

Zusammenfassung: A Tutorial on Graph-Based SLAM

3. November 2010

Inhaltsverzeichnis

1	Introduction	1
1.1	Filtering	1
1.2	Smoothing	1
2	Probabilistic formulation of SLAM	1
2.1	Dynamic Bayesian network (DBN)	2
2.1.1	Zustandsübergangsmodell	2
2.1.2	Beobachtungsmodell	3
2.2	Graph-based SLAM	3
2.2.1	Frontend	3
2.2.2	Backend	3
3	Graph-based SLAM	4

1 Introduction

SLAM (simultaneous localization and mapping) bietet eine Alternative zur Verwendung von Systemen wie GPS, vor allem wenn solche Systeme nicht verfügbar sind. Die bestehenden Ansätze lassen sich in die zwei Klassen *filtering* und *smoothing* unterteilen. Ein intuitiver Weg, um ein SLAM Problem anzugehen, ist die sogenannte “graph-based formulation”. Hierzu wird ein Graph erstellt, dessen Stützstellen Posen (also Ort und Raumwinkel des Roboters zu bestimmten Zeiten) darstellen und die Verbindungen der Stützstellen durch Sensordaten aufgestellte Zwangsbedingungen beschreiben. Diese Zwangsbedingungen sind nicht zwingend eindeutig, da die Messdaten von einem Rauschen überlagert sind.

Ist ein solcher Graph gefunden, gilt es die Konfiguration der Stützstellen zu finden, die am besten zu den durch die Messungen gegebenen Zwangsbedingungen passt. Die hierfür benötigten Minimierungslösungen sind sehr komplex, weshalb diese Methode einige Zeit gebraucht hat, um sich zu etablieren.

1.1 Filtering

Filtering ist eine live Zustandsberechnung, wobei der Zustand aus der aktuellen Roboterposition und der Karte besteht. Die Berechnung wird durch das Aufnehmen weiterer Daten immer weiter verbessert. Die Filtering Methoden werden als on-line SLAM Methoden bezeichnet, um ihre “wachsende” Natur zu unterstreichen.

1.2 Smoothing

Die *smoothing* Ansätze verwenden im Gegensatz zu den *filtering* Ansätzen den kompletten Datensatz an Karten und Positionsinformationen. Diese Ansätze behandeln das sogenannte “full SLAM problem”. Typischerweise werden hier Minimierungstechniken wie die Methode der kleinsten Quadrate verwendet.

2 Probabilistic formulation of SLAM

Die Lösung des SLAM Problems besteht darin, die Robotertrajektorie und die Karte der Umgebung zu berechnen, während der Roboter sich in ihr bewegt. Wegen dem starken Rauschen der Sensoren wird das SLAM Problem meistens mit wahrscheinlichkeitstheoretischen Mitteln behandelt. Man nimmt an, dass der Roboter sich entlang einer Trajektorie bewegt, welche durch eine Reihe von zufälligen Variablen beschrieben wird $\mathbf{x}_{1:T} =$

$\{\mathbf{x}_1, \dots, \mathbf{x}_T\}$. Während dieser Bewegung nimmt der Roboter eine Reihe von odometrischen Daten $\mathbf{u}_{1:T} = \{\mathbf{u}_1, \dots, \mathbf{u}_T\}$ und Bilder der Umgebung $\mathbf{z}_{1:T} = \{\mathbf{z}_1, \dots, \mathbf{z}_T\}$.

Die Lösung des “full SLAM” Problems besteht nun darin die bedingte Wahrscheinlichkeit der Robotertrajektorie $\mathbf{x}_{1:T}$ und Karte \mathbf{m} aus gegebenen Messdaten und einer Anfangsposition \mathbf{x}_0 zu Berechnen:

$$p(\mathbf{x}_{1:T}, \mathbf{m} | \mathbf{z}_{1:T}, \mathbf{u}_{1:T}, \mathbf{x}_0) \quad (1)$$

Die Anfangsposition \mathbf{x}_0 definiert die Position der Karte und ist beliebig wählbar. Die Roboterposen und odometrischen Daten werden generell als 2D oder 3D Transformationen in den Euklidischen Gruppe SE(2) bzw. SE(3) dargestellt. Für Darstellung der Karte bestehen diverse Möglichkeiten. Die Wahl der Darstellung hängt von Faktoren wie den Benutzen Sensoren, der Art der Umgebung oder dem Berechnungsalgorithmus ab. Unabhängig von der Wahl der Darstellung definiert sich die Karte durch die Messungen und die Orte an denen die Messungen erlangt wurden.

Da die Berechnung der bedingten Wahrscheinlichkeit (1) Arbeiten in hochdimensionalen Zustandsräumen umfasst, ist diese nur durch die wohldefinierte Struktur des SLAM Problems zu lösen. Diese Struktur basiert auf der Annahme einer fixen Welt, und der Markov Annahme¹.

2.1 Dynamic Bayesian network (DBN)

Die Struktur des SLAM Problems kann mit einem DBN, einem grafischen Modell, welches einen stochastischen Prozess mittels eines gerichteten Graphen beschreibt, behandelt werden. Der Graph enthält Knoten welche $\mathbf{x}_{1:T}$, $\mathbf{u}_{1:T}$, $\mathbf{z}_{1:T}$ und \mathbf{m} repräsentieren. Die Verbindungen der Knoten folgen einem periodischen Prinzip, welches von dem Zustandsübergangsmodell und dem Beobachtungsmodell charakterisiert wird. Das SLAM Problem als DBN darzustellen betont seine zeitabhängige Struktur und zeigt, dass dieser Formalismus hervorragend geeignet ist um *filtering* Prozesse zu beschreiben die verwendet werden können um SLAM Probleme anzugehen.

