NAWIGACJA ROBOTOW HUMANOIDALNYCH

WYKŁAD 7

Nawigacja spoleczna:

tlumy, normy kulturowe, interakcja z czlowiekiem



dr inż. Mateusz Pomianek

Modele tlumu

Proksemika

Normy kulturowe

HRI / personalizacja

Plan wykladu

Sekcja 1

Modele ruchu tłumu: Social Force Model, ORCA, modele oparte na danych, predykcja trajektorii

Sekcja 2

Proksemika i normy kulturowe: strefy Halla, różnice kultur, kontekst sytuacyjny, sygnały społeczne

Sekcja 3

Personalizacja interakcji człowiek-robot: profil użytkownika, uczenie preferencji, komunikacja, ewaluacja

Kody Python

Social Force Model, ORCA, predykcja Transformer, klasyfikator profilu, mapa społecznych kosztów

01

SEKCJA

Modele ruchu tlumu



Social Force Model, ORCA, modele predykcji trajektorii, detekcja grup

Dlaczego nawigacja społeczna jest potrzebna?

- Robot poruszający się tylko wedlug geometrii (A* + DWA) jest postrzegany jako nieprzewidywalny i nieprzyjemny
- **Badania** (Kruse et al., 2013): 70% użytkowników odczuwa dyskomfort, gdy robot nie respektuje nienawigacyjnych norm społecznych
- **Normy społeczne**: nie wchodzić miedzy rozmawiające osoby, nie ścigać pieszych, utrzymywać właściwy dystans
- **Nawigacja społeczna rozszerza klasyczne planowanie o model agentów**: każdy człowiek ma intencje, prędkość i strefę prywatności
- **Standard ISO 13482:2014**: roboty opiekuncze musza spelniac minimalne wymagania dotyczace bezpiecznej interakcji z człowiekiem
- **Cel**: robot postrzegany jako naturalny uczestnik przestrzeni społecznej, a nie przeszkoda czy zagrożenie

Social Force Model (SFM); Helbing i Molnar (1995)

- **Idea**: ruch pieszego modelowany jako suma sił: prędkość docelowa + odpychanie od przeszkód + odpychanie od innych ludzi
- **Siła własna**: $F_i^{self} = (v_i^0 * e_i^0 - v_i) / tau_i$ (powrót do pozadanej prędkosci i kierunku)
- **Siła społeczna** (odpychanie): $F_{ij} = A * exp((r_{ij} - d_{ij})/B) * n_{ij}$ (im blizej, tym silniej)
- **Siła od przeszkód**: analogiczna do społecznej; gradient prostopadły do powierzchni ściany
- **Wynikowa**: $m_i * (dv_i/dt) = F_i^{self} + sum(F_{ij}) + sum(F_{iw})$
- **Zalety**: prosta implementacja, przewidywalne zachowanie, działa w czasie rzeczywistym *(~1 ms / agent)*
- **Wady**: kalibracja parametrów (A, B, tau) wymaga danych pomiarowych; nie modeluje intencji długookresowych

KOD PYTHON

Social Force Model: implementacja dla planisty lokalnego

```python
import numpy as np

def social_force(robot_pos, robot_vel, goal_pos, pedestrians,
                 A=2.0, B=0.3, tau=0.5, v0=1.2):
    """Wyznacza wypadkowa siłę społeczną dla robota."""
    e0  = goal_pos - robot_pos
    e0 /= (np.linalg.norm(e0) + 1e-6)            # kierunek do celu
    # Siła własna: robot chce iść do celu z prędkością v0
    f_self = (v0 * e0 - robot_vel) / tau
    # Siły odpychające od pieszych
    f_social = np.zeros(2)
    for ped in pedestrians:
        diff = robot_pos - ped['pos']            # wektor od pieszego do robota
        dist = np.linalg.norm(diff)
        r_ij = 0.3 + ped.get('radius', 0.25)     # suma promieni
        n_ij = diff / (dist + 1e-6)              # wektor normalny
        f_social += A*np.exp((r_ij-dist)/B)*n_ij
    return f_self + f_social                     # [m/s^2]
```

Opis: Wywoływac co 50 ms; scałkuj siłę przez dt aby uzyskać nową predkość (clamp do v_{max})

ORCA: Optimal Reciprocal Collision Avoidance

Idea ORCA (van den Berg et al., 2011)

