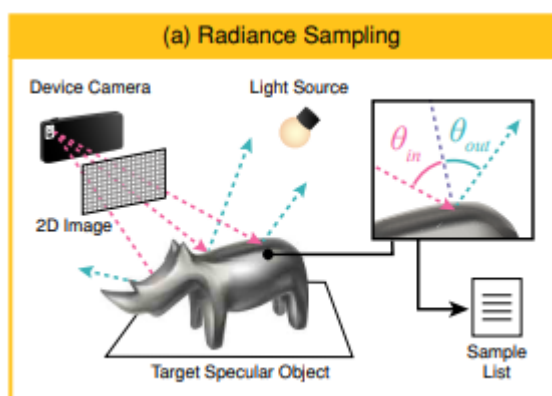
Проще говоря, это привязка конкретного кадра к конкретной точке в пространстве. При этом для следующих входных кадров можно будет учитывать информацию об освещении с других ракурсов. При получении кадра, из него извлекается информация о направлении камеры и предположения о направлении источников света.

Некоторые объекты, поверхность которых отражает свет, позволяют извлечь из фото достаточно точную информацию о направлении и интенсивности освещения при условии, что известна примерная фигура таких объектов.

Рефлективная геометрия

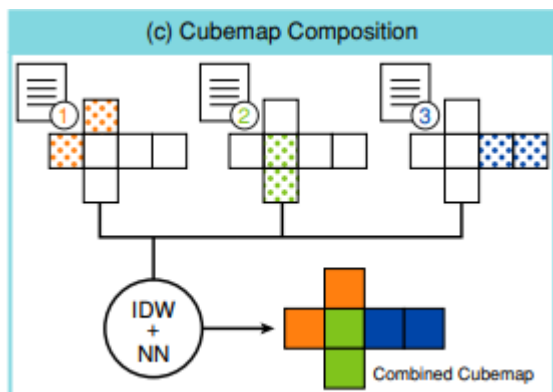
Зеркальное отражение света работает по строгому геометрическому шаблону. Угол наклона падения луча на поверхность совпадает с углом отражения этого угла от поверхности.

Модуль семплирования сияния генерирует набор вариантов сияния specularного объекта с разных ракурсов, собирая информацию о полученном отражении.



Спекулярный объект это объект, поверхность которого хорошо отражает свет. Из-за этого свойства, объект можно использовать для примерного определения геометрии света сцены.

Модуль композиции кубических якорей позволяет создать высококачественную двумерную репрезентацию пространства вокруг точки съёмки в виде куба (см. картинку ниже)



ARKit^[6]

ARKit это закрытый коммерческий проект, и, следовательно, информации о подробных механизмах его работы в открытом доступе нет.

Основываясь на исследованиях, приведённых в [\[5-1\]](#), можно сделать вывод, что технология *ARKit* сильно отстаёт от своих конкурентов.

ARKit это фреймворк для языка *Swift*, предназначенный для разработки приложений, использующих дополненную реальность.

Среди всех его возможностей, самой подходящей теме данного отчёта является возможность извлекать приближённую информацию об освещении на фото или видео потоке.

ARLightEstimate

Класс *ARLightEstimate* позволяет получать два приближённых значения:

- интенсивность света всей сцены (в Люменах);
- температура света всей сцены (в Кельвинах).

Измерение интенсивности света производится, основываясь на внутренней компенсации экспозиции камеры устройства (она

заранее известна, учитывая, что устройство управляется *iOS*), и затем нормировано, чтобы соответствовать архитектурам рендеринга, использующим реалистичные метрики для света.

Измерение температуры света производится аналогично измерению интенсивности света, отсюда логично предположение, что для измерения этих величин используются похожие алгоритмы.

Судя по тому, что описание двух извлекаемых величин очень коротко, но в них отчётливо написано, что температура и интенсивность вычисляется как среднее для всей сцены, по аналогии с *ARCore* можно предположить, что используются алгоритмы уровня *Grey-World* или *Grey-Edge*.

ARDirectionalLightEstimate

Класс *ARDirectionalLightEstimate* позволяет получать три приближённых значения:

- коэффициенты сферических функций;
- направление основного источника света;
- интенсивность основного источника света (в Люменах).

Сферические функции представляют собой компактную математическую модель освещения области вокруг точки в пространстве. Модель описывает распределение и цвета различных источников света.

Направление основного источника света возвращается в виде вектора, направленного в сторону самого интенсивного источника света.

Интенсивность основного источника света возвращается в виде числа с плавающей точкой в Люменах, описывающего яркость источника света, вектор которого был получен выше.

1. [Документация Google об определении освещённости в ARCore](#)↩

2. [Исследование ARCore](#)↩

3. [Статья о *Xihe*](#)↔↔↔
4. [Гитхаб *Xihe*](#)↔
5. [Статья о *GLEAM*](#)↔↔
6. [Статья об *ARKit*](#)↔