

Nawigacja

- * Wymień cztery pytania dotyczące nawigacji, powiązanych funkcji robotów i obszarów sztucznej inteligencji, z których te funkcje korzystają.
- * Wyjaśnij rolę pamięci przestrzennej w nawigacji i cztery sposoby jej utrzymywania.
- * Trasa kontrastowa lub nawigacja topologiczna z układem lub nawigacją metryczną.
- * Określ różnicę między naturalnym a sztucznym punktem orientacyjnym i podaj po jednym przykładzie każdego z nich.
- * Zdefiniuj bramę, stabilność percepcyjną i percepcyjną rozróżnialność.
- * Biorąc pod uwagę opis środowiska biurowego w pomieszczeniach i zestaw zachowań, zbuduj relacyjną reprezentację graficzną opisującą charakterystyczne miejsca i lokalne strategie sterowania za pomocą bramek.
- * Porównaj i skonstrastuj relacyjne i asocjacyjne metody nawigacji topologicznej.

Przegląd

Jak sugeruje termin „mobilny”, roboty mobilne potrzebują inteligencji, aby móc poruszać się w otwartym świecie. Mobilność rodzi kilka pytań, z których najbardziej oczywistym jest: Czym jest nawigacja? Biorąc pod uwagę sukces w powielaniu inteligencji biologicznej, kolejnym oczywistym pytaniem jest: jak nawigują zwierzęta? Z komputerowego punktu widzenia ważne jest, aby odpowiedzieć na pytania: Czy istnieją różne rodzaje nawigacji? a który jest najlepszy? Jak zobaczymy w tym rozdziale, nawigacja w robotyce AI oznacza przenoszenie całego robota z jednej lokalizacji do drugiej, czasami nazywane planowaniem celu. Jak widać z badań nawigacji na zwierzętach, nawigacja jest w rzeczywistości znacznie szersza niż planowanie ścieżek. Planowanie ścieżki docelowej zakłada, że cele nawigacyjne są znane, że istnieje mapa oraz że agent może wykonać ścieżkę i nadążyć za jej lokalizacją. Nawigacja oznacza również, że agent ma pewną formę pamięci przestrzennej, aby utrzymać model świata, chociaż większość zwierząt opiera się na prostych modelach topologicznych. Inteligencja biologiczna wskazuje, że planowanie tras to tylko wierzchołek góry lodowej zrozumienia nawigacji. W ten sposób metody sztucznej inteligencji dla tych czterech różnych aspektów nawigacji (planowanie misji, planowanie ścieżek, lokalizacja i mapowanie) zostaną rozłożone na wiele części. Planowanie misji zostało już omówione w Części 12. W tej i następnej Części skupimy się na planowaniu ścieżek. Wiele prac w zakresie robotyki AI poświęcono na stworzenie algorytmów planowania ścieżki, które traktują robota jako holonomiczny pojazd, który podróżuje między rozpoznawalnymi i odrębnymi lokalizacjami na mapie, zasadniczo tę samą abstrakcją i algorytmy używane w aplikacjach nawigacyjnych na smartfony do generowania kierunków; ta praca jest formalnie znana jako planowanie ścieżki. Kierunki abstrahują od szczegółów, takich jak kształt samochodu, autobusu czy osoby, i działają na symbolicznych etykietach, które mają być percepcyjnie ugruntowane w świecie. Na przykład twórca map może generować wskazówki, aby „skręcić w lewo na Bizzell Street” niezależnie od tego, czy dana osoba idzie, czy prowadzi autobus lub samochód, i zakłada, że agent będzie w stanie wykryć możliwość skrętu i że Cross Street jest oznaczona tabliczką z napisem Bizzell Street. Aplikacje do mapowania smartfonów tworzą trasy, które można wykonać bez dokładnej znajomości bezwzględnych współrzędnych agenta. Na przykład „skręć w lewo na Bizzell Street” nie wymaga od osoby znajomości lokalizacji GPS Bizzell Street, tylko topologiczną cechę skrzyżowania z Bizzell Street. Chociaż dobrze jest wiedzieć, jak daleko jest skręt w lewo od ostatniego punktu nawigacyjnego, nie jest to bezwzględnie konieczne. Dzięki temu trasa ta może być wykonana bez ilościowych informacji o lokalizacji i odległości w układzie współrzędnych. Planery ścieżek, które nie generują kierunków ilościowych, nazywane są

planistami ścieżek jakościowych. Jakościowe narzędzia do planowania ścieżek są często nazywane narzędziami do planowania ścieżek topologicznych lub przykładami nawigacji topologicznej, aby podkreślić, że nawigacja zależy od percepcyjnie wyróżniających się cech świata. Należy zauważyć, że planowanie ścieżki topologicznej różni się od uwzględniania cech terenu i wpływu na nawigację, ponieważ teren jest informacją topograficzną, a nie topologiczną. Ta część skupi się na nawigacji jakościowej lub topologicznej, która opiera się na badaniach w dziedzinie wyszukiwania i planowania w ramach sztucznej inteligencji. Zarówno wyszukiwanie, jak i planowanie polegają na znajdowaniu sekwencji działań, które osiągają cele agenta. Przypomnij sobie, że wyszukiwanie jest analogiczne do znajdowania igły w stogu siana, gdzie wiesz, że w stogu siana jest na pewno igła, masz dobre modele tego, czym jest igła i jak wygląda siano, i rozumiesz, że stóg siana ma granice. W tym przypadku wyszukiwanie i planowanie są identyczne; wyszukiwanie umożliwia robotowi znalezienie optymalnej ścieżki od punktu wyjścia do celu. Część przenosi tematy planowania ścieżki metrycznej i planowania ruchu w sytuacjach związanych z pozycją i dynamiką pojazdu do Części 14 oraz jednoczesnej lokalizacji i tworzenia map (SLAM) oraz włączania wrażliwości terenu do nawigacji do Części 15.

Cztery pytania dotyczące nawigacji

Nawigacja była szeroko badana na zwierzętach, a naukowcy postrzegają nawigację jako odpowiedź na cztery pytania niezbędne do przetrwania. Te pytania to:

Gdzie ja idę? Dokąd zmierzam (i dlaczego?) zajmuje się planowaniem misji w AI, gdzie agent wyznacza cele poprzez namysł, jak omówiono w poprzednim rozdziale. Planetarny łazik może zostać skierowany do odległego krateru i poszukiwania określonej skały w kalderze. Robotycy na ogół nie uwzględniają planowania misji na wysokim poziomie jako części nawigacji, zakładając, że robot został skierowany do określonego celu lub został zaprojektowany do określonego celu.

Jaka jest najlepsza droga? To jest problem planowania tras i jest to obszar nawigacji, któremu poświęcono najwięcej uwagi. Jak zauważono we wstępie, metody planowania ścieżek dzielą się na dwie szerokie kategorie: jakościowe (lub trasy) i ilościowe (lub metryczne). Gdzie ja byłem? Jest to obszar tworzenia map, bogatego obszaru badań nad robotyką AI ze względu na pojawienie się nowych czujników i algorytmów do postrzegania zasięgu. Robot eksplorujący nowe środowiska może być częścią jego misji polegającej na mapowaniu środowiska. Mapowanie środowiska wymaga znajomości lokalizacji robota lub jego lokalizacji. Kategoria algorytmów jest obecnie znana jako symultaniczna lokalizacja i mapowanie (SLAM). Akronim SLAM jest również żartem na temat bycia uderzonym przez znaczące wyzwania obliczeniowe związane z lokalizacją i mapowaniem. Ale SLAM to nie jedyny powód, by zapytać: gdzie byłem? Nawet jeśli robot pracuje w tym samym środowisku (np. dostarcza pocztę w biurówcu), może poprawić swoją wydajność, odnotowując zmiany. Można postawić nową ścianę, przestawić meble, przesunąć lub wymienić obrazy. Lub, jak odkrył Xavier, jeden z robotów wewnętrznych w Instytucie Robotyki na Uniwersytecie Carnegie Mellon, niektóre korytarze i foyer są zbyt zatłoczone w określonych godzinach dnia (koniec zajęć, pora obiadowa) i należy tego unikać. Aplikacje, takie jak Waze, wykorzystują informacje o tym, gdzie byli inni kierowcy w celu zastąpienia bezpośredniego doświadczenia, poprawiając w ten sposób wydajność nawigacji poprzez planowanie omijania korków lub zamkniętych dróg.

Gdzie ja jestem? Aby podążać ścieżką lub zbudować mapę, robot musi wiedzieć, gdzie się znajduje; nazywa się to lokalizacją. Jeśli robot zajmuje się tylko przybliżoną mapą topologiczną, lokalizacja jest dość prosta przy użyciu metod relacyjnych, takich jak wizualne naprowadzanie lub QualNav. Lokalizacja może odnosić się do lokalnego środowiska (np. robot znajduje się na środku pomieszczenia), we współrzędnych topologicznych (np. w Pokoju 311) lub we współrzędnych bezwzględnych (np. szerokość, długość, wysokość). Lokalizacja jest trudniejsza, jeśli lokalizacja musi być dokładna w