- **Kazdy agent** przyjmuje połowę odpowiedzialnosci za unikniecie kolizji
- **Dla kazdej pary (i, j):** wyznaczana jest half-plane bezpiecznych predkosci
- **Nowa predkosc agenta i**: najblizszy punkt do predkosci preferencji w zbiorze bezpiecznych predkosci
- **Warunek**: przy wzajemnym stosowaniu ORCA przez wszystkich agentow brak kolizji jest gwarantowany
- **Złożoność**: O(n) per agent per krok; praktyczne dla n < 500 agentow
- Popularny w symulatorach: CrowdSim, MENGE, PedSim

ORCA dla robota wposrod pieszych

- Robot zna swoja predkosc preferencji $\mathbf{v}_{pref}$ (z DWA lub WMA)
- Dla kazdego wykrytego pieszego wyznacza half-plane ORCA
- **Wynikowa predkość**: projekcja $\mathbf{v}_{pref}$ na przeciecie half-planes
- **Pieszy nie stosuje ORCA**: robot bierze pełną odpowiedzialność
- **Rozszerzenie**: RVO2 library (C++/Python); dziala w czasie rzeczywistym
- **Ograniczenie**: nie przewiduje intencji; zakresla ruch stały

Predykcja trajektorii pieszych

Modele kinematyczne

- **Stałość prędkości (CV):** $x_{t+1} = x_t + v*dt$
- **Stałość przyspieszenia (CA):** uwzględnia inercje
- **Filtr Kalmana**: estymacja + predykcja
- Prosta, szybka, dziala krotkoterminowo
- Zawodzi przy zmianie kierunku lub zatrzymaniu
- **Horyzont**: do ~1 s

Social Force / ORCA

- Model interakcji miedzy agentami
- **Predykcja,** z uwzglednieniem dynamiki tlumu
- Wymaga pozycji wszystkich pieszych
- Średnio skuteczny na krotkim horyzoncie
- Może propagować błędy siły spolecznej
- **Horyzont**: 1–3 s

Modele danych (DL)

- Social LSTM / Social GAN (Alahi 2016)
- Trajectron++ (Salzmann 2021)
- **Transformer**: AgentFormer, Wayformer
- Uwzgledniaja kontekst i interakcje
- Wymagaja treningu na danych (UCY, ETH)
- **Horyzont**: 3–5 s; SoTA na benchmarkach

Detekcja grup i zachowan zbiorowych

- Ludzie czesto poruszaja sie w grupach (dyady, triady, rodziny); robot nie moze wejsc miedzy czonkow grupy
- **Detekcja grupy**: clustering pieszych wedlug wspolnej predkosci i bliskosci; DBSCAN lub hierarchical clustering
- **Formacja grupy**: trojkat, linia, V-ksztalt; kazda ma strefe chroniona miedzy czlonkami (O-space)
- **O-space (Kendon 1990):** przestrzeń miedzy członkami grupy; robot musi ją ominąć, nie przejść na wprost
- **F-formation detection**: detekcja wspolnej orientacji i pola widzenia; robot planuje objazd z zewnatrz
- **Zastosowanie**: robot chcacy podejsc do osoby czeka na rozejscie sie grupy lub sygnalizuje chec dolaczenia
- **Implementacja**: Social NavMesh uwzgledniajacy O-spaces jako dynamiczne przeszkody kosztowe w cost map

KOD PYTHON

Społeczna mapa kosztów i integracja z Nav2

```python
import numpy as np

def build_social_costmap(grid_size, resolution, pedestrians,
                         sigma_pos=0.8, sigma_vel=0.5) -> np.ndarray:
    """Buduje mape kosztow uwzgledniajaca pieszych (Gaussowy potencjal)."""
    cost = np.zeros((grid_size, grid_size), dtype=np.float32)
    cx = cy = grid_size // 2  # srodek mapy = pozycja robota
    for p in pedestrians:
        px = int(cx + p['x'] / resolution)
        py = int(cy + p['y'] / resolution)
        vx, vy = p.get('vx', 0.0), p.get('vy', 0.0)
        for gy in range(grid_size):
            for gx in range(grid_size):
                dx, dy = gx - px, gy - py
                # Wydluzony Gauss w kierunku ruchu (anizotropowy)
                fwd  = dx*vx + dy*vy + 1e-6   # skladowa w przod
                d2   = dx**2/sigma_pos**2 + dy**2/sigma_pos**2
                d2  += max(fwd, 0)**2 / sigma_vel**2
                cost[gy,gx] += 100.0 * np.exp(-0.5*d2)
    return np.clip(cost, 0, 100).astype(np.uint8)
```

Opis: Anizotropowy Gauss: wiekszy koszt z przodu pieszego (w kierunku ruchu) niz z boku lub tylu.