¹Die Eigenschaften jedes Zustands werden nur von den Eigenschaften des unmittelbar vorausgegangenen Zustands beeinflusst.

2.1.1 Zustandsübergangsmodell

Das Zustandsübergangsmodell $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$ wird von zwei Linien repräsentiert welche zu \mathbf{x}_t führen und stellt die Wahrscheinlichkeit, dass der Roboter sich zur Zeit t in \mathbf{x}_t befindet dar. Vorausgesetzt der Roboter hat sich zur Zeit $t - 1$ am Ort \mathbf{x}_{t-1} befunden und unterwegs die odometrische Messung \mathbf{u}_t erlangt.

2.1.2 Beobachtungsmodell

Das Beobachtungsmodell $p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m}_t)$ beschreibt die Wahrscheinlichkeit, dass der Roboter die Beobachtung \mathbf{z}_t macht, sofern er sich am Ort \mathbf{x}_t in der Karte befindet. Dargestellt wird das Modell Durch Pfeile die auf \mathbf{z}_t zeigen. Die Beobachtung \mathbf{z}_t hängt nur von der gegenwärtigen Position \mathbf{x}_t und der statischen Karte \mathbf{m} ab.

2.2 Graph-based SLAM

Eine Alternative zum DBN bietet die *Graph-based* Formulierung des SLAM Problems. Hier werden die Posen des Roboters durch Knoten im Graphen dargestellt und mit der entsprechenden Position in der Umgebung beschriftet. Räumliche Zwangsbedingungen zwischen den Posen welche aus Beobachtungen \mathbf{z}_t oder odometrischen Messungen \mathbf{u}_t resultieren werden durch Verbindungslinien zwischen den Knoten im Graphen dargestellt. Eine solche Zwangsbedingung besteht aus der Wahrscheinlichkeitsverteilung über der relativen Transformation zwischen den zwei Posen. Ist ein solcher Graph konstruiert so wird versucht die Konstellation von Posen zu finden unter der die Zwangsbedingungen zwischen den Posen am besten erfüllt sind.

2.2.1 Frontend

Die Erstellung des Graphen bezeichnet man üblicherweise als *frontend* oder *graph construction*. Der Graph wird aus den rohen Messdaten der Sensoren erzeugt. Diese kommen entweder von odometrischen Messungen oder durch Ausfluchten bzw. Abstimmen der Beobachtungen welche an den zwei Punkten gemacht wurden.

2.2.2 Backend

Die Optimierung des Graphen wird als *backend* oder *graph optimization* bezeichnet. Die Optimierung hängt nicht vom verwendeten Sensor ab, sondern nur von der abstrakten Darstellung der Daten.

3 Graph-based SLAM

Die Messdaten werden durch die Verbindungen im Graph dargestellt und können als “virtuelle Messungen” angesehen werden. Genauer repräsentiert die Verbindung zweier Posen eine Wahrscheinlichkeitsverteilung über den relativen Orten der Posen im Bezug auf ihre gegenseitige Messungen. Allgemein ist das Beobachtungsmodell $p(z_t|x_t, m_t)$ multimodal weswegen die Annahme einer Gaußverteilung nicht zutrifft. Dies bedeutet, dass eine einzelne Beobachtung z_t mehrere potentielle Verbindungen zwischen verschiedenen Posen darstellen könnte. Deshalb müssen die Gesamtheit der Verbindungen im Graph selbst als Wahrscheinlichkeitsverteilung beschrieben werden. Eine direkte Behandlung der Multimodalität würde zu einer kombinatorischen Explosion der Komplexität führen. Aus diesem Grund schätzt man meistens die Wahrscheinlichste Topologie ab, man muss also die Zwangsbedingung bestimmen die am wahrscheinlichsten aus einer Beobachtung resultiert. Diese Entscheidung basiert auf der Wahrscheinlichkeitsverteilung über den Roboterposen. Bekannt ist dieses Problem unter dem Namen *data association* und wird normalerweise vom Frontend durchgeführt. Um die richtige *data association* zu berechnen muss dem Frontend eine konsistente Abschätzung der unbedingten Wahrscheinlichkeit über der Robotertrajektorie $p(x_{1:T}|z_{1:T}, u_{1:T})$ gegeben sein. Dies erfordert eine Verschachtelung der Ausführung von Frontend und Backend während der Roboter die Umgebung erkundet. Weswegen die Genauigkeit und Geschwindigkeit des Backends für eine gutes SLAM System ausschlaggebend ist.

Sind die Beobachtungen von Gaußverteilterm Rauschen überlagert und die *data association* bekannt, ist das Ziel des Algorithmus die Anpassung einer Gaußverteilung an die bedingte Wahrscheinlichkeitsverteilung der Robotertrajektorie. Dies beinhaltet die Berechnung des Erwartungswerts dieser Gaußverteilung als Konfiguration der Knoten die die Wahrscheinlichkeit der Beobachtungen maximiert.