bezwzględnej ramce współrzędnych, to znaczy w celu zbudowania mapy w skali. Niestety, jak już widzieliśmy, robot jest bardzo słaby w martwym rachunku. Wyobraź sobie, co dzieje się z tworzeniem map, gdy robot porusza się po obwodzie pomieszczenia, ale nie może rejestrować swoich kroków. Czujniki zasięgu nie eliminują problemu lokalizacji, w rzeczywistości komplikują problem obliczeniowo, mimo że zapewniają bogatszy zestaw danych do wykorzystania w lokalizacji. Zarówno tworzenie map, jak i lokalizacja zostały omówione w części 15. Przed niedawnym zainteresowaniem autonomicznymi samochodami, naukowcy zajmujący się sztuczną inteligencją demonstrowali niezawodną nawigację w pomieszczeniach w złożonych środowiskach już w 1998 roku. Roboty Rhino i Minerva Tour Guide stanowią doskonały przykład tego, jak te cztery pytania pojawiają się naturalnie w aplikacjach. Rhino i Minerva zostały zaprojektowane przez naukowców z CMU i Uniwersytetu w Bonn do pełnienia wszystkich funkcji przewodnika po muzeum, w tym prowadzenia grup ludzi na wystawy na żądanie i odpowiadania na pytania. Rhino był gospodarzem wycieczek po Deutsches Museum w Bonn w Niemczech, podczas gdy Minerva była używana w Narodowym Muzeum Historii Smithsonian w Waszyngtonie. Roboty-przewodniki musiały wiedzieć, gdzie w danym momencie się znajdują (lokalizacja), aby odpowiadać na pytania dotyczące eksponatu lub wskazywać drogę do innego eksponatu. Minerva może również stworzyć niestandardową wycieczkę po żądanych eksponatach, która wymaga, aby robot wiedział, jak dostać się do każdego z eksponatów (planowanie ścieżki). Ścieżka zwiedzania musiała być dość wydajna, w przeciwnym razie robot mógłby wielokrotnie kazać grupie mijać docelowy eksponat, zanim się na nim zatrzyma. Oznaczało to również, że robot musiał pamiętać, gdzie się znajdował. Rhino działała na podstawie mapy muzeum, Minerva stworzyła własną. Ważnym aspektem mapy był czas potrzebny na nawigację między eksponatami, a nie tylko odległość. Ten rodzaj informacji pozwala planistom uwzględnić obszary, których należy unikać w określonych porach dnia. Na przykład Xavier z CMU nauczył się unikać przedzierania się przez konkretny foyer, gdy zmieniają się klasy, ponieważ gęstość poruszających się ludzi spowalnia ich postępy.

Pamięć przestrzenna

Odpowiedź na pytanie Jaka jest najlepsza droga? zależy od reprezentacji świata, z którego korzysta robot, ponownie nawiązując do powiedzenia, że dobre algorytmy są możliwe dzięki dobrym strukturom danych. Reprezentacja świata robota i sposób, w jaki jest on utrzymywany w czasie, to jego pamięć przestrzenna. Pamięć przestrzenna jest sercem modułu Cartographer, który powinien zapewniać metody i struktury danych do przetwarzania i przechowywania danych wyjściowych z bieżących wejść sensorycznych. Załóżmy na przykład, że robotowi polecono „zejść korytarzem do trzecich czerwonych drzwi po prawej”. Nawet w celu koordynacji i kontroli zachowań reaktywnych robot musi zoperacjonalizować pojęcia, takie jak „hala”, „czerwony”, „drzwi”, na cechy, których należy szukać za pomocą schematu percepcyjnego. Musi również pamiętać, ile czerwonych drzwi minęło (i nie liczyć tych samych drzwi dwa razy!). Byłoby również korzystne, gdyby robot wyczuł barierę lub ślepy zaułek i zaktualizował swoją mapę świata. Pamięć przestrzenna powinna być również zorganizowana tak, aby wspierać metody, które mogą wydobyć odpowiednie oczekiwania dotyczące zadania nawigacyjnego. Załóżmy, że kierowany jest robot tym razem, aby „zejść korytarzem do trzecich drzwi po prawej”. Mógłby sprawdzić swoją pamięć przestrzenną i zauważyć, że nieparzyste numery drzwi są czerwone, a parzyste są żółte. Szukając „czerwonych” i „żółtych”, oprócz innych percepcyjnych cech drzwi, robot może bardziej wiarygodnie identyfikować drzwi, albo przez skupienie uwagi (robot wykrywa drzwi tylko na obrazach z czerwonymi i żółtymi obszarami, a nie na każdym obrazie) lub przez fuzję czujników (więcej źródeł danych oznacza pewniejszą percepcję). Pamięć przestrzenna obsługuje cztery podstawowe funkcje:

UWAGA

1. Uwaga. Jakich funkcji lub punktów orientacyjnych powinienem szukać w następnej kolejności?

ROZUMOWANIE

2. Rozumowanie. Czy ta powierzchnia utrzyma moją wagę?

PLANOWANIE ŚCIEŻKI

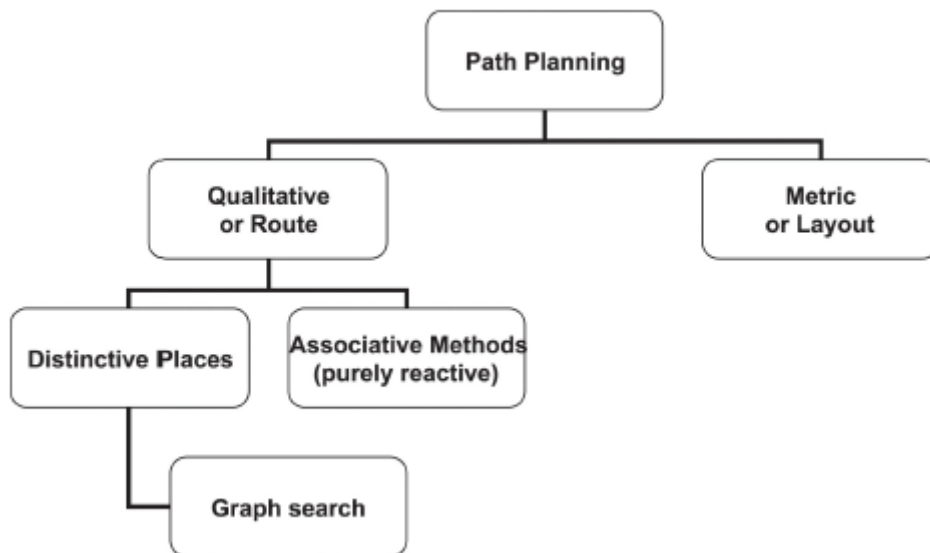
3. Planowanie ścieżki. Jaka jest najlepsza droga przez tę przestrzeń?

ZBIERANIE INFORMACJI

4. Zbieranie informacji. Jak wygląda to miejsce? Czy kiedykolwiek to widziałem? Co się zmieniło od czasu mojej ostatniej wizyty tutaj?

Rodzaje planowania ścieżki

Pamięć przestrzenna przybiera dwie formy: trasy lub jakościową i układu, czyli metryczną lub ilościową, reprezentacje. Wynikiem tego są dwa różne style planowania ścieżki robotyki, pokazane na rysunku



Tutaj dwa style planowania ścieżki manifestują się jako różne typy algorytmów przeszukiwania przestrzeni. Ogólnie rzecz biorąc, algorytmy planowania ścieżki abstrahują robota jako pojazd holonomiczny i abstrahują świat jako graf. Planowanie ścieżki topologicznej wykorzystuje klasyczne algorytmy informatyki, takie jak algorytm najkrótszej ścieżki Dijkstry z jednego źródła, do optymalnego wyszukiwania tras na grafie. Liczba możliwych miejsc do skręcenia (węzłów) jest dość mała, biorąc pod uwagę całkowitą przestrzeń. Planowanie ścieżki metrycznej stosuje bardziej wyspecjalizowane algorytmy planowania ścieżki, często wariant algorytmu A*, które mogą wydajnie obsługiwać planowanie przez duże puste przestrzenie, gdzie każdy blok pustej przestrzeni może być węzłem na grafie.

NAWIGACJA TOPOLOGICZNA

Reprezentacje trasowe, topologiczne czy jakościowe wyrażają przestrzeń w kategoriach powiązań między punktami orientacyjnymi. Przykładem reprezentacji trasy jest sytuacja, w której dana osoba podaje trasę w propozycjach (w postaci listy): „wyjdź z parkingu i skręć w lewo w Fowler Drive. Poszukaj muzeum po prawej i skręć w lewo na następnych światłach”. Zauważ, że nawigacja po trasie jest zależna od perspektywy; punkty orientacyjne, które są łatwe do zauważenia przez człowieka, mogą nie być widoczne dla małego robota pracującego blisko podłogi. Reprezentacje tras mają również

tendencję do dostarczania wskazówek dotyczących orientacji: „z parkingu” (w przeciwieństwie do tego, że jest na nim zawarty), „skręć w lewo”, „w prawo”. Te wskazówki orientacyjne są egocentryczne, ponieważ zakładają, że agent postępuje zgodnie ze wskazówkami na każdym kroku. Reprezentacje tras zakładają również, że istnieje fizyczne uziemienie symbolu. Nawigacja z wykorzystaniem tras jest powszechnie nazywana nawigacją topologiczną, ponieważ wykorzystuje topologię do przestrzennego powiązania punktów orientacyjnych.

NAWIGACJA METRYCZNA

Reprezentacje układu, metryki lub ilościowe są przeciwieństwem reprezentacji tras. Gdy dana osoba podaje wskazówki, rysując mapę, mapa jest reprezentacją układu. Reprezentacje układu są często nazywane reprezentacjami metrycznymi, ponieważ większość map ma pewną przybliżoną skalę do szacowania odległości. Główne różnice między reprezentacjami układu i trasy to punkt widzenia i użyteczność. Reprezentacja układu jest zasadniczo widokiem świata z lotu ptaka. Nie zależy od perspektywy agenta; Zakłada się, że agent jest w stanie przełożyć układ na wyczuwane cechy. Układ jest niezależny od orientacji i położenia. Reprezentacje układu mogą być używane do generowania reprezentacji trasy, ale niekoniecznie działa to w drugą stronę. (Zastanów się, jak łatwo jest czytać mapę i udzielać ustnych wskazówek kierowcy, w przeciwieństwie do rysowania dokładnej mapy drogi, którą jechałeś tylko raz). Większość map zawiera dodatkowe informacje, takie jak skrzyżowania ulic. Agent może użyć tych informacji do wygenerowania alternatywnych tras, jeśli żądana trasa jest zablokowana. Nawigacja przy użyciu metrycznych układów świata jest czasami określana jako nawigacja ilościowa, ale najczęściej jako nawigacja metryczna. Nawigacja trasowa lub topologiczna jest najczęstszą formą nawigacji u zwierząt. Robotycy preferują tworzenie stylów układów pamięci przestrzennej, ponieważ posiadanie układu metrycznego lub mapy może służyć do generowania tras topologicznych lub ścieżek metrycznych. Różnicę między nawigacją topologiczną a metryczną ilustruje różnica między nawigacją autonomicznym samochodem a bezzałogowym statkiem powietrznym. Myślenie o samochodach poruszających się według trasy jest naturalne, ponieważ niezależnie od dokładnej odległości między segmentami — powiedzmy między czterokierunkowym przystankiem na Parkway a sygnalizacją świetlną na Parkway i Avenue — samochód może skręcać tylko w tych zewnętrznie rozpoznawalnych miejscach. W przeciwieństwie do tego, naturalne jest myślenie o nawigacji UAV opartej wyłącznie na punktach GPS. W praktyce samochód wykorzystuje kombinację nawigacji topologicznej i metrycznej, aby przezwyciężyć niepewność i zasygnalizować percepcję. Na przykład pomocna jest wiedza, że odległość między czterokierunkowym przystankiem na Parkway a sygnalizacją świetlną na Parkway i Avenue wynosi 0,8 km. Jeśli auto jedzie więcej niż 1km, widać, że minęło sygnalizację, natomiast jeśli znajdzie sygnalizację na 0,75km, to najprawdopodobniej sygnalizacja poprawna i wystąpił błąd w pomiarze odległości. Podobnie system percepcyjny może nie musieć poświęcać cykli procesora na szukanie sygnalizacji świetlnej, dopóki nie przejedzie 0,5 km poza czterokierunkowy przystanek.