02

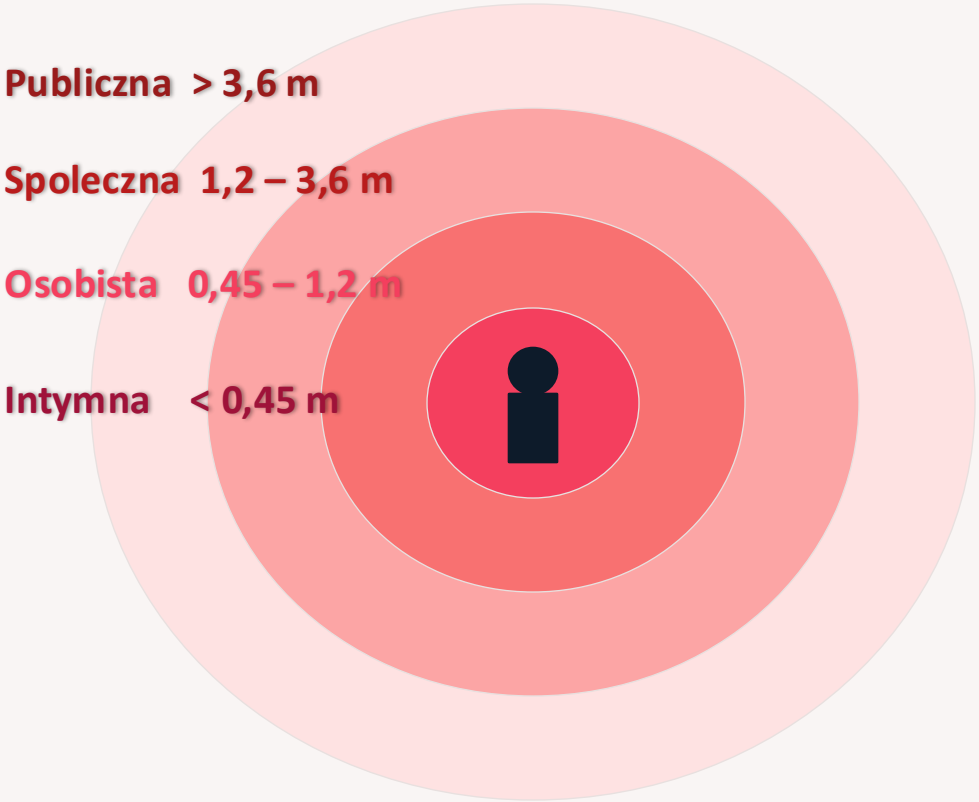
SEKCJA

Proksemika i normy kulturowe

Strefy osobiste, roznice kulturowe, kontekst sytuacyjny, sygnaly spoleczne



Strefy proksemiczne E. Halla: przestrzeń osobista człowieka



Implikacje dla robota

- Robot powinien utrzymywać dystans społeczny (1,2 m) przy podejściu do nieznanej osoby
- **Wejscie w strefę osobistą** wymaga sygnału zaproszenia lub wyraznej zgody użytkownika
- **Strefa intymna**: tylko w przypadku asystencji medycznej lub dostepu do dozownika lekow
- **Nawigacja**: robot omija ludzi z marginesem 0,5 m = strefa bezpieczna powyzej osobistej
- **Monitoring**: czujnik F/T lub detekcja YOLO jako trigger zatrzymania przy nadmiernym zbliżeniu

