

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. ПЕТРА ВЕЛИКОГО
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ И МЕХАНИКИ
ВЫСШАЯ ШКОЛА ПРИКЛАДНОЙ МАТЕМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ФИЗИКИ

ОТЧЕТ ПО КУРСОВОЙ РАБОТЕ
РАСПРЕДЕЛЕННАЯ СЕТЬ ДЛЯ ВИЗУАЛЬНОЙ ОДОМЕТРИИ

Выполнил

студент гр. 3640102/90201

Н.С. Коваленко

Научный руководитель

к.ф.-м.н., доцент ВШПМиВФ ИПММ

А.Н. Баженов

Санкт-Петербург
2021 год

Содержание

Введение	3
Постановка задачи	3
1 Теория	3
1.1 Консенсус в распределенной системе	3
1.2 Paxos	4
1.3 Описание подходов к визуальной одометрии	5
2 Реализация	5
2.1 Используемые данные	5
Заключение	6
Список использованных источников	7

Введение

Визуальная одометрия - это метод оценивания положения и ориентации робота или другого объекта с помощью анализа последовательности изображений, снятых с камеры или нескольких камер, которые расположены на исследуемом объекте.

Визуальная одометрия подходит для точной навигации использующих любой тип передвижения на твёрдой поверхности.

Постановка задачи

Необходимо организовать хранение и обработку изображений с помощью протокола Raft для дальнейшего решения задачи одометрии.

1 Теория

1.1 Консенсус в распределенной системе

Опр. 1.1. Задача консенсуса. Есть N процессов. У каждого из них есть набор данных. Процесс выполняет некоторый распределенный алгоритм и прийти к решению. Для этого требуется:

- **Согласие:** все не отказавшие (не упавшие навсегда) процессы должны завершиться с решением и все эти решения должны совпадать;
- **Нетривиальность:** должны быть варианты исполнения, приводящие к разным решениям (возможно, просто с разными исходными данными или разным исходным состоянием процессов).

Протокол должен завершиться за конечное время. Каждый процесс рассылает всем остальным своё предложение. Каждый процесс ждёт предложения остальных, после чего детерминированной функцией выбирает элемент из множества. У остальных процессов получилось такое же множество, такая же функция, следовательно, такой же результат. Работает даже в асинхронной системе (N^2 сообщений).

Задача достижения консенсуса – задача получения согласованного значения группой участников в ситуации, когда возможны отказы отдельных компонент, предоставление ими некорректной информации, искажения переданных значений средой передачи данных. В целом сценарии нештатного функционирования компонент распределенных систем можно разделить на два класса:

- **Полный отказ компонента.** Характеризуется этот класс проблем тем, что такой отказ приводит к недоступности одного из компонент распределенной системы (или сегментации сети, в случае отказа коммутатора). К этому классу проблем относятся: отказ сервера, отказ системы хранения данных, отказ коммутатора, отказ операционной системы, отказ приложения;
- **Византийская ошибка.** Характеризуется тем, что узел системы продолжает функционировать, но при этом может возвращать некорректную информацию. Допустим, при использовании оперативной памяти при нарушении адреса к данным может привести к считыванию некорректных данных из памяти, ошибки сетевого оборудования могут приводить к повреждению пакетов и т.п.

Ошибки второго класса намного более сложны в обнаружении и исправлении. В целом, Лесли Лэмпортом было доказано, что для исправления Византийской проблемы в N узлах распределенная система должна состоять как минимум из $3N+1$ узлов и должна реализовывать специальный алгоритм консенсуса. Отказоустойчивость на этом уровне требуется по большей части в системах, критичность функционирования

которых крайне высока (например, в задачах космической промышленности).

В кластерных вычислениях под отказоустойчивостью обычно понимают устойчивость системы к полным отказам компонент. Для достижения консенсуса в таких системах и применяется алгоритм Paxos. Алгоритм был предложен Лесли Лэмпортом в 90х годах прошлого века и назван в честь греческого острова Паксос с вымышленной системой организации работы парламента. Для достижения консенсуса данному алгоритму необходимо, чтобы в системе из $2N+1$ узлов функционировали как минимум $N+1$ узла, эти $N+1$ узлы называются «кворум».

Опр. 1.2. Кворум. Это семейство Q подмножеств множества процессов 2^P причем:

- Q замкнуто относительно взятия надмножества, т.е. если $Q \ni A \subseteq B, B \in Q$;
- Любые два элемента (т.е подмножества относительно множества процессов) этого семейства имеют непустое пересечение.

1.2 Paxos

Paxos - это алгоритм консенсуса в распределенной системе, который детерминированно работает в асинхронной системе с отказами узлов, гарантирует корректный консенсус, но не гарантирует, что тот при наличии отказов будет достигнут на конечное время.

Система состоит из следующих объектов:

- Client – клиент распределенной системы, который может отправить запрос и получить на него ответ;
- Proposer – компонент распределенной системы, отвечающий за организацию процесса голосования;
- Acceptor – компонент распределенной системы, имеющий право голоса за принятие или отклонение конкретного предложения от Proposer;
- Learner – компонент системы, который запоминает принятое решение.

