

RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

Lietišķo datorsistēmu institūts

Mākslīgā intelekta un sistēmu inženierijas katedra

Arkādijs Sergejevs

bakalaura akadēmiskās studiju programmas "Intelektuālas robotizētas sistēmas"
students, stud. apl. nr. 211RDB320

**PEPPER ROBOTA LOKALIZĀCIJA
UN NAVIGĀCIJA, IZMANTOJOT
DZIĻO MAŠĪNMĀCĪŠANOS**

BAKALaura DARBS

Zinātniskais vadītājs PhD.sc.comp., pētnieks
Ēvalds Urtāns

RĪGA 2024

RĪGAS TEHNISKĀ UNIVERSITĀTE
DATORZINĀTNES UN INFORMĀCIJAS TEHNOLOĢIJAS
FAKULTĀTE

Lietišķo datorsistēmu institūts

Mākslīgā intelekta un sistēmu inženierijas katedra

Bakalaura darba izpildes lapa

Noslēguma darba autors:

students Vārds Uzvārds

(paraksts, datums)

Noslēguma darbs ieteikts aizstāvēšanai:

Zinātniskais vadītājs:

PhD.sc.comp., pētnieks Ēvalds Urtāns

(paraksts, datums)

ANOTĀCIJA

Atslēgvārdi: Pepper, SLAM, mašīnmācīšanās, neironu tīkli
Bakalaura darba tips: Moderno risinājumu izpēte

Mobila robota navigācija sastāv no soļiem kā lokalizācija vidē, šķēršļu atpazīšana un ceļa plānošana atbilstoši mērķim. Autonomam robotam šīs darbības jāspēj veikt patstāvīgi mainīgās vidēs, līdz ar to robotam jāspēj pielāgoties. Sensoru dati var būt trokšņaini, kā arī situācijas, kādās robotam jāspēj pieņemt lēmumu, ir dažādas. Dziļā mašīnmācīšanās piedāvā veidu, kā vispārināt un padarīt noturīgāku pret trokšņiem mobila robota navigāciju.

ABSTRACT

Atslēgvārdi: Pepper, SLAM, machine learning, neural networks
Bachelor thesis type: Research of modern solutions.

The navigation of a mobile robot consists of steps such as environment localization, obstacle recognition, and path planning according to the goal. For an autonomous robot, these actions must be able to be performed independently in variable environments, so the robot needs to be adaptable. Sensor data can be noisy, and the situations in which the robot needs to make decisions are diverse. Deep machine learning offers a way to generalize and make mobile robot navigation more robust against noise.

The bachelor's thesis is composed of 50 pages, including 19 figures, 10 tables and 12 attachments. The list of references and sources includes 49 entries.

SATURS

APZĪMĒJUMU SARAKSTS	6
IEVADS	7
1. SAISTĪTIE PĒTĪJUMI	7
1.1. Navigācija, izmantojot iepriekš izveidotas kartes	7
1.2. Navigācija neizpētītā vidē	9
1.3. Vienlaicīgā lokalizācija un kartēšana	11
1.4. Datos balstīta lokalizācija un navigācija	13
1.5. ROS nozīme robotizētu sistēmu izstrādē	15
1.6. Pepper robota platforma	17
2. METODOLOĢIJA	20
2.1. Datu kopas	20
2.2. Rādītāji	20
2.3. Salīdzināšanas protokols	20
3. REZULTĀTI	21
3.1. Some first subsection	21
3.2. Some second subsection	24
4. TĀLĀKIE PĒTĪJUMI	26
5. SECINĀJUMI	26
IZMANTOTIE INFORMĀCIJAS AVOTI	27
PIELIKUMI	29

APZĪMĒJUMU SARAKSTS

SLAM - simultaneous localization and mapping

CNN - Convolutional Neural Network

IEVADS

1. SAISTĪTIE PĒTĪJUMI

Lorem ipsum...