Punkty orientacyjne i bramy

ELEMENT PUNKTU ORIENTACYJNEGO

Nawigacja topologiczna zależy od obecności punktów orientacyjnych. Punkt orientacyjny to jedna lub więcej percepcyjnie wyróżniających się cech obiektu lub lokalizacji. Zwróć uwagę, że punkt orientacyjny niekoniecznie jest pojedynczym, samodzielny obiekt, takim jak „czerwone drzwi”. Punkt orientacyjny może być grupą obiektów; na przykład wielu osobom zdanie „McDonald’s” kojarzy się z wysokim znakiem, jasnym budynkiem o określonym kształcie i bardzo ruchliwym parkingiem. Innym punktem orientacyjnym na zewnątrz może być „stoisko z osikami”. Punkty orientacyjne są używane w większości aspektów nawigacji. Jeśli robot znajdzie punkt orientacyjny na świecie i ten

punkt orientacyjny pojawi się na mapie, robot jest lokalizowany względem mapy. Jeśli robot planuje ścieżkę składającą się z segmentów, potrzebne są punkty orientacyjne, aby robot mógł stwierdzić, kiedy ukończył segment i powinien rozpocząć kolejny. Jeśli robot znajdzie nowe punkty orientacyjne, można je dodać do jego pamięci przestrzennej, tworząc lub rozszerzając mapę.

BRAMY

Dave Kortenkamp spopularyzował szczególnie interesujący, szczególny przypadek punktów orientacyjnych zwany bramą. Brama jest okazją dla robota do zmiany ogólnego kierunku nawigacji. Na przykład skrzyżowanie dwóch korytarzy jest bramą; robot może jechać prosto lub skręcać w lewo lub w prawo. Ponieważ bramy są możliwościami nawigacyjnymi, rozpoznawanie bram ma kluczowe znaczenie dla lokalizacji, planowania tras i tworzenia map.

SZTUCZNE PUNKTY ORIENTACYJNE

NATURALNE PUNKTY ORIENTACYJNE

Punkty orientacyjne mogą być sztuczne lub naturalne. Pojęć „sztuczny” i „naturalny” nie należy mylić z „stworzonym przez człowieka” i „organicznym”. Sztuczny punkt orientacyjny to zestaw funkcji dodanych do istniejącego obiektu lub lokalizacji w celu wspomaganie rozpoznawania punktu orientacyjnego lub innej aktywności percepcyjnej. Znak zjazdu z autostrady międzystanowej jest przykładem sztucznego punktu orientacyjnego. Umieszczono go tam, aby był dobrze widoczny (retrorefleksyjny), a biało-zielona czcionka ma rozmiar zapewniający optymalną widoczność (aktywność percepcyjna to czytanie znaku). Naturalny punkt orientacyjny to konfiguracja istniejących cech wybranych do rozpoznania, które nie zostały wyraźnie zaprojektowane dla czynności percepcyjnej. Jeśli ktoś podaje wskazówki do jej domu, na przykład „weź drugie miejsce zaraz po McDonald’s”, McDonald’s jest używany jako wskazówka orientacyjna do nawigacji do domu. Najwyraźniej McDonald’s nie został zbudowany w celu bycia wskazówką nawigacyjną do prywatnego domu. Fakt, że został wykorzystany w innym celu oznacza, że jest zabytkiem przyrody.

KRYTERIA DLA PUNKTÓW ORIENTACYJNYCH

Niezależnie od tego, czy punkt orientacyjny jest sztuczny czy naturalny, musi spełniać trzy kryteria:

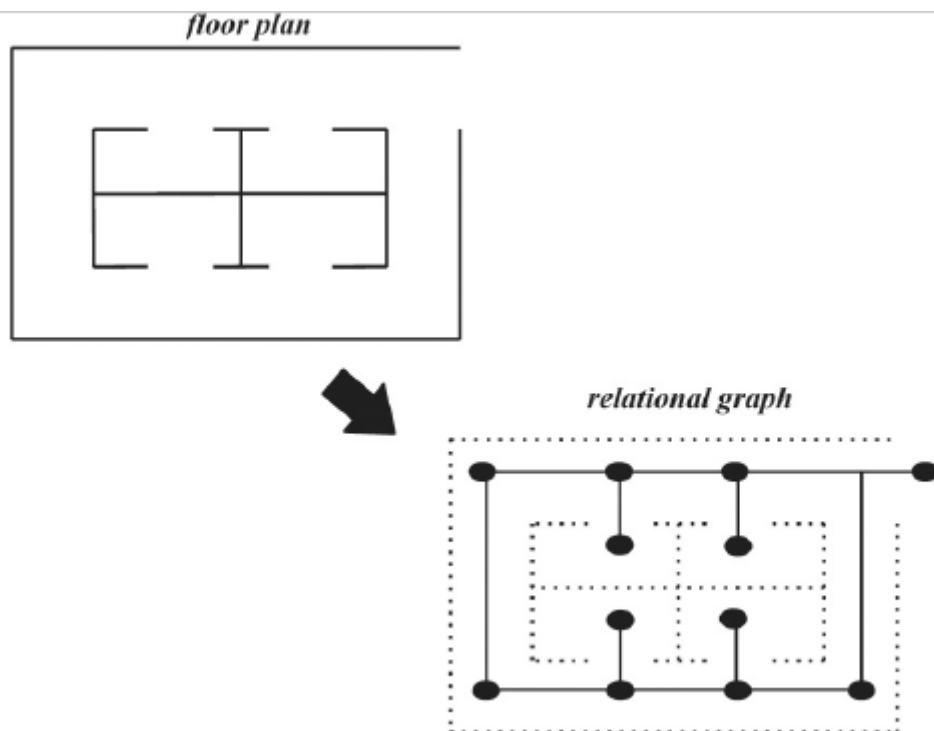
1. Bądź łatwo rozpoznawalny. Jeśli robot nie może znaleźć punktu orientacyjnego, nie jest to przydatne.
2. Wspieraj aktywność zadaniową. Jeśli czynność zależna od zadania jest po prostu wskazówką orientacyjną („weź drugą zaraz po McDonaldzie”), wystarczy bycie rozpoznany. Załóżmy, że punkt orientacyjny ma zapewnić oszacowanie pozycji w celu pokierowania dokowaniem wahadłowca kosmicznego do stacji kosmicznej. W takim przypadku punkt orientacyjny powinien ułatwiać wyodrębnienie względnej odległości do punktu kontaktu.
3. Bądź widoczny z wielu różnych punktów widzenia. Jeśli punkt orientacyjny jest dobrze widoczny, robot może go nigdy nie znaleźć.

Każdy robot miał podążać trasą pomiędzy dowolną sekwencją punktów na arenie. Zespoły mogły oznaczyć punkty drogi sztucznymi punktami orientacyjnymi. Każdy punkt był łatwo rozpoznawalny dzięki szachownicy. Zależną od zadania czynność dojścia do właściwego punktu trasy ułatwiły cylindryczne kody kreskowe, które są unikalne dla każdej stacji. Zauważ, że użycie cylindra gwarantowało, że punkty orientacyjne będą widoczne z każdego punktu widzenia na arenie. Kiedy rozpoczęła się mała liga piłkarska RoboCup, wiele wysiłku włożono w wybór punktów orientacyjnych, które pomogą w nawigacji. Bramki i rogi pierścienia wokół boiska były ozdobione jasnymi kolorami, które można było łatwo wyodrębnić za pomocą algorytmów komputerowych. Względne położenie

