



视觉 SLAM 十四讲

从理论到实践

高翔 张涛 等著

中国工信出版集团

电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Preface

이 문서는 중국어 원서인 “입문 Visual SLAM 이론에서 연습까지 14 강([视觉SLAM十四讲 从理论到实践](#))” 책의 원저자로부터 한글 번역 [허가](#)를 받고 구글 번역기를 이용하여 작성된 문서입니다. 본 문서는 아래의 Contribution을 특징으로 합니다.

- 중국어 전공 서적을 구글 번역기를 이용해 한글로 초벌 번역했습니다.
- 초벌 번역 후 매끄럽지 않은 문장은 문맥에 맞게 수정되었습니다.
- 문서 내용 중 참고할만한 웹문서를 코멘트로 추가했습니다.
- SLAM 연구에서 주로 사용되는 용어는 한글로 번역된 용어보다 주로 사용되는 영어로 된 용어 그대로 표시하였습니다.

그럼에도 불구하고 부정확하거나 매끄럽지 않은 부분이 있을수 있습니다. 그런 부분은 코멘트로 제안해주시면 반영하도록 노력하겠습니다. 또한 읽으시다가 잘 이해가 가지 않는 부분도 코멘트로 질문해주시면 답변해드리도록 하겠습니다.

번역 참가자:

신동원 (광주과학기술원 박사과정)
김선호 (VIRNECT 선임연구원)
조원재 (일본국립농업기술혁신공학센터 연구원)
장형기 (Imperial College London 석사과정)
박준영 (광주과학기술원 석사과정)

2018년 10월 1일
신동원 드림

부록A: 가우시안 분포의 본질

다음은 이 책의 여러 곳에서 사용되는 일반적인 가우스 분포의 특성을 요약 한 것입니다.

A.1 가우스 분포

확률 변수 x 가 가우스 분포 $\mathcal{N}(\mu, \sigma)$ 를 따른다면, 그 확률 밀도 함수는

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}\right). \quad (\text{A.1})$$

위 식을 다변수 가우스 분포로 확장하면

$$p(x) = \frac{1}{\sqrt{(2\pi)^N \det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right). \quad (\text{A.2})$$

A.2 가우스 분포적 연산

A.2.1 선형동작

두 개의 독립적 인 가우시안 분포를 설정합니다:

$$x \sim N(\mu_x, \Sigma_{xx}), \quad y \sim N(\mu_y, \Sigma_{yy}),$$

그렇다면 그들의 합은 여전히 가우스입니다.

$$x + y \sim N(\mu_x + \mu_y, \Sigma_{xx} + \Sigma_{yy}). \quad (\text{A.3})$$

x 에 상수 a 를 곱하면 ax 는 다음을 만족시킵니다.

$$ax \sim N(a\mu_x, a^2\Sigma_{xx}). \quad (\text{A.4})$$

$y = Ax$ 인 경우, y 는 다음을 만족합니다.

$$y \sim N(A\mu_x, A\Sigma_{xx}A^T). \quad (\text{A.5})$$

A.2.2 곱

2개의 가우스 분포의 곱을 만족 $p(xy) = \mathcal{N}(\mu, \sum)$ 시키려면:

$$\begin{aligned}\Sigma^{-1} &= \Sigma_{xx}^{-1} + \Sigma_{yy}^{-1} \\ \Sigma^{-1}\mu &= \Sigma_{xx}^{-1}\mu_x + \Sigma_{yy}^{-1}\mu_y.\end{aligned}\tag{A.6}$$

이 공식은 임의의 개수의 가우스 분포의 곱으로 일반화 될 수 있습니다.

A.2.3 복합 연산

또한 $\mathbf{x}$ 와 $\mathbf{y}$ 를 고려해보십시오. 독립적이지 않다면 합성 분포는 다음과 같습니다.

$$p(\mathbf{x}, \mathbf{y}) = N\left(\begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}\right).\tag{A.7}$$

조건부 확률 분포 $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$ 에서 미분 될 수 있으며 조건부 확률 $p(\mathbf{x}|\mathbf{y})$ 은 다음을 만족합니다.

$$p(\mathbf{x}|\mathbf{y}) = N(\mu_x + \Sigma_{xy}\Sigma_{yy}^{-1}(\mathbf{y} - \mu_y), \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx}).\tag{A.8}$$

A.3 복합적 예제

다음은 칼만 필터와 관련된 예입니다. 임의의 변수를 고려할 때 $x \sim (\mu_x, \sum_{xx})$, 다른 변수 $\mathbf{y}$ 는 다음을 만족합니다.

