

```

import numba
import numpy as np

@numba.jit(nopython=True, nogil=True)
def agent_distance(agent, i, j, x_tilde, d_ij):
    """Three circles model"""
    # TODO: Pre-compute tangents
    t_i = np.array((-np.sin(agent.angle[i]), np.cos(agent.angle[i])))
    t_j = np.array((-np.sin(agent.angle[j]), np.cos(agent.angle[j])))

    tr_i = agent.radius_torso_shoulders[i] * t_i
    tr_j = agent.radius_torso_shoulders[j] * t_j

    r_i = (agent.radius_torso[i],
            agent.radius_shoulders[i],
            agent.radius_shoulders[i])
    r_j = (agent.radius_torso[j],
            agent.radius_shoulders[j],
            agent.radius_shoulders[j])

    k = (0.0, 1.0, -1.0)
    r = (0.0, 0.0)
    c = (np.zeros(2), np.zeros(2))
    x_rel = np.zeros(2)
    e_ij = np.zeros(2)
    h_min = d_ij
    for ri, k_i in zip(r_i, k):
        for rj, k_j in zip(r_j, k):
            c_i = k_i * tr_i
            c_j = k_j * tr_j
            x = c_i - c_j + x_tilde
            d = np.hypot(x[0], x[1])
            h = d - (ri + rj)
            if h < h_min:
                h_min = h
                r = ri, rj
                c = c_i, c_j
                x_rel = x
                e_ij = x / d

    r_tot = r[0] + r[1]
    r_moment_i = agent.position[i] + c[0] + r[0] * e_ij
    r_moment_j = agent.position[j] + c[1] - r[1] * e_ij

    return x_rel, r_tot, h_min, r_moment_i, r_moment_j

def agent_wall_distance():
    pass

```