Базовый алгоритм Paxos состоит из следующих этапов:

1. Prepare («предложение»). На этой фазе proposer генерирует «предложение» с порядковым номером N и отправляет его всем acceptors. Для каждого из последующих «предложений» номер N должен быть больше выбранного ранее;
2. Promise («обещание»). Каждый acceptor получает «предложение» с порядковым номером N и значением V . Если номер «предложения» больше чем все принятые ранее данным acceptor, он обязан ответить на это сообщение «обещанием» не принимать более «предложений» с порядковым номером меньше N . Если данный acceptor уже принимал какое-либо «предложение», он должен вернуть номер N_i этого «предложения» и принятое значение V_i , в противном случае он возвращает пустое значение;
3. Accept! («принять»). Если proposer получил «обещания» от кворума acceptor, он считает запрос готовым к дальнейшей обработке. В случае, если с «обещаниями» от acceptor пришли также значения N_i и V_i , proposer выбирает V равное значению V_i «обещания» с максимальным N_i . Затем proposer отправляет запрос «принять» всем acceptor, который содержит значения N и V ;
4. Accepted («принято»). Когда acceptor получает сообщение «принять» со значениями N и V , он принимает его только в том случае, если ранее он не «обещал» принимать предложения с номерами строго больше N . В противном случае он принимает значение и отвечает сообщением «принято» всем learner.

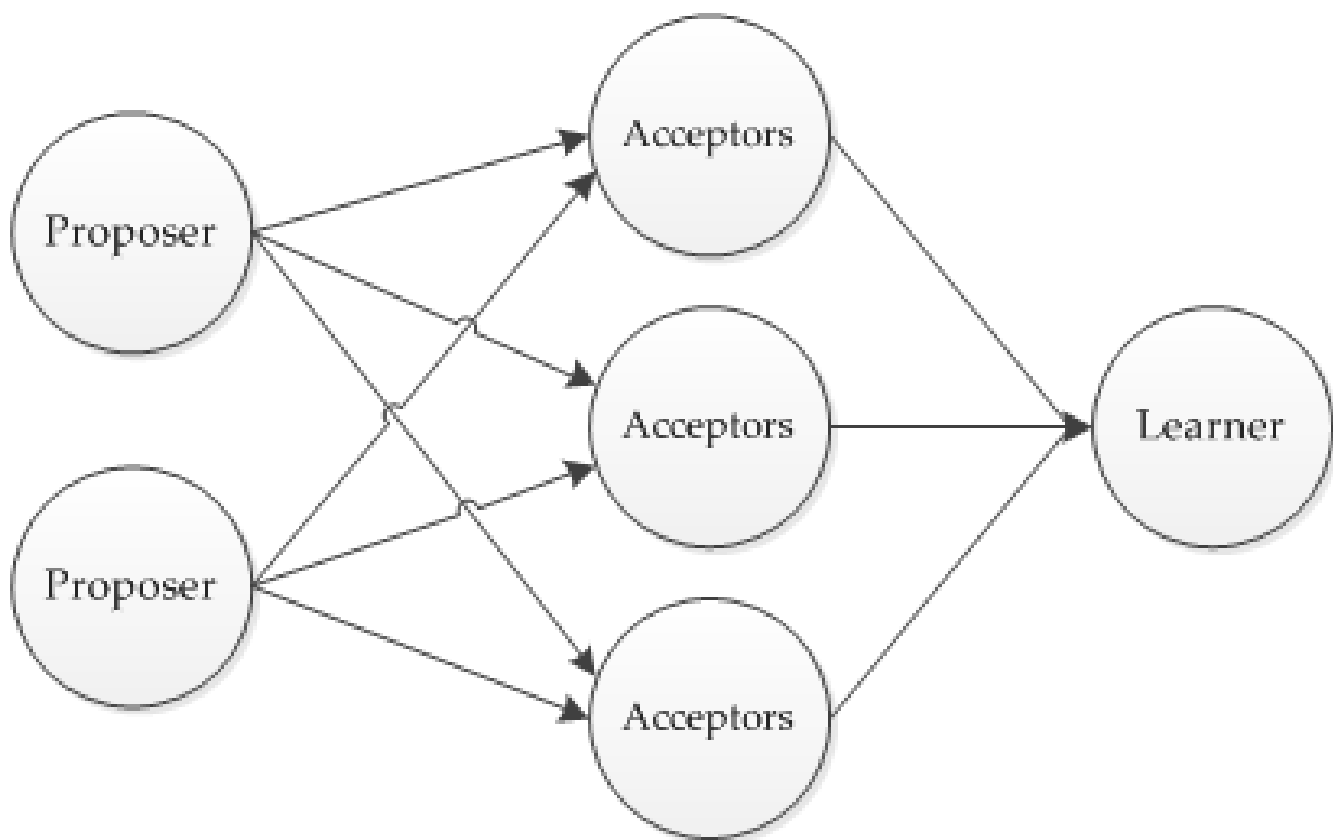


Рис. 1: Схема взаимодействия узлов.

1.3 Описание подходов к визуальной одометрии

Большинство существующих подходов к визуальной одометрии основаны на следующих этапах:

1. Получение входного изображения - получение видео с камер, находящихся на объекте, разбиение видео на кадры;
2. Коррекция изображения - использование методов обработки изображений для устранения искажений объектива;
3. Обнаружение и отслеживание характерных признаков в разных кадрах - поиск признаков, сопоставление изображений и другое;
4. Выявление выбивающихся значений векторов поля оптического потока и их коррекция;
5. Оценка движения камеры по скорректированному оптическому потоку.

2 Реализация

2.1 Используемые данные

Представлен набор данных для оценки точности отслеживания монокулярной визуальной одометрии. Он содержит 50 реальных эпизодов, включающих более 100 минут видео, записанных в различных средах - от узких внутренних коридоров до широких сцен на открытом воздухе.



Рис. 2: Пример представленных изображений.

Заключение

Здесь написано заключение

Список литературы

[1] <https://vision.in.tum.de/data/datasets/mono-dataset>

[2]