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b} + \mathbf{w}\tag{A.9}$$

여기서 $\mathbf{A}$, $\mathbf{b}$ 는 선형 변수의 계수 행렬 및 오프셋이고, $\mathbf{w}$ 는 잡음 항이고, $\mathbf{0}$ 평균의 가우시안 분포이다:

$$\mathbf{w} \sim N(\mathbf{0}, R)$$

$\mathbf{y}$ 의 분포를 봅시다. 이전 소개에 따르면 다음을 시작할 수 있습니다.

$$p(\mathbf{y}) = N(\mathbf{A}\mu_x + \mathbf{b}, R + \mathbf{A}\Sigma_{xx}\mathbf{A}^T).\tag{A.10}$$

이것은 칼만 필터의 예측 부분에 대한 이론적 근거를 제공합니다.

부록B: ROS 시작하기

ROS는 로봇 연구 분야에서 널리 탐구되는 주제입니다. 책을 너무 많이 읽지 않으려면 신체와 일상에서 언급하지 않았습니다. 그러나 근년에는 주요 대학의 학생들이 점차 ROS를 추진하고 있으며, 점차적으로 사람들에게 알려지고 받아 들여지고 있습니다. 따라서 ROS도 여기에 소개되어 독자에게 도움이되기를 바랍니다.

B.1 ROS란 무엇입니까?

ROS (Robot Operating System)는 2007년 Willow Garage가 공개한 오픈 소스 로봇 운영 체제입니다. 소프트웨어 개발자가 로봇 응용 프로그램을 개발할 수 있는 많은 훌륭한 도구와 라이브러리를 제공합니다. 동시에 코드를 지속적으로 제공하는 뛰어난 개발자도 있습니다. 본질적으로 ROS는 실제 운영 체제가 아니라 오히려 운영 체제를 기반으로 한 소프트웨어 패키지입니다. 실제 로봇에서 발생할 수 있는 여러 가지 알고리즘 (네비게이션, 통신, 경로 계획 등)을 제공합니다.

ROS 용 버전 코드는 알파벳순으로 구성되어 있으며 우분투 시스템과 함께 출시됩니다. 일반적으로 1 피트 ROS 버전은 2 ~ 3개의 우분투 시스템 버전을 지원합니다. ROS는 상자 거북 (Box Turtle)을 시작으로 이 글 (2016년) 현재 KineticKame으로 업데이트되었습니다 (그림 B-1 참조). 동시에 ROS는 완전히 리팩토링되었으며,보다 실시간 버전 2.0이 도입되었습니다.

ROS는 많은 운영체제를 지원합니다. 가장 완벽한 지원은 Ubuntu와 그 미분 버전 (Kubuntu, Linux, Mint, Ubuntu GNOME 등)입니다. 다른 Linux 배포판, Windows 등을 지원하지만 완벽하지는 않습니다. 따라서 개발 및 연구를 위해 Ubuntu 운영 체제를 사용하는 것이 좋습니다. ROS는 널리 사용되는 객체 지향 프로그래밍 언어 C++ 및 스크립팅 언어 Python을 지원합니다. 원하는 언어를 선택하여 개발할 수 있습니다.



그림 B-1 각 ROS 버전의 이름 지정 방법.

B.2 ROS의 특징

ROS의 원래 의도는 로봇 개발이 컴퓨터 개발과 마찬가지로 기본 하드웨어와 인터페이스의 불일치를 막고 결국 소프트웨어를 재사용할 수 있게 하는 것입니다.

소프트웨어 재사용도 중 하나 인 다이애나는 소프트웨어 엔지니어링의 발전을 집중 아름다운, ROS 우리는 디자인의 측면에 초점을 필요로하는 알고리즘을 만드는 통합 메시징 형식이 될 수 있지만, 기본 하드웨어의 기본 목적은 다양한 메시지를받을 것입니다, 이미지, 데이터 등 각 하드웨어 제조업체는 수신 된 데이터를 ROS에서 지정한 통합 메시지 형식으로 통합하여 사용자가 다양한 오픈 소스 로봇 관련 방법을 쉽게 사용할 수 있도록합니다.