1.1. Navigācija, izmantojot iepriekš izveidotas kartes

Klasiski robotu navigāciju realizē, izmantojot iepriekš izveidotas kartes. Kartes darbojas kā atskaites sistēma, kas palīdz robotam sevi lokalizēt un noteikt iespējamās braukšanas trajektorijas. Kartes iegūšanai tiek izmantota vienlaicīgā lokalizācija un kartēšana jeb SLAM. Izmantojot SLAM, tiek salīdzināti esošā laika stāvokļa sensoru mērījumi ar iepriekšējiem, rezultātā lokalizējot robotu un iegūstot karti, kas var tikt izmantota navigācijai. Darbā (Leonard & Durrant-Whyte, 1991) SLAM veic, izmantojot ar skaņas sensoriem iegūtas ģeometriskas īpašības kā taisnes, plaknes un cilindriskas virsmas, lai uzlabotu lokalizāciju. Pastāv arī SLAM risinājumi kā ORB-SLAM (Mur-Artal, Montiel et al., 2015), kas salīdzina vizuāli izgūtas īpašības, taču ģeometriskā informācija tāpat ir noderīga, lai veidotu kartes ar informāciju par atļauto pārvietošanās apgabalu.

Klasiski karti attēlo kā aizņemtības režģi (Thrun, Burgard et al., 2005). Nosakot robota nobraukto trajektoriju ar SLAM vai ar citu ārēju metožu palīdzību, iespējams izveidot aizņemtības režģi, kas ļauj izplānot robota trajektoriju, balstoties uz atļauto pārvietošanās apgabalu. Atkarībā no vides, kādā robots pārvietojas, un kustības veida izvēlas, cik dimensijās tiks aprakstīts aizņemtības režģis. Kustībai pa līdzenu virsmu ar robotu, kura sensori ievāc informāciju tikai par daļu pārvietojuma telpas, visbiežāk izmanto divdimensionālus aizņemtības režģus. Aizņemtības režģis sastāv no šūnām, kurai katrai piešķir vērtību no 0 līdz 1, kas apzīmē varbūtību, ka tajā atrodas šķērslis. Grāmatā “Probabilistic Robotics” (Thrun, Burgard et al., 2005) izmanto Beijesa filtru, lai noteiktu šūnu aizņemtības varbūtības.

(a)

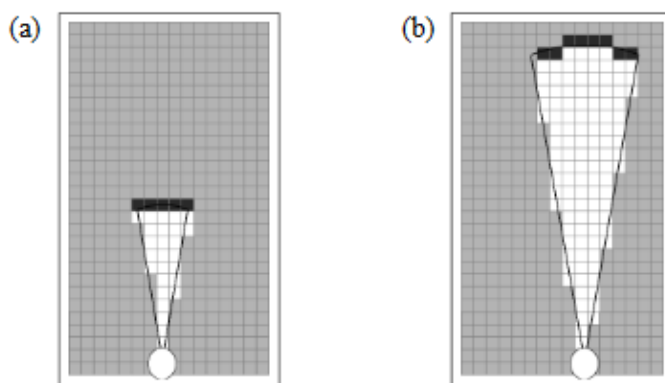


(b)



1. att. (a) Aizņemtības režģis, (b) atbilstošais plānojums (aizgūts no (thrun2005probabilistic))

Aizņemtības režģis tiek izmantots kā karte, kuru izmanto, lai aprēķinātu trajektoriju līdz noteiktajam mērķim, taču navigācijas laikā var tikt sastapti jauni šķēršļi, kuri netika pievienoti oriģinālajam režģim. Šim nolūkam tiek veidoti atsevišķi lokāli aizņemtības režģi, kurus pielieto, lai izmainītu robota trajektoriju un izvairītos no jaunajiem šķēršļiem.



2. att. Aizņemtības režģi divu dažādu attālumu sensoriem (šūnu tumšums apzīmē aizņemtību) (aizgūts no (Thrun, Burgard et al., 2005))

Karšu veidošana pirms navigācijas atvieglo procesu, jo tiek samazināts mainīgo skaits, kurus jāņem vērā, veicot navigāciju. Vairumā gadījumu robota navigācija norit atkārtoti nemainīgā pārvietošanās apgabalā, līdz ar to vienreizēja kartēšana kļūst izdevīga, lai samazinātu navigācijas sarežģītību. Izveidojot karti pirms navigācijas, iespējams arī to pielabot, tādējādi uzlabojot navigāciju kopumā. Ievāktos sensoru mērījumus iespējams apstrādāt ar dažādiem SLAM algoritmiem, izmantojot dažādus parametrus, lai noteiktu visprecīzāko metodi, kā izveidot karti, kuru izmantot navigācijai.