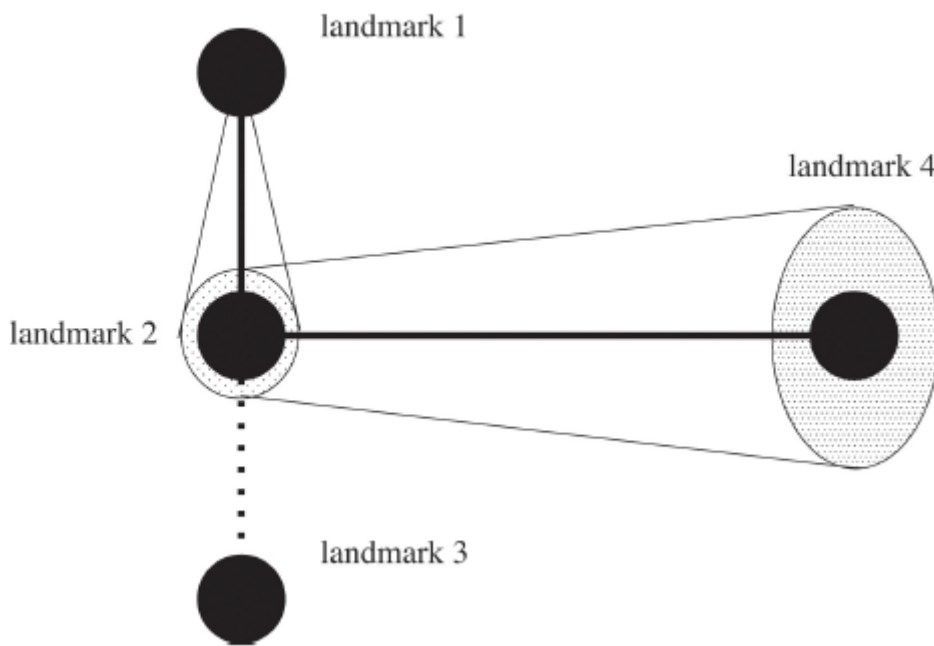
kolorów wskazywało lokalizację (zielony na górze oznaczał po stronie obrońcy, zielony na dole oznaczał stronę przeciwnika). Dobry punkt orientacyjny ma wiele innych pożądanых cech. Powinien być pasywny, to znaczy nie emitować energii, aby był dostępny pomimo awarii zasilania. Powinno być widoczne w całym zakresie, w którym robot może go potrzebować. Powinien mieć charakterystyczne cechy i, jeśli to możliwe, unikalne cechy. Charakterystyczne cechy to te, które są lokalnie unikalne; pojawiają się tylko jako część punktu orientacyjnego z każdego punktu widzenia robota w tym regionie świata (np. na Busch Boulevard jest tylko jeden McDonald's). Jeśli dana funkcja występuje tylko raz w całym regionie działalności (np. w Tampie jest tylko jeden McDonald's), wówczas można ją uznać za globalnie unikalną. Oprócz tego, że jest dostrzegalny dla celów rozpoznania, musi być dostrzegalny dla zadania. Jeśli robot musi ustawić się w odległości nie większej niż 0,5 metra od punktu orientacyjnego, punkt orientacyjny musi być tak zaprojektowany, aby umożliwił uzyskanie takiego stopnia dokładności.

Metody relacyjne

Metody relacyjne przedstawiają świat jako graf lub sieć węzłów i krawędzi.



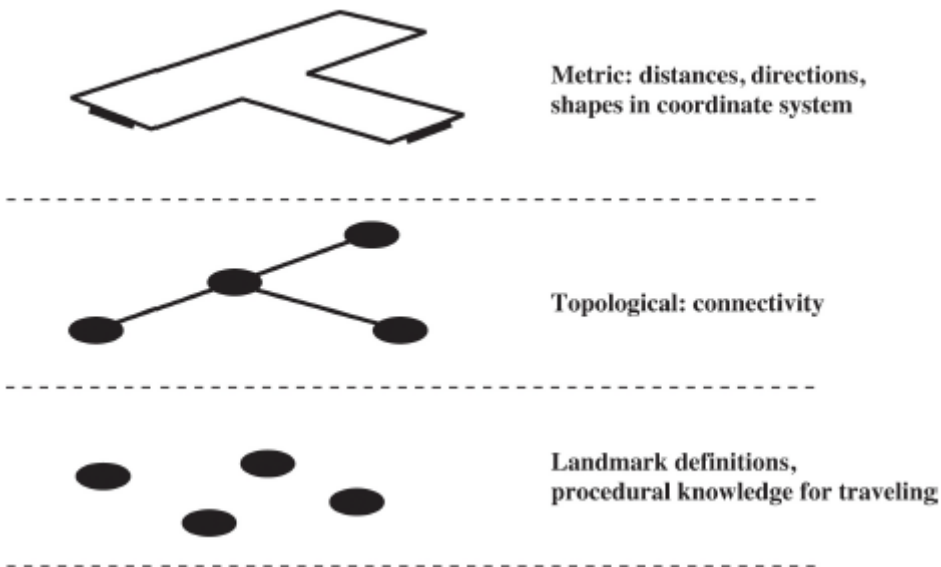
Węzły reprezentują bramy, punkty orientacyjne lub cele. Krawędzie reprezentują ścieżkę żeglowną między dwoma węzłami, w efekcie reprezentując, że dwa węzły mają relację przestrzenną. Do krawędzi mogą być dołączone dodatkowe informacje, takie jak kierunek (północ, południe, wschód, zachód), przybliżona odległość, typ terenu lub zachowania potrzebne do nawigacji tą ścieżką. Ścieżki można obliczyć między dwoma punktami przy użyciu standardowych algorytmów grafowych, takich jak algorytm najkrótszej ścieżki Dijkstry z jednego źródła. Jedno z najwcześniejszych badań nad grafami relacyjnymi do nawigacji zostało przeprowadzone przez Smitha i Cheesemana¹⁹⁷. Przedstawiali oni świat jako graf relacyjny, w którym krawędzie reprezentowały kierunek i odległość między węzłami i symulowali, co by się stało, gdyby robot używał martwego liczenia do nawigować. Jak można się było spodziewać po sekcji dotyczącej propriocepcji w części 10, odkryli, że błąd będzie stale wzrastał i wkrótce robot nie będzie w stanie dotrzeć do żadnego z węzłów.



Charakterystyczne miejsca

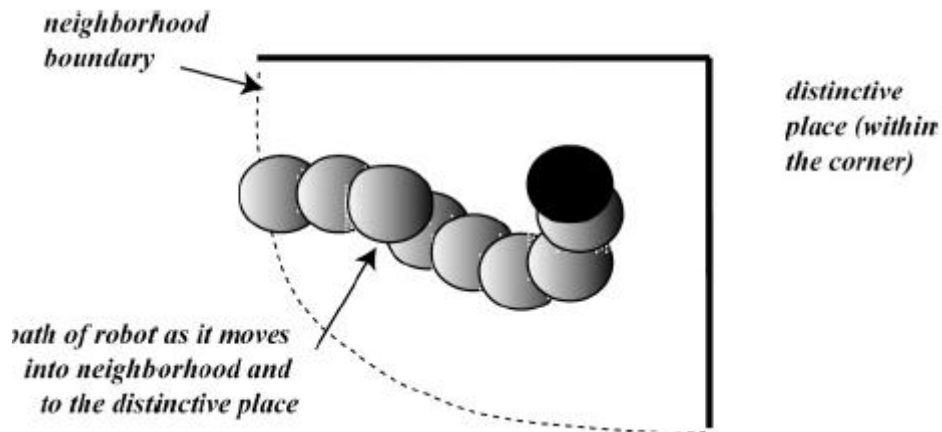
WYRÓZNIAJĄCE SIĘ MIEJSCE

Kuipers i Byun powiązali relacyjne grafy z wyczuwaniem w swojej przełomowej pracy z wyróżniającymi się miejscami. Charakterystyczne miejsca to punkt orientacyjny, który robot mógłby wykryć w pobliskim regionie zwanym sąsiedztwem. Motywacją do ich pracy były badania kognitywistyczne wskazujące, że reprezentacja przestrzenna w królestwie zwierząt tworzy wielopoziomową hierarchię).



(Nowsze badania sugerują, że ta hierarchia nie jest tak wyraźnie podzielona, jak wcześniej sądzono.) Najniższym poziomem lub najbardziej prymitywnym sposobem przedstawiania przestrzeni było identyfikowanie punktów orientacyjnych (drzwi, korytarze) i posiadanie wiedzy proceduralnej potrzebnej do podróżowania między nimi (postępując zgodnie z: przedpokój, przeprowadzka przez drzwi). Następną warstwą była topologiczna. Reprezentował punkty orientacyjne i wiedzę

proceduralną na wykresie relacyjnym, który wspierał planowanie i rozumowanie. Najwyższy poziom to system metryczny, na którym agent poznał odległości i orientację między punktami orientacyjnymi i mógł umieścić je w ustalonym układzie współrzędnych. Wyższe warstwy reprezentowały rosnącą inteligencję. Szczególnie interesująca jest topologiczna reprezentacja Kuipersa i Byuna. Każdy węzeł reprezentuje odrębne miejsce. Po znalezieniu się w pobliżu robot może ustawić się, korzystając z odczytów czujników, w znanym miejscu względem punktu orientacyjnego. Jednym z przykładów charakterystycznego miejsca był róg .

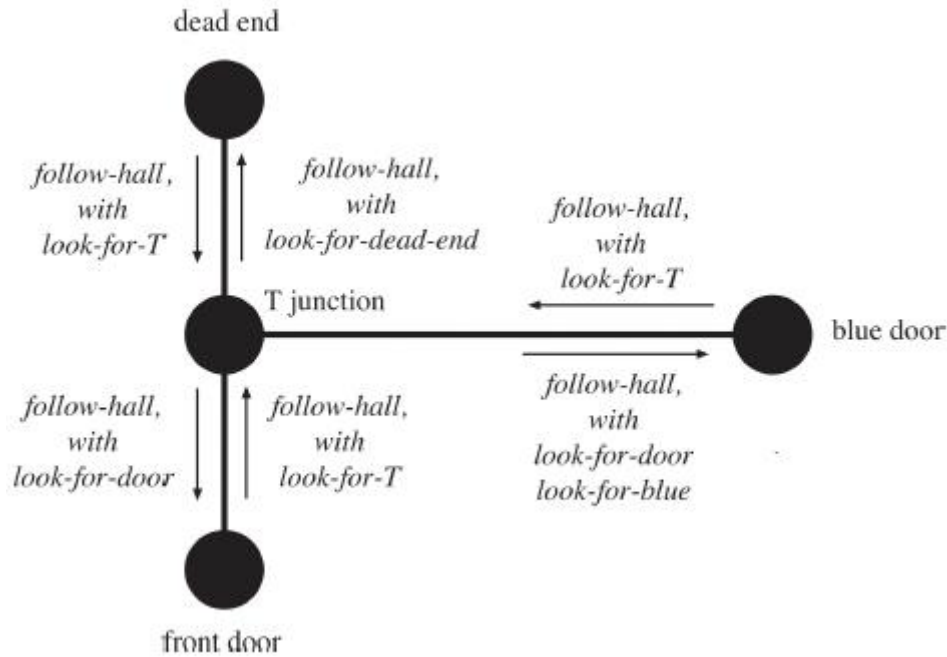


(Kuipers i Byun pracowali w symulacji; róg jako charakterystyczne miejsce nie okazał się realistyczny w przypadku sonarów). Idea sąsiedztwa polegała na tym, że robot poruszałby się po okolicy, dopóki nie osiągnie określonej względnej lokalizacji względem punktu orientacyjnego ; na przykład narożny robot może ustawić się w odległości 1,0 metra od każdej ściany. Wtedy robot byłby zlokalizowany na mapie.