Różnice kulturowe w przestrzeni osobistej

- **Kultury kontaktowe** (Ameryka Poludniowa, Bliski Wschod, Europa Poludniowa): dystans konwersacyjny 0,5–0,9 m
- **Kultury zdystansowane** (Europa Polnocna, Japonia, USA): dystans konwersacyjny 0,9–1,5 m
- **Kultury Azji Wschodniej**: wyjątkowo duży dystans w kontekscie formalnym; mały dystans w zatłoczonej przestrzeni publicznej
- **Robot wspolpracujacy globalnie** musi adaptowac parametry stref do kontekstu kulturowego uzytkownika
- **Dane badawcze**: Hall (1966), Beaulieu (2004), meta-analiza Sorokowska et al. (2017) na 42 krajach
- **Implementacja**: baza danych norm kulturowych + modul detekcji jezyka / regionu z konta uzytkownika
- **Ostrozność**: stereotypizacja kulturowa jest ryzykowna; preferencje indywidualne maja priorytet nad normami grupowymi

Kontekst sytuacyjny, a normy nawigacyjne

Środowiska o wysokiej gestości *(public)*

- **Dworce, lotniska, centra handlowe**: tlum > 1 os/m^2
- **Akceptowalny kontakt fizyczny** (ramie w ramie), norma kulturowa
- **Robot**: tryb zwarty; waska trasa, mniejszy dystans bezpieczny (0,4 m)
- **Predkosc zredukowana** do max 0,6 m/s; czeste zatrzymania
- **Priorytet**: przepuszczanie grup idacych w tym samym kierunku
- **Sygnalizacja**: oswietlenie LED i dzwiek przy zblizaniu

Środowiska opieki *(szpital, dom opieki)*

- **Pacjenci, osoby starsze, dzieci**: wzmocniona strefa bezpieczna (+50%)
- **Predkosc** max 0,4 m/s; plynne przyspieszenia i hamowania
- **Brak naglych ruchow**; robot nie powinien pojawisc sie od tylu
- **Zachowanie przewidywalne** i komunikowane z wyprzedzeniem
- **Pierwszenstwo** dla personelu medycznego niezaleznie od sciezki
- **Dzwiek sygnalizacyjny**: male natezenie, przyjazny ton

Sygnały społeczne: co robot musi rozpoznawać

Gesty i mowa ciala

Wyciagnieta reka = stop; kiwnecie glowa = zgoda;
odwrocenie sie = koniec interakcji.
Detekcja: MediaPipe Pose

Kontakt wzrokowy

Wzrok skierowany na robota = oczekiwanie interakcji;
brak kontaktu = nie przeszkadzac.
Detekcja: gaze estimation

Mowa i prosodia

Komenda glosowa; ton i tempo sygnalizuja stan
emocjonalny. ASR + klasyfikator sentymentu
w czasie rzeczywistym

Orientacja ciala

Zwrot frontem do robota = zaproszenie;
bok lub tyl = brak zainteresowania.
Kluczowe dla decyzji o podejsciu

Ekspresja twarzy

Dezorientacja, niezadowolenie, panika.
Detekcja: modele AU (Action Units)
lub end-to-end (AffectNet, FER2013)

Czynniki sytuacyjne

Osoba smieciaca z tornistrem: zajeta.
Osoba stojaca i patrzaca: czeka.
Kontekst wnioskowany z polozenia i gestow

Kodowanie norm społecznych w mapie kosztów

- **Warstwa geometryczna**: klasyczne przeszkody statyczne (sciany, meble) z inflacja 0,3 m
- **Warstwa pieszych**: anizotropowy Gauss wokol kazdej osoby; wiekszy koszt w kierunku ruchu i z przodu twarzy
- **Warstwa grup**: O-space miedzy czlonkami grupy kodowany jako strefa zakazana (koszt = 100)
- **Warstwa konwersacji**: obszar miedzy rozmawiajacymi osobami; robot nie moze go przeciac
- **Warstwa kulturowa**: globalny mnoznik kosztow (x1.0 – x1.5) zaleznie od profilu kulturowego srodowiska
- **Warstwa kontekstowa**: szpital: +30% kosztu wszedzie; ruchliwy korytarz biurowy: -10% kosztow bocznych
- **Aktualizacja**: wszystkie warstwy odswiezane co 100 ms na podstawie detekcji pieszych z LiDAR + kamery

03

SEKCJA

Personalizacja interakcji czlowiek-robot (HRI)



Profil uzytkownika, uczenie preferencji, komunikacja, ewaluacja

Czym jest personalizacja w nawigacji spolecznej?