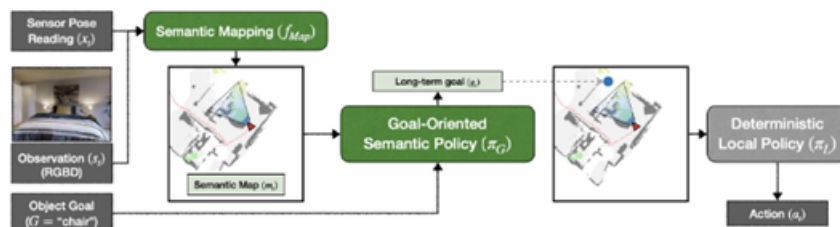
1.2. Navigācija neizpētītā vidē

Pastāv arī pētījumi, kuros tiek aplūkota navigācija iepriekš neizpētītā vidē. Atšķirībā no klasiskās pieejas robotu navigācijai nav zināma vide un atļautais braukšanas apgabals, līdz ar to nav iespējams navigācijas uzsākšanas brīdī izplānot trajektoriju līdz mērķim. Šāda veida navigācijā ir nepieciešams vienlaikus noteikt savu atrašanos vidē, kā arī pieņemt stratēģiskus lēmumus par nākamajiem soļiem, lai maksimizētu iespēju nokļūt līdz mērķim. Pastāv dažādi veidi, kā to risina, taču bieži tiek izmantotas dziļās mašīnmācīšanās metodes, lai realizētu nākamo darbību izvēli. Lai gan klasiskās metodēs (Leonard & Durrant-Whyte, 1991) ģeometriskas īpašības, kuras izmanto karšu veidošanā, var tikt definētas ar izraudzītām formulām, iespējamo navigācijas lēmumu skaits ir ļoti liels, līdz ar to aptvert tos ar izraudzītām formulām ir sarežģīti. Dziļās mašīnmācīšanās metodes ir noturīgākas jaunās vidēs, tādēļ tiek plaši izmantotas šāda veida navigācijas realizācijā.

Darbā (Cimurs, Suh et al., 2021) sadala navigāciju globāla mērķa izvēlē un lokālā navigācijas stratēģijā. Izmantojot ar heuristisku metodi noteiktus globālā mērķa punktus un realizējot navigāciju starp punktiem ar imitācijas vidē apmācītu dziļās mašīnmācīšanās modeli, tiek risināta lokālā optimuma problēma, kas rodas, mācot modelim pieņemt lēmumus par visu trajektoriju, kuru ietekmē daudz faktoru, tai skaitā nenoteiktība. Pilnu trajektoriju no atrašanās vietas līdz tālam punktam kartē, kas nav iepriekš izpētīta, ir sarežģīti iemācīt modelim, jo

trajektorija var būt atkarīga no dažādiem konkrētajā laika brīdī robotam nezināmiem faktoriem. Nosakot globālo mērķi heuristiski – izvēloties kādu punktu, kas atrodas pietiekamā attālumā no diviem secīgiem lāzeru stariem vai arī ārpus lāzera staru uztveršanas diapazona -, var iegūt punktus, līdz kuriem trajektorijas nenoteiktība nav tik augsta un ir apmācāma ar dziļo mašīnmācīšanos (Cimurs, Suh et al., 2021).

Pētījums (Chaplot, Gandhi et al., 2020) adresē neefektīvo ilgtermiņa plānošanu, izmantojot semantiskas kartes objektu atrašanas uzdevumam. Objekta mērķa navigācijā ir nepieciešams nokļūt līdz objektam, kas definēts ar meklēto klasi, kas nozīmē, ka modelim ir jābūt semantiskai izpratnei par meklētajiem objektiem un vidi. Autoru (Chaplot, Gandhi et al., 2020) piedāvātā metode sastāv no vairākām daļām – divdimensionālas semantiskas kartes iegūšana, globāla mērķa noteikšana, izmantojot neironu tīklu, kā arī lokāla deterministiska navigācija, izmantojot klasisku trajektorijas plānošanas algoritmu. Semantiskā karte tiek iegūta, segmentējot attēlu klasēs, izmantojot Mask R-CNN tīklu, apvienojot izejas vērtības ar dziļuma karti vienotā vokseļu režģī, kuru apstrādā trokšņu novēršanas tīkls, rezultātā iegūstot divdimensionālu, mazāk trokšņainu kartes attēlojumu. Globālā mērķa noteikšana tiek apmācīta imitācijas vidē, izmantojot stimulēto mašīnmācīšanos, savukārt trajektorija līdz mērķim tiek aprēķināta, izmantojot trajektorijas plānošanas algoritmu. Raksts norāda uz kartes pielietojamību objekta mērķa navigācijas uzdevumā. Izmantojot karti, iespējams uzglabāt semantiskas un ģeometriskas zināšanas par karti, kuras neironu tīkls var izmantot, lai noteiktu nākamo punktu, lai sasniegtu objekta mērķi.