STRATEGIA KONTROLI LOKALNEJ

ALGORYTM WSPINACZKI

Łuk lub krawędź w grafie relacyjnym nazwano lokalną strategią sterowania lub lcs. Lokalna strategia sterowania to procedura przechodzenia z bieżącego węzła do następnego węzła. Gdy robot wykryje punkt orientacyjny, wprowadzi wartości dla zestawu funkcji. Robot wykorzystuje algorytm pokonywania wzniesień. Algorytm wspinania się na wzgórze kieruje robotem dookoła, aż funkcja pomiarowa (np. jak daleko znajdują się ściany) wskaże, kiedy robot znajduje się w pozycji, w której wartości funkcji są zmaksymalizowane (np. obie są oddalone o 1,0 metr). Punktem, w którym zmaksymalizowane są wartości cech, jest miejscem wyróżniającym. Algorytm pokonywania wzniesień to bardzo proste podejście. Chodzi o to, że w przypadku większości skoczni, jeśli zawsze wybierasz następny stopień, który jest najwyższy (nie możesz patrzeć przed siebie), to szybko dostać się na szczyt. Dlatego robot zawsze porusza się w kierunku, który wydaje się powodować najbardziej pozytywną zmianę funkcji pomiarowej. Chociaż badacze odkryli zachowania reaktywne i charakterystyczne miejsca niezależnie od siebie, zachowania reaktywne dobrze odwzorowują charakterystyczne miejsca i lokalne strategie kontroli, jak pokazano poniżej.



Wyobraź sobie robota poruszającego się po szerokiej hali do ślepego zaułka. Lokalna strategia sterowania to zachowanie, takie jak follow-hall, które działa w połączeniu z wyzwalaczem, look-for-corner. Wyzwalacz jest wskazówką eksteroceptywną. Po uruchomieniu robot znajduje się w pobliżu charakterystycznego miejsca, a uwolnione zachowanie, wspinanie się po wzniesieniu do narożnika, kieruje robota na odległość 1 metra od każdej ściany.

Zalety i wady

Koncepcja wyróżniającego miejsca eliminuje wszelkie błędy nawigacyjne w każdym węźle. Robot może zboczyć z kursu, bo hala jest szeroka, ale gdy tylko dotrze do sąsiedztwa, sam się koryguje i lokalizuje. Kuipers i Byun byli w stanie pokazać w symulacji, w jaki sposób robot z błędami martwego liczenia może wykorzystywać wielokrotne podróże między węzłami, aby zbudować rozsądną mapę metryczną, ponieważ większość błędów byłaby uśredniona. Innym atrakcyjnym aspektem podejścia wyróżniającego miejsce jest to, że wspiera ono odkrywanie nowych punktów orientacyjnych, gdy robot eksploruje nieznanne środowisko. Dopóki robot mógł znaleźć coś wyróżniającego się, do czego mógłby się niezawodnie zlokalizować, to miejsce można było umieścić na mapie topologicznej. Następnie, gdy robot wielokrotnie przemieszczał się w to miejsce, robot mógł skonstruować mapę metryczną. Wracając do dyskusji o punktach orientacyjnych, należy zauważyć, że punkt orientacyjny musi być unikalny dla pary węzłów. W rzeczywistym świecie nie może być żadnych narożników, których nie ma na wykresie pomiędzy węzłami, w przeciwnym razie robot będzie się niepoprawnie lokalizował. Podejście wyróżniające miejsce, jak zostało pierwotnie sformułowane, napotkało pewne problemy, gdy roboty bazujący na zachowaniu zaczęły stosować je do prawdziwych robotów. Jednym z najtrudniejszych problemów była percepcja. Trudno o dobre, charakterystyczne miejsca; konfiguracje, które wydawały się przydatne dla ludzi, takie jak narożniki, okazały się trudne do wiarygodnego wycucia i lokalizacji względem. Łatwo wyczuwalne elementy, takie jak narożniki, drzwi i poziome linie, często były zbyt liczne na świecie, a więc nie były lokalnie unikalne. Kolejnym wyzwaniem było poznanie lokalnej strategii sterowania. Ponieważ robot eksplorował nieznanne środowisko, łatwo było wyobrazić sobie, że może znaleźć charakterystyczne miejsca. Ale jak nauczył się odpowiedniej lokalnej strategii kontroli? W środowisku wewnętrznym robot może zawsze uciekać się do podążania za ścianą, nawet jeśli inne zachowania byłyby lepsze. Jak robot mógłby kiedykolwiek spróbować czegoś innego?

Kolejną otwartą kwestią jest problem nieodróżnialnych lokalizacji. Kwestia nieodróżnialnych lokalizacji została do pewnego stopnia rozwiązana za pomocą metod probabilistycznych, które zostaną omówione w części 15.

Metody asocjacyjne

Metody asocjacyjne dla nawigacji topologicznej zasadniczo tworzą zachowanie, które przekształca obserwacje czujników na kierunek, w którym należy iść, aby dotrzeć do określonego punktu orientacyjnego. Metody asocjacyjne są rzadko stosowane w systemach robotycznych, ale ważne jest, aby być ich świadomym. Istnieją podstawowe podejścia: wizualne naprowadzanie, gdzie robot rozpoznaje wizualne wzorce związane z lokalizacjami i trasami (podejście jest podobne do sposobu nawigacji pszczół), oraz QualNav, w którym robot określa swoją lokalizację na podstawie tego, jak zmieniają się względne pozycje punktów orientacyjnych (to podejście jest podobne do nawigacji w konstelacjach). Podstawowym założeniem jest to, że lokalizacja lub punkt orientacyjny będący przedmiotem zainteresowania nawigacji zwykle mają dwa atrybuty:

STABILNOŚĆ PERCEPTUALNA

1. stabilność percepcyjna: widoki lokalizacji znajdujących się blisko siebie powinny wyglądać podobnie

PERCEPTYJNE ROZRÓZNIACZALNOŚĆ

2. percepcyjna rozróżnialność: poglądy odległe powinny wyglądać inaczej.

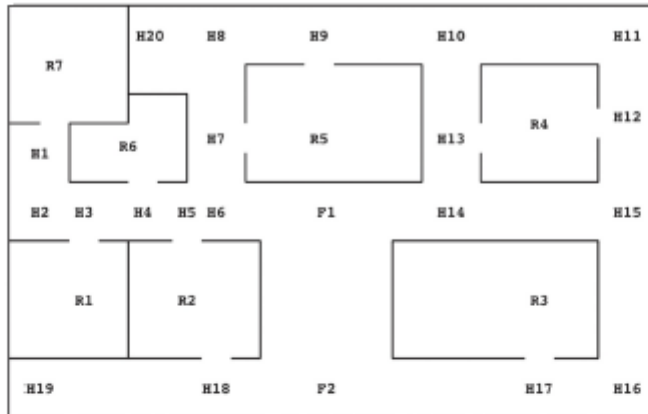
Zasady te są zawarte w idei sąsiedztwa wokół wyróżniającego się miejsca. Jeśli robot znajduje się w pobliżu, widoki punktu orientacyjnego wyglądają podobnie. Główna różnica między wyróżniającymi się miejscami a metodami asocjacyjnymi polega na tym, że metody asocjacyjne wykorzystują bardzo ordynarne widzenie komputerowe. Metody asocjacyjne są bardzo dobre do odtwarzania kroków lub tras.

Studium przypadku nawigacji topologicznej z architekturą hybrydową

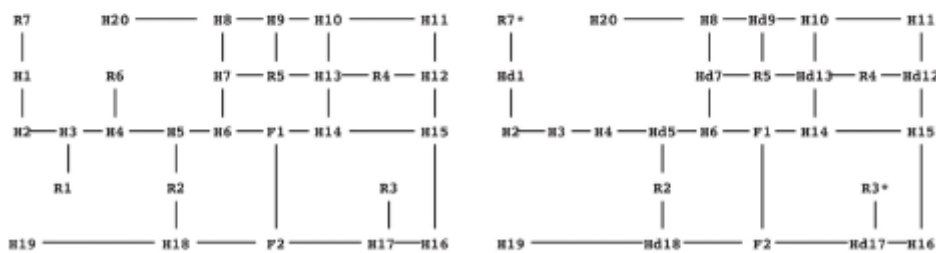
Ta sekcja przedstawia studium przypadku nawigacji topologicznej z wykorzystaniem architektury SFX w konkursie AAAI Mobile Robot Competition w 1994 roku, autorstwa zespołu studentów Colorado School of Mines. W konkursie w 1994 r. odbyła się nawigacja w biurze.¹⁸ Każdy robot został umieszczony w losowym pomieszczeniu, a następnie w ciągu 15 minut musiał wyjść z pokoju i przejść do innego pomieszczenia. Uczestnicy otrzymali mapę topologiczną, ale nie mogli zmierzyć układu pomieszczeń i korytarzy. To studium przypadku pokazuje, w jaki sposób wprowadzono mapę topologiczną, działania Kartografa, w jaki sposób skrypty zostały wykorzystane do uproszczenia zarządzania behawioralnego oraz wyciągnięte wnioski.

Planowanie ścieżki topologicznej

Mapa topologiczna została wprowadzona jako plik ASCII w formularzu Backus-Naur. Przykładowa mapa jest pokazana na rysunku.



a.



b.

c.