Robot generyczny zachowuje sie tak samo wobec kazdego ⟷ **Robot spersonalizowany** adaptuje zachowanie do konkretnej osoby

Wymiary personalizacji: dystans fizyczny, styl komunikacji, prędkosc ruchu, poziom proaktywnosci

- **Użytkownik 1**: osoba starsza z trudnosciami w chodzeniu; oczekuje wolnego robota i wyraznych sygnalow
- **Użytkownik 2**: pracownik biura; oczekuje robota ktory nie przeszkadza i nie blokuje drogi
- **Użytkownik 3**: dziecko; robot powinien byc bardziej ostrozosc i uzywac prostszych komunikatow

Cel: minimalizacja zaklocen i maksymalizacja poczucia komfortu konkretnej osoby, nie sredniej uzytkownikow

Metody: profil explict (preferencje podane przez uzytkownika) + profil implicit (wnioskowany z zachowania)

Profil użytkownika: składowe i źródła

Cechy stale (explicit)

- **Wiek i mobilność** (poruszanie sie z laską, na wózku)
- **Preferencje kulturowe** (kraj pochodzenia, jezyk)
- **Znane fobie lub lęki** (np. lęki przed maszynami)
- **Preferowany styl komunikacji** (werbalny / niewerbalny)
- **Historia interakcji z robotami** (nowicjusz / doswiadczony)
- **Rola w organizacji** (pacjent, personel, gość)

Cechy dynamiczne (implicit)

- **Aktualny stan emocjonalny** (ekspresja twarzy, tempo mowy)
- **Poziom pospiechu** (prędkość chodzenia, częstość kontaktu)
- **Odpowiedzi na poprzednie zachowania robota** (wycofanie, kiwnięcie)
- **Zajętość** (słucha muzyki, rozmawia przez telefon)
- **Pozycja i orientacja ciala** wzgledem robota
- **Historia dzisiejszej sesji** (akceptacje, odrzucenia propozycji)

Uczenie preferencji uzytkownika w czasie rzeczywistym

- **Cel**: robot aktywnie aktualizuje model preferencji na podstawie sygnalu zwrotnego uzytkownika
- **Sygnaly zwrotne**: wycofanie sie (zbyt blisko), kiwniec (OK), komenda slowa (szybciej / wlnniej), styl chodzenia
- **Regresja Bayesowska**: model dystansu preferowanego d* ~ N(mu, sigma^2); aktualizacja mu i sigma po kazdej obserwacji
- **Inverse RL (IRL):** z obserwacji ludzkich zachowan wnioskuj funkcje nagrody; zastosuj do polityki robota
- **Gaussian Process Regression**: interpolacja preferencji dystansu dla roznych kontekstow (korytarz, winda, sala)
- **Prywatnosc**: dane profilowe przechowywane lokalnie; opcja reset profilu przez uzytkownika w dowolnej chwili
- **Cold start**: nowy uzytkownik startuje z domyslnymi parametrami kulturowymi; personalizacja po ~5 minutach interakcji

KOD PYTHON

Bayesowska aktualizacja preferencji dystansu

```python
import numpy as np

class DistancePreferenceModel:
    """Gaussowski prior nad preferowanym dystansem d* [m]."""
    def __init__(self, mu0=1.2, sigma0=0.4):
        self.mu    = mu0       # wartość oczekiwana początkowa
        self.sigma = sigma0    # odchylenie standardowe (niepewność)
        self.obs_sigma = 0.15  # szum pomiaru (gęstość obserwacji)

    def update(self, observed_distance: float, feedback: str):
        """Aktualizacja po obserwacji; feedback: "ok"/"too_close"/"too_far"."""
        if feedback == 'too_close':
            z = observed_distance + 0.3  # użytkownik chce większy dystans
        elif feedback == 'too_far':
            z = observed_distance - 0.2  # użytkownik akceptuje mniejszy
        else:
            z = observed_distance        # OK – to jest d*

        # Kalman update: posterior Gaussian
        K = self.sigma**2 / (self.sigma**2 + self.obs_sigma**2)
        self.mu    += K * (z - self.mu)
        self.sigma  = np.sqrt((1 - K) * self.sigma**2)
        return self.mu, self.sigma
```

Opis: Po 5–10 interakcjach sigma spada do ~0,05 m; mu zbiezna do prawdziwej preferencji uzytkownika

Komunikacja robota z człowiekiem: kanaly

Mowa (TTS)