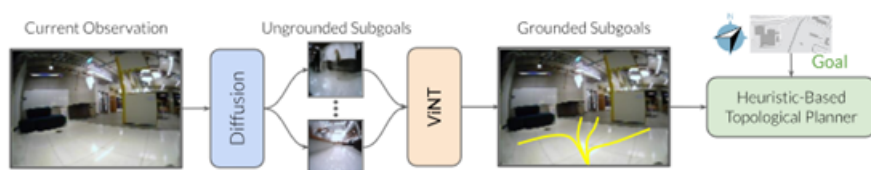


3. att. Objekta mērķa navigācijas metodes shēma (Chaplot, Gandhi et al., 2020)

Darbā (Partsey, Wijmans et al., 2022) aplūko punkta mērķa navigāciju, neizmantojot kartes, bet gan vizuālo odometriju. Zinot patiesās GPS un kompasas vērtības, Gibson-v2 val vidē iespējams veiksmīgi nokļūt līdz punkta mērķim 99.8% gadījumu. Autori (Partsey, Wijmans et al., 2022) aplūko gadījumu, kad patiesās GPS un kompasas vērtības nav zināmas – tās izgūst no vizuāliem datiem. Realizētā metode sastāv no daļām – vizuālās odometrijas un navigācijas modeļa. Izmantojot konvolūciju neironu tīklu, tiek minēta pozīcijas izmaiņa starp diviem attēla un dziļuma kartes pāriem, kā rezultātā tiek atjaunota relatīvā lokācija starp robotu un mērķi. Relatīvā lokācija tiek savienota ar cita konvolūcijas tīkla, kas iekodē esošā laika stāvokļa attēlu un dziļuma karti, izejas vērtībām. Navigācijas modelis izmanto savienotās vērtības, lai minētu nākamās darbības. Pētījumā (Partsey, Wijmans et al., 2022) demonstrē, kā iespējams izmantot vizuālo odometriju, lai realizētu punkta mērķa navigāciju, kā arī apliecina precīzas lokalizācijas nozīmi

robotu navigācijas uzdevumos.

Darbā (Shah, Sridhar et al., 2023) piedāvā pamata modeli ViNT vizuālai navigācijai, kurš apmācīts ar vizuāliem un odometrijas datiem no dažādām robotu platformām. Pamatā tiek aplūkota attēla mērķa navigācija. Izmantojot difūzijas modeli, kurš apmācīts veidot ticamus nākamā mērķa attēlus, iegūst izlasi ar attēliem. Ar navigācijas modeli tiek noteikta trajektorija normalizētās vienībās līdz katram no mērķa attēliem. Lēmums, kuru trajektoriju izvēlēties, tiek pieņemts, balstoties uz heuristisku plānotāju, kurš izmanto topoloģisku grafu ar attēliem kā mezgliem un trajektorijām kā šķautnēm.



4. att. ViNT navigācijas metodes shēma (aizgūts no (Shah, Sridhar et al., 2023))

Autori (Shah, Sridhar et al., 2023) demonstrē, kā iespējams apvienot datus balstītus dziļās mašīnmācīšanās navigācijas modeļus ar topoloģisku grafu, kas darbojas kā karte, lai izlemtu, kur pārvietoties. Šādai pieejai nav nepieciešams pielietot imitācijas vides, kurām ir raksturīgas pārnese problēmas no imitācijas vides un reālo pasauli.

Precīzākai lokalizācijai tomēr varētu būt nepieciešamas marķētas patiešās lokācijas, ne tikai sensoru odometrija. Tāpat arī garāku trajektoriju minēšanā būtu nepieciešami paraugi ar pārvietošanos pa īsāko trajektoriju, nevis vispārīgas demonstrācijas ar pārvietošanos, jo modelim ir svarīgi ne tikai iemācīties semantiskas īpašības, kas ietekmē pārvietošanos, bet arī pieņemt lēmumus, lai brauktu pa iespējami īsāko trajektoriju.