Mapa wejściowa składa się z trzech typów węzłów: pokój (R), hol (H) i foyer (F). Zakłada się, że świat jest ortogonalny. Ponieważ krawędzie między węzłami mogą istnieć tylko w czterech kierunkach, wygodnie jest nazywać je północą (N), południem (S), wschód (E) i zachód (W), gdzie N jest ustawione arbitralnie na mapie. Robot otrzymuje swój węzeł początkowy, ale dodatkowym wyzwaniem jest to, że robot nie otrzymuje kierunku, w którym początkowo jest skierowany w stosunku do mapy. Mapa topologiczna jest strukturalnie poprawna, ale niekoniecznie wskazuje, czy korytarz lub drzwi są zablokowane. Takie blokady mogą wystąpić lub zostać przeniesione w dowolnym momencie. Dodatkowym założeniem jest to, aby na zewnątrz każdego drzwi znajdował się punkt orientacyjny, taki jak numer pokoju lub nazwa pokoju. Kartograf w SFX jest odpowiedzialny za budowę trasy. Jako dane wejściowe przyjmuje typ bramy mapy topologicznej oraz węzeł początkowy i węzeł celu, a następnie tworzy listę węzłów reprezentujących najlepszą ścieżkę między początkiem a celem. Kartograf działa w dwóch krokach: wstępne przetwarzanie mapy w celu wsparcia planowania tras i samo planowanie tras. Etap przetwarzania wstępnego rozpoczyna się od zbudowania bazy danych węzłów na mapie wejściowej, przeklasyfikowania węzłów korytarza, które reprezentują połączenie od korytarza do drzwi, jako Hd. Gdy znane są węzły początkowe i docelowe, kartograf eliminuje obce bramy. Węzeł Hd może być połączony z pokojem, którego nie można odwiedzić, to znaczy, że nie jest pokojem docelowym, pokojem początkowym ani pokojem z więcej niż jednym drzwiami. Jeśli pokój nie jest dostępny, oba wpisy węzłów R i Hd są usuwane z bazy danych. Przykładowa wejściowa graficzna reprezentacja topologiczna dla mapy metrycznej jest pokazana na rysunku. Jeśli jako węzeł początkowy wybrano R3, a cel R7, dopracowaną reprezentację graficzną pokazano na rysunku. Zwróć uwagę, że Hd3-R1 i Hd4-R6 zostały wyeliminowane, ponieważ nie były powiązane z pokojami początkowymi lub bramkowymi i nie mogły być używane jako skrót, ponieważ każde z nich miało tylko jedno wejście. Węzły bramy, takie jak H10, pozostają na ścieżce, ponieważ mogą być przydatne w przypadku

wystąpienia zablokowanej ścieżki. Na przykład, jeśli robot jedzie korytarzem z H11 do H8, a ścieżka jest zablokowana, robot będzie musiał wrócić do znanej lokalizacji, aby zmienić orientację i plan. Jeśli jednak zlikwidujemy H10, to robot musi wrócić na sam początek korytarza, ponieważ nie wie, gdzie to jest w stosunku do H10. Aby rozwiązać ten dylemat ewentualnego trzykrotnego przebycia tej samej części korytarza, kartograf utrzymuje węzły, które reprezentują możliwe alternatywne strategie. Należy zauważyć, że różne typy bramek, zadań i możliwości robota będą prowadziły do różnych strategii przetwarzania wstępnego. Optymalna ścieżka jest obliczana przy użyciu algorytmu najkrótszej ścieżki Dijkstry z jednego źródła. Algorytm generuje najkrótszą ścieżkę, biorąc pod uwagę koszty między każdą parą połączonych węzłów; koszty są wyrażone jako długość lub waga krawędzi na wykresie. Ponieważ reprezentacja topologiczna nie jest metryczna, połączenia między węzłami odzwierciedlają preferencje w konstruowaniu tras. Preferencje te są wyrażone jako wagi krawędzi. W tej implementacji poruszanie się po foyer było znacznie bardziej kosztowne obliczeniowo i zawodne; w związku z tym waga krawędzi dla dowolnego podsegmentu ścieżki wychodzącego z foyer została ustawiona na 3. Aby zniechęcić robota do szukania rozwiązań, które były „niegrzeczne” (np. urzędy ludowe). Wszystkie inne połączenia zostały ustawione na 1.

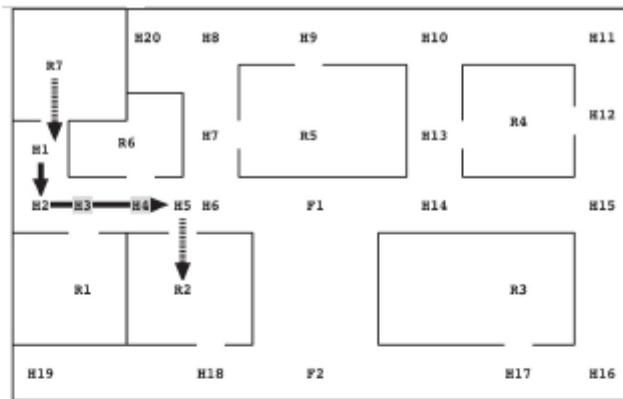
TABELA PRZEJŚĆ

Menedżer zadań w SFX wykorzystuje ścieżkę obliczoną przez kartografa, aby wybrać odpowiednie abstrakcyjne zachowanie nawigacji dla podróżowania między określoną parą węzłów. Utrzymuje wskaźnik do bieżącego węzła i do następnego zamierzonego węzła ze ścieżki. Typ bieżącego i następnego węzła określają odpowiednie zachowanie zgodnie z tabelą przejść, pokazaną poniżej:

	To			
From	H	F	R	H
H	Navigate-hall	Navigate-hall	Undefined	Navigate-hall
F	Navigate-hall	Navigate-foyer	Navigate-door	Navigate-hall
R	Undefined	Navigate-door	Navigate-door	Navigate-door
Hd	Navigate-hall	Navigate-hall	Navigate-door	Navigate-hall

Tabela przejść pokazuje, że nie wszystkie kombinacje węzłów są dozwolone; z definicji robot nie może przejść z węzła hali H do węzła pomieszczenia R bez przejścia przez węzeł Hd. Również stół niekoniecznie jest symetryczny. W przypadku pomieszczeń, drzwi nawigacyjne muszą być używane do wejścia lub wyjścia, ale w przypadku przejścia do foyer będą stosowane różne strategie w zależności od tego, czy robot przemierza halę, czy foyer. Sam ANB wykorzystuje informacje we wpisach bazy danych dla węzłów jako parametry do tworzenia instancji skryptu w bieżącej parze punktów na drodze. Jednym z nowatorskich aspektów tej implementacji jest sposób, w jaki Menedżer zadań obsługuje zablokowaną ścieżkę; nie odwraca aktualnie utworzonego abstrakcyjnego zachowania nawigacji (ANB). Jeśli zachowanie związane z unikaniem przeszkód zostanie opublikowane w strukturze tablicy, że wystąpił warunek ZABLOKOWANY, Menedżer zadań kończy aktualnie aktywne zachowanie abstrakcyjnej nawigacji. Ponieważ robot znajduje się między węzłami, Menedżer Zadań nakazuje robotowi powrót do bieżącego węzła. Ale uruchamia proste zachowanie polegające na przejściu do celu, które pozwala robotowi przeorientować się szybciej, niż byłby w stanie, gdyby musiał przywrócić abstrakcyjne zachowanie nawigacyjne z nowymi parametrami. Gdy robot powróci mniej więcej do ostatniej znanej lokalizacji, Menedżer Zadań żąda od Kartografa nowej ścieżki. Kartograf usuwa połączenie między bieżącym a zamierzonym węzłem, a następnie ponownie oblicza nową ścieżkę z bieżącym węzłem jako początkiem. Po zakończeniu nowej ścieżki Menedżer zadań wznawia kontrolę. Poniższy rysunek przedstawia robota poruszającego się po mapie rozdawanej podczas rzeczywistych zawodów robotów mobilnych AAI. Węzłem początkowym jest R7, a celem jest R2. Kartograf oblicza

ścieżkę jako R7-H1-H2-H5-R2. Zauważ, że H3 i H4 nie są uważane za odpowiednie bramy dla tego zadania i dlatego nie pojawiają się na ścieżce.



```

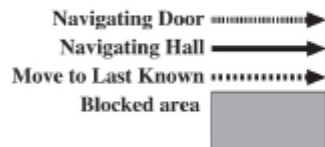
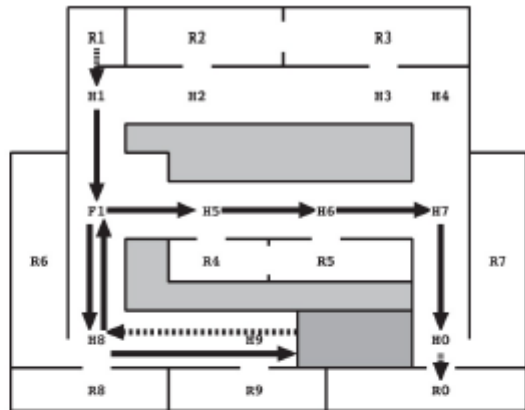
R7 -> R2
R7 - H1 - H2 - H5 - R2
Moving from R7 to R2, going SOUTH
In navigating door behavior
  ultra looking for door towards the: SOUTH
  HEAD AHEAD MOTOR ACTIVE
  Found door - Initialization terminated
  HEAD THROUGH DOOR MOTOR ACTIVE
  Moved through door - Nominal behavior terminated

Moving from H1 to H2, going SOUTH
In navigating hall behavior
  turning towards the: SOUTH
  Turned towards hall - Initialization terminated
  Looking for hall towards the: EAST
  WALL FOLLOW MOTOR ACTIVE
  Found hall - Nominal behavior terminated

Moving from H2 to H5, going EAST
In navigating hall behavior
  turning towards the: EAST
  Turned towards hall - Initialization terminated
  vision looking for door relative: # (right side)
  WALL FOLLOW MOTOR ACTIVE
  Found door (vision) - Nominal behavior terminated

Moving from H5 to R2, going SOUTH
In navigating door behavior
  ultra looking for door towards the: SOUTH
  WALL FOLLOW MOTOR ACTIVE
  following wall on left (right ground truth)
  Found door - Initialization terminated
  HEAD THROUGH DOOR MOTOR ACTIVE
  Moved through door - Nominal behavior terminated
  Goal Reached (success)
  