Synteza mowy: przyjazny glos, odpowiednia glosnosc. Robot informuje o zamiarach: 'Przechodze po lewej stronie'

Swiatla LED

Kolory sygnalizuja stan: zielony = gotowy, zolty = zatrzymanie, czerwony = ostrozenie, niebieski = naladowanie

Gestykulacja

Ramie wskazuje kierunek lub zaprasza; kiwnicie glowy potwierdza. Wymaga platformy z ramionami

Wyswietlacz twarzy

Animowane oczy lub emotikonki: przekazuja stan emocjonalny (zdziwienie, zaproszenie, przepraszam)

Ruch anticipacyjny

Robot zaczyna skrecac wczesniej niz wymagane, sygnalizujac zamiar. Badania: anticipatory motion zmniejsza konflikty o 40%

Personalizacja głosu

Ton, tempo i jezyk dopasowane do profilu uzytkownika. Dziecko: wolniej, prostsze slowa. Senior: wieksze natezenie

Pipeline personalizacji HRI: od kontaktu do adaptacji

1

Identyfikacja uzytkownika

Rozpoznawanie twarzy / tokenu RFID / konta; ladowanie lub tworzenie profilu uzytkownika

2

Ocena kontekstu

Środowisko (szpital/biuro), godzina dnia, tlok; wybor bazowej polityki nawigacyjnej

3

Inicjacja interakcji

Kontakt wzrokowy, kiwnięc lub glos; robot sygnalizuje gotowość do interakcji i zamiar ruchu

4

Wykonanie z monitoringiem

Nawigacja z aktywna obserwacja sygnalizacji zwrotnej; rejestracja odleglosci, reakcji ciala, slow

Ewaluacja nawigacji społecznej: metryki i benchmarki

- **Intrusion into Personal Space (IPS):** procent czasu, gdy robot narusza strefe osobista (< 1,2 m) bez zaproszenia
- **Path Irregularity (PI):** odchylenie trasy robota od trajektorii bez pieszych; wskaznik zaklocen w nawigacji
- **Social Space Violation (SSV):** liczba wejsc w O-space grupy lub strefe konwersacji
- **Time-to-pass (TTP):** czas potrzebny robotowi do mineciia pieszego; im krotszy tym mniej ograniczen
- **Perceived Safety Rating (PSR):** subiektywna ocena bezpieczenstwa przez pieszych (kwestionariusz Likerta)
- **Comfort Score (CS):** srednio oceniana wygoda interakcji; zbierany przez aplikacje mobilna lub kiosk
- **Benchmarki**: SocNav1/2 (Manso et al.), SEAN-EP (Tsoi et al.), CrowdNav (Chen et al.); porownywalne wyniki

KOD PYTHON

Klasyfikator profilu uzytkownika i adaptacja parametrow

```python
import numpy as np

PROFILE_PARAMS = {
    'senior':  {'v_max': 0.4, 'dist_safe': 1.5, 'tts_speed': 0.8},
    'child':   {'v_max': 0.5, 'dist_safe': 1.8, 'tts_speed': 0.9},
    'adult':   {'v_max': 0.8, 'dist_safe': 1.2, 'tts_speed': 1.0},
    'worker':  {'v_max': 1.0, 'dist_safe': 1.0, 'tts_speed': 1.1},
}

def classify_user(face_age: float, walking_speed: float,
                  uses_aid: bool) -> str:
    """Prosty klasyfikator profilu uzytkownika."""
    if uses_aid or face_age > 65:
        return 'senior'
    if face_age < 14:
        return 'child'
    if walking_speed > 1.2:
        return 'worker'
    return 'adult'

def get_nav_params(profile: str, pref_model) -> dict:
    base = PROFILE_PARAMS[profile].copy()
    # Nadpisz dystans z modelu Bayesowskiego (z Kodu 3)
    base['dist_safe'] = pref_model.mu
    return base
```

Opis: Klasyfikacja wstepna z szybkich cech; model Bayesowski (Kod 3) nastepnie personalizuje *dist_safe* online