1.3. Vienlaicīgā lokalizācija un kartēšana

Vienlaicīgā lokalizācija un kartēšana jeb SLAM ir būtiska sastāvdaļa navigācijai, kurā robota lokalizācijai tiek izmantota karte. Lai veiktu pirmo trajektorijas aprēķinu izpētītā vidē, ir nepieciešama karte, kas nosaka braukšanas apgabalu. Lai rekonstruētu ģeometriski pareizu karti, ir nepieciešams zināt, kurās lokācijās ir veikti konkrētie sensoru mērījumi. Lai gan informāciju par robota relatīvo atrašanos var sniegt odometrijas sensori, tie nav gan precīzi, lai uz tiem paļautos. Vide palīdz uzlabot odometrijas minējumus, darbojoties kā atskaites sistēma robotam. SLAM izmanto laika gaitā uzņemtos sensoru mērījumus, lai izveidotu vides karti un vienlaikus lokalizētu robotu tajā.

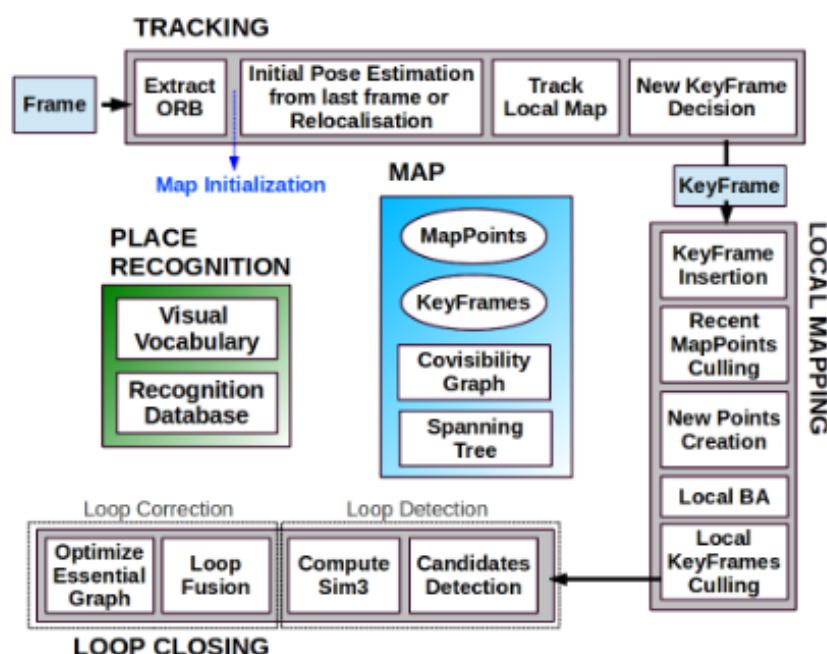
Grāmatā (Thrun, Burgard et al., 2005) iedala SLAM nepārtrauktajā un diskrētajā formā. Nepārtrauktajā formā apraksta objektu kā vizuālu īpašību vai lāzera sensoru apgabalu atrašanos, savukārt diskrētā formā izsaka sakarības starp objektiem. Klasiski SLAM izmanto no sensoru datiem izgūtas īpašības – taisnes, loki un citas figūras attēlos un lāzera datus -, lai uzlabotu pozīcijas izmaiņas, kā arī pašas kartes precizitāti. Autori (Thrun, Burgard et al., 2005) min paplašinātā

Kalmana filtra jeb EKF SLAM kā vēsturiski pirmo un tajā laikā nozīmīgāko. Atšķirībā no oriģinālā Kalmana filtra, kurā stāvokļa mainīgie mainās lineāri, EKF tiek ieviestas nelineāras funkcijas, kas ļauj aprakstīt robota kustību precīzāk. Lai efektīvi izmantotu sensoru mērījumus, kas satur trokšņainus datus, izmanto unikālas īpašības, lai noteiktu pozīcijas izmaiņas. Īpašību skaitu cenšas izvēlēties pietiekami lielu, lai lokalizācija varētu tikt veikta precīzi, taču pietiekami mazu, lai īpašības netiktu sajauktas, kā arī netiktu pārsniegti pieejamie skaitļošanas resursi. Sastopot īpašības atkārtoti, minējumi tiek veikti ar mazāku nenoteiktību, tādējādi piešķirot lielāku nozīmi minējumiem, par kuriem ir lielāka pārliecība (Thrun, Burgard et al., 2005).