등 ORB-SLAM, ORB-SLAM2, LSD-SLAM, SVO, DVO, RTAB-MAP, RGBD-SLAM-V2, 벡터 SLAM, Gmapping, Rovio의, 강의 (14)에 언급 된 SLAM 일반적인 오픈 소스 프로그램입니다 ROS 버전의 오픈 소스 코드는 ROS에서 쉽게 실행, 디버그 및 수정할 수 있습니다.

SLAM 프로그램을 디버깅 할 때 데이터 소스는 대개 3개의 메트릭 : 센서, 데이터 세트 및 bag 파일을가집니다. 해당 센서가 없다면 일반적으로 가상 데이터를 사용하여 SLAM 프로그램을 실행해야 합니다. 그 중에서도 ROS에서 bag 파일을 사용하여 topic를 게시하는 가장 편리한 방법은 SLAM 프로그램이 실제 센서를 사용하여 데이터를 수집하는 것처럼 topic가 전송 한 데이터를 모니터 할 수 있습니다. 나중에 우리는 bag 파일을 사용하여 실제 센서 데이터를 시뮬레이션하는 방법을 간략하게 소개 할 것입니다.

B.3ROS의 신속한 시작방법

ROS는 잘 정립 된 위키 시스템을 가지고 있습니다. 따라서 공식 웹 사이트 (<http://wiki.ros.org/ROS/Installation>)의 소개에 따라 해당 버전의 ROS를 시스템에 설치 한 다음 ROS와 함께 제공되는 교육 프로그램을 읽으십시오. ROS의 기본 개념, 주제 릴리스 및 구독을 배우고 Python 및 C++로 제어 할 수 있습니다. 문제가있는 경우 ROS 용으로 사용자 정의 된 Ubuntu를 사용할 수도 있습니다.

http://www.aicrobo.com/ubuntu_for_ros.html

--

기본외에도 다음과 같은 몇가지 ROS도구를 배울 수 있습니다.

1. **rqt**. Rqt는 플러그인에서 다양한 사용하기 쉬운 GUI (사용자 그래픽 인터페이스)를 제공하는 ROS의 소프트웨어 프레임 워크입니다. Rqt는 매우 강력하며 ROS에 흐르는 메시지를 실시간으로 볼 수 있습니다.
2. **rosbag**. Rosbag는 ROS에서 제공하는 주제 데이터를 기록하고 재생하는 데 매우 유용한 도구입니다. 실제로 SLAM 프로그램을 실행하려고하지만 실제 센서가 없다면 공개적으로 사용 가능한 백 파일을 사용하여 이미지 또는 데이터를 시뮬레이션하는 것을 고려할 수 있습니다.이 방법은 실제 센서를 사용하는 것과 다르지 않습니다. . **rosbag** 사용법은 ROS 위키 페이지를 참조하십시오. 또한 많은 공용 데이터 세트는 데이터 파일을 bag 형식으로 제공합니다.
3. **rviz**. Rviz는 ROS가 제공하는 시각화 모듈로서 실시간으로 ROS에서 이미지, 포인트, 맵, 계획된 경로 등을 볼 수 있으므로 프로그램을 디버그 하기가 더 쉽습니다.

우리는 로봇의 하드웨어 및 소프트웨어 측면을 통일 된 아키텍처의 방향으로 이동해야하며 ROS는 소프트웨어 아키텍처의 표준화에서 중요한 이정표라고 생각합니다. 실험실 연구, 제품 데모 또는 회사의 연구 개발 단계에 광범위하게 사용되기 전에 어떤, ROS1.X하지만 ROS의 실시간 ROS2의 문제를 해결, 미래를 촉진하기 위해, 직접 실제 제품 개발에 이용 될 가능성이 산업용 로봇 및 서비스 로봇의 적용은 중요한 기여를합니다. 이

부록에서는 SolidWorks의 시각화 도구를 사용하여 SLAM 프로그램 개발을 지원하는 방법, 이점 및 이점에 대한 개요를 제공합니다. 우리는 독자가 체계적으로 ROS를 배우고 ROS를 사용하여 자체 SLAM 프로그램을 개발하기를 바랍니다.