Cartographer SLAM

Cartographer Slam, Google tarafından geliştirilen gelişmiş bir eşzamanlı konum belirleme ve haritalama (SLAM) sistemidir. Robotların ve otonom araçların çevrelerinin son derece doğru ve ayrıntılı haritalarını oluşturmalarını ve aynı zamanda kendilerini bu haritalar içinde gerçek zamanlı olarak konumlandırmalarını sağlar. Cartographer Slam sistemi robotik, sürücüsüz otomobiller, endüstriyel otomasyon ve sanal gerçeklik gibi çeşitli sektörlerde yaygın olarak kullanılmaktadır. Tarama eşleştirme, döngü kapatma tespiti ve graf optimizasyonu gerçekleştirmek için lazer tarayıcılardan gelen menzil verileri, odometri pozu ve IMU verileri dahil olmak üzere çeşitli sensör verilerini birleştiren Graf Tabanlı (Graph-Based) SLAM yöntemini kullanır.

Giriş Sensör Verileri (Input Sensor Data)

Cartographer SLAM'in başarısı büyük ölçüde çeşitli kaynaklardan aldığı doğru ve senkronize sensör verilerine dayanır.

a. Menzil Verileri (Range Data):

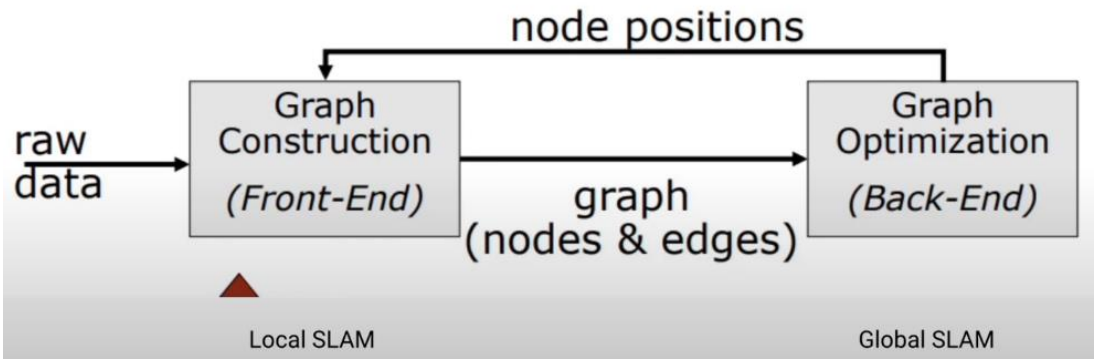
Genellikle LiDAR sensörleri aracılığıyla elde edilen menzil verileri, robot ile çevresindeki nesneler arasındaki mesafeler hakkında bilgi sağlar. Bu ölçümler, 2D veya 3D harita oluşturmak için temel oluşturur. Cartographer, ardışık taramalardan gelen menzil verilerini işleyerek ve hizalayarak robotun hareketini tahmin eder ve yerel SLAM işlemi sırasında poz doğruluğunu artırır.

b. Odometri Pozu (Odometry Pose):

Odometri verileri, öncelikle tekerlek kodlayıcıları (wheel encoders) veya diğer hareket sensörleri aracılığıyla robotun zaman içindeki poz değişikliklerini izler. Bu veriler, SLAM süreci için bir başlangıç noktası olarak hizmet ederek robotun konumu ve yönü hakkında bir ilk tahmin sağlar.