```

Jak pokazują dane wyjściowe, Kartograf oblicza ścieżkę pokazaną w pierwszym wierszu. Druga linia pokazuje, że aktualnym zadaniem nawigacyjnym jest przejście z R7 do H1. Menedżer zadań wybiera drzwi nawigacyjne, a robot zaczyna od szukania drzwi, ponieważ użytkownik nie określił, gdzie się znajdowały. Po zbadaniu bazy danych wiadomo, że drzwi znajdują się na południe. Dlatego zachowanie inicjuje zachowanie ruchu naprzód na południe. Gdy drzwi zostaną znalezione, faza inicjalizacji zachowania abstrakcyjnego dobiega końca i zostaje wyzwolona nominalna czynność przechodzenia przez drzwi. Gdy robot przejdzie przez drzwi, nominalne zachowanie zostaje zakończone, kończąc również całe abstrakcyjne zachowanie. Kolejnym zadaniem nawigacyjnym jest przejście z H1 do H2, ponownie w kierunku południowym. Kierownik zadania wybiera halę nawigacyjną. Zadanie przejścia z H2 do H5 jest interesujące, ponieważ pokazuje, że warunek zakończenia śledzenia hali różni się od przejścia z H1 do H2. Ponieważ H5 jest bramą do interesującego pokoju, do wizualnej identyfikacji pokoju używany jest inny proces percepcyjny. Po wizualnej identyfikacji drzwi uznaje się, że robot znajduje się w H5. Połączenie H5-R2 tworzy instancję drzwi nawigacji. Jednak w tym przypadku ultradźwięki nie zidentyfikowały jeszcze otwarcia drzwi, dlatego faza inicjalizacji polega na podążaniu za ścianą, aż do rozpoznania otwarcia drzwi. Następnie aktywowane jest zachowanie nominalne, a robot wchodzi do pokoju, pomyślnie wykonując zadanie. Druga symulacja pokazana na rysunku



```

Moving from H8 to H0, going EAST
In navigating hall behavior
  turning towards the: EAST
  TURNED towards hall - initialization terminated
  HALL FOLLOW MOTOR ACTIVE
User stopped or BLOCKED - behavior terminated
In last known location loop
  MOVE TO ORIAL MOTOR ACTIVE

Moving from H8 to F1, going NORTH
In navigating hall behavior
  turning towards the: NORTH
  TURNED towards hall - initialization terminated
  HALL FOLLOW MOTOR ACTIVE
  will follow wall on right (left ground truth) to stay in foyer
Found foyer - Nominal Behavior terminated

Moving from F1 to H5, going EAST
In navigating hall behavior
  following wall on left (right ground truth)
  trying to exit foyer
  looking for hall towards the: EAST
  WALL FOLLOW MOTOR ACTIVE
  Found hall - initialization terminated
  HALL FOLLOW MOTOR ACTIVE
Found door (vision) - Nominal Behavior terminated
  
```

wykorzystuje przykładową mapę topologiczną dostarczoną przed zawodami. Zamierzona ścieżka to R1-H1-F1-H8-H0-R0, ale H8-H0 został zablokowany. Jak pokazano na rysunku, robot wykrywa, że ścieżka jest zablokowana i wykorzystuje ruch do celu, aby powrócić do H8. Kartograf aktualizuje mapę topologiczną i oblicza nową trasę, a robot wznawia wykonywanie, w tym przypadku H8-F1-H5-H6-H7-H0-R0. Wynik poniższego skryptu pokazuje robota potwierdzającego węzły H5 i H6, mimo że plan nie przewidywał, że robot wszedł do powiązanych pomieszczeń. Węzły te były utrzymywane na ścieżce, ponieważ w przypadku napotkania innej zablokowanej hali robot mógłby wtedy wrócić do R5 lub R6 i spróbować użyć trasy przez drzwi pomiędzy nimi.

Skrypty nawigacyjne

Elementy planowania i realizacji ścieżki są wyraźnie przemyślane. Kartograf utrzymuje mapę w formie wykresu i monitoruje postępy. Tabela przejść pełni rolę sekwencera wysokiego poziomu. Skrypty są używane do określania i realizacji dorozumianych szczegółów planu w sposób modułowy i wielokrotnego użytku. Wdrożenie składało się z trzech skryptów do poruszania się po drzwiach, holach i foyer. Pseudokod dla nawigacji-drzwi jest pokazany poniżej w stylu C++ i wykorzystuje właściwości indeksowania instrukcji switch:

```

switch(door)
case door-not-found:
    //initialization phase
    //follow wall until find door
    if wall is found
        wallfollow to door
    else
        move-ahead to find a wall
case door-found:
    //nominal activity phase
    move-thru-door(door-location)

```

Nominalne zachowanie, `move-thru-door`, kończy się samoczynnie, więc nie ma oddzielnego warunku zakończenia. Percepcja drzwi jest kluczowym wyznacznikiem tego, co robi robot. Zachowanie `navigate-hall` jest używane do podróżowania między korytarzami, foyer i korytarzami oraz z korytarzy do węzłów korytarza/drzwi. Zachowanie ma dwa różne warunki początkowe. Jednym z warunków jest to, że robot znajduje się w foyer i wykrywa halę. Ponieważ hala może być nieco z przodu, a `hallfollow` zakłada, że robot jest w korytarzu, skrypt nakazuje robotowi `wallfollow`, aby znaleźć halę, a następnie rozpocząć podążanie za halą. Menedżer zadań wykorzystuje informacje kierunkowe przechowywane w bazie danych, aby określić, za którą ścianą foyer należy podążać. W drugim stanie robot jest już w hali. Ponieważ nie ma gwarancji, że robot będzie zwrócony w stronę linii środkowej hali (ANB nie mają wiedzy o tym, co zostało zrobione przed ich utworzeniem, z wyjątkiem zmiennych globalnych), skrypt dolny ustawia go tak, aby zrównał się z żadaną halą.

```

switch(hall)
case not-facing-hall:
    //initialization phase
    if starting in a FOYER
        if hall-not-found
            wallfollow until find the hall
        else
            if not facing hall
                turn to face hall
    else starting in a HALL
        if not facing hall
            turn to face hall
case facing-hall:
    //nominal activity phase
    hallfollow until next gateway

```

ANB `navigate-hall` kończy działanie, gdy zostanie znaleziona następna oczekiwana bramka (następny węzeł na ścieżce). Istnieją trzy zachowania, które szukają bramek. `hallwatch` szuka ultradźwiękowej sygnatury hali w oczekiwanym kierunku; `foyerwatch` podobnie szuka foyer, a potwierdza, że drzwi wykorzystuje wizję do wykrycia punktu orientacyjnego związanego z pomieszczeniem. Te zachowania działają jednocześnie z zachowaniami nominalnymi. ANB `navigate-foyer` służy do przemieszczania robota między dwoma foyer. Zakłada się, że dwa połączone węzły foyer stanowią wygodną reprezentację jednego dużego foyer z wejściami z różnych kierunków (tj. wiele bram do foyer). Skrypt przenosi robota o n stóp do pierwszego foyer w kierunku drugiego foyer. Dzięki temu robot z dala od potencjalnie mylących sygnatur ultradźwiękowych. Następnie menedżer zadań określa, po której stronie foyer należy podążać wzdłuż ściany, aż zostanie wykryta następna oczekiwana brama. W pseudokodzie nie ma instrukcji `case`, ponieważ kolejność działań jest ustalona.


```
//step 1
move-to-goal(n, dir) in direction of next foyer
//step 2
wallfollow until next gateway is detected
```

Zdobyta wiedza

Zespół CSM nie zajął miejsca w konkursie AAAI; robot doznał szeregu awarii sprzętowych, które spowodowały awarię zarówno zasilania, jak i sonarów. Robot został ostatecznie naprawiony, a oprogramowanie działało dobrze. Testy przed i po zawodach dały kilka praktycznych lekcji. Po pierwsze, bardzo ważne jest zbudowanie abstrakcyjnych zachowań nawigacji z solidnych prymitywów. Największy problem z implementacją nie dotyczył samych skryptów, ale raczej jakości prymitywnych zachowań. Jeśli robot nie może wiarygodnie wykryć otwartych drzwi, gdy podąża za ścianą wokół pokoju, nigdy nie wyjdzie bez względu na to, co mówi skrypt. Po drugie, rywalizacja o zasoby detekcyjne to pojawiający się problem w schematach sterowania robotami. Kamera wideo w robocie nie miała mechanizmu przesuwania; zasadniczo sterowanie kierunkowe Clementine to efektor kamery. W jednym przypadku zachowania korytarza nawigacyjnego między węzłem H a węzłem Hd nominalne zachowanie hallolow często wskazywało Clementine i jej kamerę z dala od przewidywanego położenia drzwi, zakłócając zachowanie percepcyjne potwierdzania drzwi używane do zakończenia nawigacji - sala. Nawet przy niskich prędkościach Clementine często mijała drzwi. Jedną z alternatyw jest umożliwienie dwóm aktywnym zachowaniom na zmianę kontrolowania robota. Prowadzi to do scenariusza ruchu, zatrzymania, wyczuwania, który spowolnił postęp robota i nadal pozwalał robotowi przetoczyć się przez drzwi. Należy zauważyć, że zastosowany algorytm reprezentacji i planowania ścieżki obsługuje dodawanie informacji metrycznych. W bazie danych można przechowywać metryczną odległość między każdą odwiedzoną parą węzłów. Klasa węzła przechowuje indeks odniesienia do węzłów podłączonych do N, E, S, W; odległość można również dołączyć do tych wskaźników. Jednym z problemów związanych z prostym dołączaniem odległości metrycznej jest podjęcie decyzji, której wartości odległości użyć. Jeśli wallfollow zabierze robota wokół obwodu foyer w celu wyjścia, przebyta odległość nie będzie odzwierciedlać odległości w linii prostej między węzłami, ani pomiar nie będzie dwukierunkowy; przejście N-S przez foyer może być znacznie dłuższe niż S-N, jeśli foyer jest asymetryczne, a robot za każdym razem podąża za różnymi ścianami. Z drugiej strony, gdyby znana była szerokość foyer, robot mógłby prawdopodobnie użyć martwego liczenia, aby przejść bezpośrednio przez nie do pożądanego wyjścia. Problemem może być również wpływ przeszkód wydłużających przebytą odległość, zwłaszcza jeśli przeszkodą są ludzie poruszający się po hali. Nawigacja po krótkiej hali może zająć dużo czasu, jeśli podczas jednej wizyty jest zatłoczona, a podczas innej jest pusta. Inną trudnością związaną z dodawaniem odległości metrycznych jest określenie, jak efektywnie wykorzystywać odległości. Odległości metryczne mogą nie być znane dla wszystkich par węzłów, co utrudnia zastosowanie algorytmu Dijkstry. Podobnie odległość może być tylko jednym z czynników przy wyborze trasy; nasza obecna realizacja preferuje przechodzenie przez hale, a nie chodzenie na skróty między pomieszczeniami. Teoria użyteczności jest jednym z mechanizmów ilościowego określania wpływu tych konkurujących ze sobą problemów.