Studium przypadku: robot w szpitalu wielokulturowym

- **Platforma**: robot usługowy na korytarzach szpitalnych; personel z 12 krajów, pacjenci różnych kultur i grup wiekowych
- **System detekcji**: YOLO dla pieszych + MediaPipe Pose dla orientacji + gaze estimation dla kontaktu wzrokowego
- **Profil domyślny**: strefa 1,5 m, prędkość max 0,5 m/s, komunikacja głosowa po polsku lub angielsku
- **Adaptacja kulturowa**: detekcja jezyka (ASR) uaktywnia moduł proksemiki (+20% dystansu dla kultur nordyckich)
- **Adaptacja do seniora**: twarz + laska / wózek -> tryb senior: v_max = 0,4 m/s, dystans 1,8 m, glos wolniejszy
- **Adaptacja do dziecka**: wiek < 14 z rozpoznania twarzy -> tryb dziecko: jeszcze większy dystans, komunikacja prostsza
- **Wyniki**: IPS zredukowane o 62% względem klasycznej nawigacji, PSR wzrost o 1,8 pkt (skala 1–7), TTP o 8% dluzszy

Wyzwania i kierunki badań w nawigacji społecznej

Wyzwania techniczne i spoleczne

- **Niezawodność detekcji emocji i gestów** w różnych warunkach oświetlenia i katach widzenia
- **Translacja norm kulturowych** na parametry numeryczne bez stereotypizacji
- **Zimny start nowego użytkownika**: brak danych historycznych utrudnia personalizacje
- **Konflikty norm**: jedna osoba prosi o szybszy ruch, druga w tej samej chwili o zwolnienie
- **Prywatność:** biometryczna identyfikacja użytkownika wymaga zgody i zasad RODO
- **Obliczenia w czasie rzeczywistym**: wielowarstwowa mapa kosztów społecznych musi być odświeżana < 100 ms

Kierunki badan (2024-2026)

- **Foundation models dla HRI**: VLM jako zrozumialy interpreter sygnałow spolecznych
- **Uczenie przez demonstracje**: robot uczy sie norm z filmów zachowań pieszych bez ręcznie zdefiniowanych reguł
- **Teorie umyslu (ToM):** robot wnioskuje o intencjach i stanach psychicznych ludzi z obserwacji
- **Swoj tlum vs obcy tlum**: wielorobotowe systemy spolecznej nawigacji z konsensusem norm grupowych
- **Ewaluacja dlugookresowa**: badania wielomiesięczne w szpitalach i biurach; mierzenie zaufania w czasie

PODSUMOWANIE

01 **Social Force Model i ORCA** daja deterministyczne podstawy unikania tlumu; modele danych (Transformer) przewiduja intencje.

02 **Proksemika Halla** definiuje strefy, ktore robot musi szanowac; naruszenie strefy osobistej niszczy zaufanie uzytkowników.

03 **Normy kulturowe** istotnie roznia sie w skali global; robot dzialajacy w wielu krajach wymaga modulu adaptacji kontekstu kulturowego.

04 **Personalizacja HRI** opiera sie na profilu uzytkownika, uczeniu online preferencji i wielokanalowej komunikacji niewerbalnej.

05 **Ewaluacja nawigacji spolecznej** wymaga metryk jakosci HRI (ITT, PSR, PFC) obok klasycznych wskaznikow nawigacji (RMSE, sukces).

Zadanie projektowe i pytania problemowe

1. Dlaczego SFM moze generowac oscylacje w waskim korytarzu?
   Zaproponuj modyfikacje parametrow lub dodatkowe ograniczenie, ktore usuwa ten problem.
2. Robot wchodzi miedzy dwie osoby rozmawiajace (O-space).
   Jakie sygnaly zachowania powinny temu zapobiec i jak je zakodowac w mapie kosztow?
3. Uzytkownik z Japonii oczekuje dystansu 1,5 m, a uzytkownik z Brazylii 0,7 m.
   Jak w jednym systemie obs luzyc obie preferencje bez recznie wprowadzonych warunkow?

Zadanie: zaimplementuj klase SocialCostmap z anizotropowym Gaussem (kod 2) i przetestuj ją na symulowanej scenie z 5 pieszymi poruszajacymi sie w roznych kierunkach. Zwizualizuj wynikowa mape kosztow i sprawdz, ze planista A* omija ludzi z boku, nie z przodu.

Bonus: zintegruj DistancePreferenceModel (kod 3) z Nav2 jako dynamiczny mnoznik strefy inflacji i pokaz zbieznosc preferencji dla 3 roznych profilow uzytkownikow w symulacji.