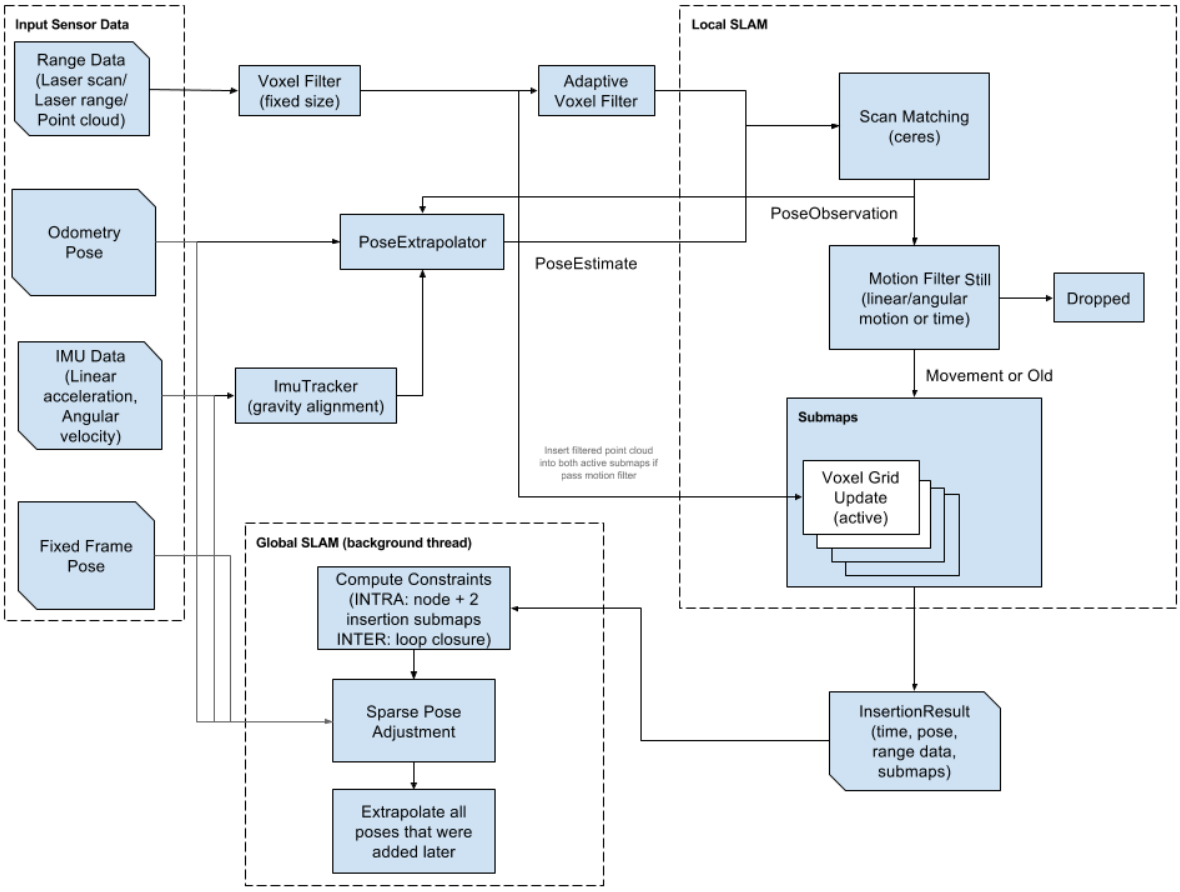
c. IMU Verileri (IMU Data):

Atalet Ölçüm Birimi (IMU) verileri robotun yönelimi, ivmesi ve açısal hızı hakkında bilgi sağlar. IMU verilerinin Cartographer SLAM'e dahil edilmesi, robotun hareket dinamiklerinin hesaba katılmasına yardımcı olur ve özellikle manevralar veya engebeli arazide hareket sırasında poz tahminlerinin doğruluğunu artırır.



Local SLAM (Front-end) ve Global SLAM (Back-end)

Cartographer iki ayrı alt sistemden oluşur: yerel SLAM (frontend veya trajectory builder) ve küresel SLAM (backend). Yerel SLAM'ın görevi iyi alt haritalar üretmektir. Global SLAM'ın görevi ise alt haritaları tutarlı bir şekilde birbirine bağlamaktır.



Local SLAM (Front-end)

Cartographer SLAM'de ön uç olarak da bilinen yerel SLAM iş parçacığı, gelen sensör verilerini işlemekten sorumludur. Cartographer, haritalama sürecini daha küçük bölgelere ayırarak, bellek ve hesaplama kaynaklarını verimli bir şekilde yönetir, döngü kapanışlarını etkili bir şekilde ele alır. Haritalama yolculuğu boyunca harita tutarlılığını koruyarak sağlam ve doğru haritalar oluşturulmasını sağlar.

a. Ceres Kullanarak Tarama Eşleştirme:

Tarama eşleştirme, robotun hareketini tahmin etmek ve pozunu güncellemek için ardışık menzil veri taramalarının hizalandığı yerel SLAM'de kritik bir adımdır. Cartographer SLAM bu görev için güçlü Ceres çözücüsünden yararlanır. Ceres, tarama eşleştirme problemini doğrusal olmayan en küçük kareler optimizasyonu olarak formüle eder ve gözlemlenen menzil verileri ile odometri, IMU verilerine dayalı tahmini menzil verileri arasındaki tutarsızlığı en aza indirir. Rafine poz tahmini, haritalama doğruluğunu önemli ölçüde artırır.

b. Graf Oluşumu:

Robot hareket ettikçe, yerel SLAM iş parçacığı robotun pozları ile gözlemlenen yer işaretleri arasındaki ilişkiyi temsil eden bir graf oluşturur. Her robot pozu grafta bir düğüm (node) olarak temsil edilir, kenarlar tarama eşleştirme ve odometri verilerinden elde edilen kısıtlamaları kodlar. Bu graf zaman içinde gelişerek robotun çevresinin tutarlı ve dinamik bir haritasını oluşmasını sağlar.

c. Alt Haritaların Oluşturulması:

Bellek ve hesaplama kaynaklarını verimli bir şekilde yönetmek için Cartographer SLAM, harita oluşturma sürecini alt haritalar (submaps) adı verilen daha küçük bölgelere ayırarak gerçekleştirir. Her alt harita, ortamın bir bölümünü temsil eder, yerel nokta bulutlarından ve robotun o bölgede veri toplama sırasında yaptığı poz tahmininden oluşur. Robot sürekli olarak çevresini tarar ve yeni sensör verileri geldikçe yeni alt haritalar oluşturulur daha sonra genel haritaya entegre edilir.

d. Alt Haritalar İçinde Döngü Kapatma:

Her alt harita içinde Cartographer SLAM, tekrar ziyaret edilen konumları belirlemek için döngü kapatma tespiti gerçekleştirir. Döngü kapanışları, poz tahminlerindeki hataları düzeltmek ve harita tutarlılığını sağlamak için çok önemlidir. Bir alt eşlemde döngü kapanması tespit edildiğinde, graf optimizasyon süreci döngü kapanma hatasını en aza indirmek için o alt eşlemdeki pozları ve yer işaretlerini ayarlar.

Global SLAM (Back-end)

Yerel SLAM iş parçacığı, yeni sensör verileri elde edildikçe grafı sürekli olarak günceller. Bu sırada, arka plandaki global SLAM iş parçacığında döngü kapatma işlemi gerçekleştirilir. Döngü kapatma, sensör verilerindeki özelliklere veya benzerliklere dayalı olarak daha önce ziyaret edilen konumların tespit edilmesini ve bu konumları birbirine bağlayan kısıtlamalar ekleyerek döngülerin kapatılmasını içerir. Bu süreç ile biriken hatalar düzeltilir, genel harita ve robot yörüngesi iyileştirilmiş olur.

a. Döngü Kapatma Tespiti (Loop Closure):

Arka uç, benzer görünen konumları belirlemek için yerel haritanın özelliklerini tüm yörünge ile karşılaştırır. Bu süreç, robotun tekrar ziyaret ettiği yerleri tanımak için nokta bulutu verilerinden çıkarılan temel özelliklerin eşleştirilmesini içerir.

b. Döngü Kısıtı Oluşturma (Loop Constraint):

Döngü kapanışları tespit edildikten sonra arka uç, graftaki ilgili pozlar arasında döngü kapanış kısıtlamaları oluşturur. Bu kısıtlamalar, döngü kapalı pozlar arasındaki göreceli dönüşümü belirtir ve poz grafına eklenir.

c. Graf Optimizasyonu (Graph Optimization):

Döngü kapatma kısıtlamaları eklendikten sonra poz grafı global optimizasyona tabi tutulur. Arka uç, yerel ve küresel bilgileri uzlaştırmak için pozları ve dönüm noktası konumlarını ayarlayarak graftaki genel hatayı en aza indirir. Bu optimizasyon haritanın tutarlılığını artırır ve biriken yerelleştirme hatalarını azaltır.