Darbā (Zhao, Liu et al., 2022) aplūko citus SLAM veidus kā Gmapping, Hector SLAM, Karto SLAM. Gmapping ir SLAM paveids, kurš balstās uz daļiņu filtriem. Izmantojot odometrijas un lāzeru datus, vispirms tiek noteikta robota trajektorija, pēc tam konstruēta karte. Hector SLAM darbojas, meklējot atbilstības lāzeru datus. Hector SLAM neizmanto odometrijas datus, tikai lāzeru, līdz ar to ir nepieciešams lāzers ar lielu staru skaitu un augstu skenēšanas frekvenci, lai izveidotu precīzu karti. Hector SLAM veido rasterizētas kartes, kuras salīdzina ar veiktajiem skenējumiem, lai noteiktu, kā atjaunot karti. Karto SLAM ir grafa optimizācijā balstīts SLAM. Odometrijas dati tiek izmantoti, lai iegūtu sākotnējo pozīciju. Balstoties uz iegūto pozīciju, tiek meklētas lāzeru datu atbilstības ar karti. Atrastās atbilstības tiek optimizētas, optimizējot robota pozīciju kopumā. Karto SLAM izmanto arī cikla noslēgšanu, kas atjauno grafu, ja ir noteikts, ka robots nokļuvis esošajā pozīcijā atkārtoti, tādējādi uzlabojot iegūto karti. Pētījumā iegūtie rezultāti parāda, ka Gmapping un Karto SLAM pārspēj Hector SLAM robotu platformās, kurām nav augsts lāzera staru blīvums un skenēšanas frekvence. Gan Gmapping, gan Karto SLAM ir noturīgi pret vides izmaiņām, taču Karto SLAM nav nepieciešama tik precīza odometrija kā Gmapping, kā arī tam piemīt cikla noslēgšana, kas var uzlabot kartes precizitāti.

Pastāv arī SLAM veidi, kuri primāri izmanto vizuālus datus. Tos sauc par vizuāliem SLAM. ORB-SLAM (Mur-Artal, Montiel et al., 2015) demonstrē, kā iespējams realizēt SLAM, izmantojot tikai vizuālus datus. Arī ORB-SLAM veic odometriju, meklē atbilstības grafā, kā arī optimizē grafu cikla aizvēšanas gadījumā, taču tiek izmantotas tikai no attēliem izgūtas īpašības. ORB-SLAM sastāv no vairākām daļām – sekošanas (tracking), lokālās kartēšanas un cikla noslēgšanas. Sekošanas daļā tiek noteikta relatīvā pozīcijas izmaiņa salīdzinājumā ar iepriekšējā kadra pozīciju. Kameras pozīciju nosaka, balstoties uz atbilstošo vizuālo īpašību pozīcijas izmaiņām. Ja kameras pozīciju nav iespējams izsekot, veic globālu relokalizāciju, meklējot citus kadrus ar atbilstošajām īpašībām. Tiek meklētas atbilstības starp lokālās kartes kadru un esošā kadra īpašībām, lai optimizētu kameras pozīciju, kā arī noteiktu, vai pievienot kadru lokālajai kartei. Lokālās kartēšanas posmā tiek pievienotas īpašības kartei, ja tās vēl nav pievienotas, tiek veikta grafā esošo kadru pozīcijas optimizācija, kā arī tiek izņemti liekie kadri, kuru īpašības atrodamas citos kadrus, kuri precīzāk savietojas ar pārējiem grafa kadriem. Visbeidzot, tiek veikta cikla aizvēšana, kurā pārbauda, vai kamera nav jau iepriekš atradusies esošajā pozīcijā. Tiek noteikta atbilstība starp esošā kadra un grafā blakus esošajiem grafa kadriem, zemāko izvēlas par sliekšni. Par atbilstošiem neizvēlas kadrus, kuru atbilstība ir zemāka par sliekšni, kā arī blakus esošos. Ja tiek noteikti trīs secīgi kadri, kuru atbilstība pārsniedz sliekšni,

veic cikla aizvēršanu. Tiek izgūta atbilstības transformācija, kas parāda, cik liela ir uzkrātā kļūda. Tiek izņemti punktu dublikāti, kas rodas, veicot transformāciju. Transformācija tiek pielietota pozīcijas atjaunošanai, izplatīta pa pārējiem grafa kadriem.