Omówienie możliwości AI

Podjęcia robotyki AI do nawigacji koncentrowały się na planowaniu ścieżek oraz jednoczesnej lokalizacji i mapowaniu. Tematy te w dużej mierze opierają się na obszarach sztucznej inteligencji związanych z reprezentacją wiedzy, wyszukiwaniem i planowaniem. Metody topologiczne, zwłaszcza metody asocjacyjne, są często podzbiorem badań wizji komputerowej. Metody topologiczne są zasadniczo metodami behawioralnymi i cierpią na ten sam problem braku ogólnej odporności; jeśli

metoda zawiedzie, trudno jest wykryć i zastosować techniki rozwiązywania problemów. Inne obszary sztucznej inteligencji zazwyczaj nie są włączane do implementacji nawigacji topologicznej. Kluczową reprezentacją wiedzy wykorzystywaną w planowaniu ścieżek topologicznych jest pewna forma pamięci przestrzennej. Paradigmat reaktywności inspirowane podejście topologiczne „mniej znaczy więcej”, które minimalizuje pamięć przestrzenną i etykiety semantyczne stosowane do bram i punktów orientacyjnych. Ilość pamięci przestrzennej, jakiej potrzebuje agent, aby nawigować, jest szarą strefą i zależy od wielu czynników. Jak dokładnie i wydajnie robot musi się poruszać? Czy czas jest krytyczny, czy może to być nieco nieoptymalna trasa? Zadania nawigacyjne, które wymagają optymalności, zwykle wymagają bardziej gęstych i złożonych reprezentacji świata. Jakie są cechy środowiska? Czy istnieją punkty orientacyjne, które zapewniają wskazówki orientacyjne? Czy odległości między punktami orientacyjnymi są dokładnie znane? Jakie są źródła informacji o tym środowisku, które określają teren, właściwości nawierzchni, przeszkody itp.? Jakie są właściwości czujników dostępnych w tym środowisku? Nawigacja topologiczna obejmuje generowanie ścieżki z wyborem i implementacją procedur do wykonania nawigacji wzdłuż ścieżki. W metodach relacyjnych model świata składa się z odrębnych miejsc połączonych lokalnymi strategiami kontroli, które zasadniczo są zachowaniami. Historycznie rzecz biorąc, wyszukiwanie ścieżek nie wymagało skomplikowanych algorytmów wyszukiwania, ponieważ sieć wyróżniających się miejsc jest na tyle rzadka, że może korzystać ze standardowych algorytmów grafowych. W metodach asocjacyjnych model świata składa się z ramek widokowych, które mają powiązany kierunek poruszania się, lub kombinacji punktów orientacyjnych, z których skojarzony kierunek można zapamiętać dla wcześniej napotkanych ścieżek lub eksploracji środowiska lub wywnioskować dla nowych celów. Nawigacja topologiczna zakłada, że percepcja, zwłaszcza widzenie komputerowe, jest wystarczająca do identyfikacji i uczenia się punktów orientacyjnych i bram, chociaż metody te starają się ominąć problem uziemienia symboli. Charakterystyczne miejsca „rozpoznają” miejsce, ale to rozpoznanie jest bliższe afordancji niż rzeczywistemu rozpoznaniu, powracając do dwóch typów percepcji Neissera. Metody asocjacyjne są interesujące, ponieważ były zorientowane na wizję komputerową. We wszystkich przypadkach motywem przewodnim było to, że agent buduje model świata, badając środowisko i ucząc się tego, co jest wyjątkowe (np. charakterystyczne miejsca, punkty orientacyjne, punkty widzenia) oraz zachowań (np. lokalne strategie kontroli, kierunek ruchu). W praktyce praktyczne wyzwania niezawodnej percepcji i wykonania behawioralnego miały tendencję do przytłaczania implementacji topologicznych, a zatem rozpoczęły się od mapy a priori. Podczas gdy metody topologiczne wykorzystują planowanie, mniejszy nacisk kładzie się na aspekt rozwiązywania problemów w planowaniu. Jednym z powodów może być to, że bez informacji metrycznych, na przykład, ile czasu zajmuje dotarcie z jednego miejsca do drugiego, trudno jest monitorować postępy. W zawodach AAAI i innych wydarzeniach rozgrywanych w pomieszczeniach,

środowisko było wystarczająco bogate w bramki, aby robot mógł szybko wykryć, że posunął się za daleko i przegapił wskazówkę. Na wiejskich drogach robot mógłby jeździć na duże odległości, zanim zorientowałby się, że nie jest na właściwej ścieżce, jeśli nie ma GPS lub innej formy lokalizacji metrycznej.

Podsumowanie

W tym rozdziale podjęto próbę omówienia: Czym jest nawigacja? w zakresie sztucznej inteligencji. Nawigację można podzielić na cztery funkcje lub pytania, z których każde wymaga inteligencji. Pierwsza to: dokąd jadę? uwzględniony w planowaniu misji. Po drugie: Jaki jest najlepszy sposób? lub problem planowania tras, który zazwyczaj jest synonimem nawigacji. Gdzie ja byłem? i gdzie jestem? stanowią podstawę dla pola poszukiwań i jednoczesnej lokalizacji i mapowania. Wracając do pytań, które motywowały tą część, Jak nawigują zwierzęta? można ogólnie odpowiedzieć jako: Nawigują

topologicznie. Należy jednak zauważyć, że kilka zwierząt używa liczenia martwych, aby odnaleźć drogę do domu. Jednak nawigacja to obszar, w którym ludzie radykalnie odbiegają od zwierząt. Projektujemy nasze środowisko za pomocą punktów orientacyjnych, takich jak znaki i dodatkowe informacje, takie jak znaczniki mil, a także używamy GPS, aby uprościć nawigację. Część udzieliła częściowo odpowiedzi: Czy istnieją różne rodzaje nawigacji? Istnieją dwie szerokie klasy nawigacji, topologiczna i metryczna, ale, jak widać z pytań funkcjonalnych nawigacji, nawigacja może być również postrzegana jako posiadająca cztery ogniska. Jednym z celów jest planowanie ścieżki, które opiera się na mapach a priori lub pamięci przestrzennej, a innym ważnym celem pracy jest eksploracja i tworzenie map w celu wygenerowania map. Aspekt planowania misji w nawigacji lub wyznaczanie celów, dokąd się udać, opisany w poprzednim rozdziale, nie był głównym przedmiotem pracy w robotyce. Założenie było takie, że osoba (lub jakiś inny agent) wyznaczy cele, zaprogramowane lub komunikowane przez język naturalny lub przez wskazywanie (gesty deityczne). Odpowiedź na jaki rodzaj nawigacji jest najlepszy? zależy od aplikacji. Nawigacja topologiczna dobrze nadaje się do sytuacji, w których wystarczające są reaktywne style pamięci przestrzennej i implementacje behawioralne. W rozdziale zagłębiono się w metody nawigacji topologicznej. Nawigacja topologiczna, zwana także trasą lub nawigacją jakościową, jest często postrzegana jako prostsza i bardziej naturalna dla robota opartego na zachowaniu. Z pewnością ludzie często dają innym trasy jako wytyczne; dlatego wydaje się naturalne, że robot będzie w stanie analizować polecenia, takie jak „idź korytarzem, skręć w lewo w ślepych zaułku i wejdź do drugiego pokoju po prawej”. Nawet bez mapy, gdzie wszystko się znajduje, jest wystarczająco dużo informacji do nawigacji, o ile robot wie, czym jest „hala”, „ślepa uliczka” i „pokój”. Nawigacja topologiczna w robotach opiera się na punktach orientacyjnych. Punkty orientacyjne upraszczają pytanie Gdzie jestem? problem, dostarczając wskazówki dotyczące orientacji. Bramy są szczególnym przypadkiem punktów orientacyjnych, które odzwierciedlają potencjał robota do zmiany kierunku (skręcenie inną drogą lub korytarzem, wejście do pokoju itp.). Istnieją dwie kategorie jakościowych metod nawigacji: relacyjne i asocjacyjne. Metody relacyjne łączą ze sobą charakterystyczne sznurki (węzły) za pomocą lokalnych strategii sterowania (krawędzi) lub zachowań potrzebnych do przemieszczania się między węzłami tworzących graf. Robot może wykorzystać wykres do planowania ścieżki przy użyciu technik, takich jak algorytm najkrótszej ścieżki z jednego źródła. Wykonuje ścieżkę, wykorzystując zachowanie związane z krawędzią, którą przemierza. Gdy widzi punkt orientacyjny lokalizacji, znajduje się w sąsiedztwie, a następnie może użyć innego zachowania, na przykład wspinania się po wzgórzach, aby zlokalizować się względem punktu orientacyjnego. Metody asocjacyjne wiążą percepcję z lokalizacją albo poprzez zapamiętanie wizualnego wzorca zmian, które umożliwiają robotowi dotarcie do miejsca, albo poprzez wywnioskowanie lokalizacji z układu punktów orientacyjnych. Metody asocjacyjne są lepsze niż inne metody tras do odtwarzania znanych ścieżek. Żeby zrobić kiepski kalambur, nawigacja topologiczna tylko zarysowuje powierzchnię nawigacji i sztucznej inteligencji.