5. att. ORB-SLAM sistēmas shēma (aizgūts no (Mur-Artal, Montiel et al., 2015))

SLAM metodes pielieto karti, lai uzlabotu lokalizācijas precizitāti. Iz-mantojot tikai odometriju, zūd konteksts par vidi kopumā, kas var noderēt, lai izlabotu robota lokāciju. SLAM gan ļauj ieglabāt informāciju par vidi kartē, ku-ru var izmantot navigācijā, gan arī uzlabo lokalizāciju, sniedzot informāciju par sakarībām starp sensoru mērījumiem un vidi.

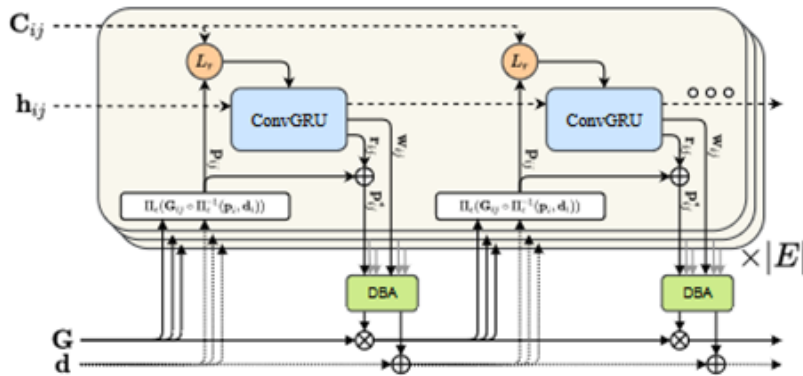
1.4. Dato balstīta lokalizācija un navigācija

Attīstoties dziļās mašīnmācīšanās nozarei un pieaugot pieejamajam datu apjomam, arī vienlaicīgās lokalizācijas un kartēšanas un navigācijas uzdevumos tiek atrasti jauni veidi, kā pielietot dziļo mašīnmācīšanos. SLAM metodes izmanto sensoru lasījumos atrastas īpašības, lai veiktu pozīcijas uzlabošanu vai pat pašu odometriju, kā tas ir ORB-SLAM (Mur-Artal, Montiel et al., 2015).

LIFT-SLAM izmanto neironu tīklu, lai izgūtu vizuālas īpašības, taču pārējo SLAM daļu veic, balstoties uz ORB-SLAM (Bruno & Colombini, 2021). Autori (Bruno & Colombini, 2021) piemin arī DeepVO (S. Wang, Clark et al., 2017) metodi, ar kuru mēģināts odometriju realizēt, apmācot rekurentu konvolū-ciju neironu tīklu noteikt kameras pozīcijas no attēlu sekvenču. Lai gan gala-līdz-galam risinājumi piedāvā atbrīvoties no manuāli veidotām instrukcijām, apmācot neironu tīklu atrast sakarības datus, praksē novērots, ka tās nav tik precīzas kā hibrīda metodes, kuras pielieto gan datus balstītus neironu tīklus, gan atziņas no

klasiskajām metodēm (Bruno & Colombini, 2021). Pastāv arī SuperGlue (Sarlin, DeTone et al., 2020), kura funkcija ir noteikt sakrītošās īpašības starp 2 attēliem, kuriem izgūtas īpašības ar klasiskiem algoritmiem vai neironu tīkliem. SuperGlue realizēts kā grafa neirona tīkls ar uzmanības mehānismu. Ieejā saņemot unikālos punktus ar to aprakstošajām vērtībām, SuperGlue veic uzmanību gan starp katra individuālā attēla punktiem, gan arī starp abu attēlu punktiem, tādējādi nosakot, kuri punkti ir nozīmīgāki. Izvēloties punktus, kuru atbilstība noteikta ar lielāku pārlicību, arī tālāko SLAM darbību precizitāte tiek palielināta. Aprakstītā hibrīda SLAM metode izmanto neironu tīklu īpašību atbilstību atrašanai, taču pārējā optimizācija tiek veikta, izmantojot klasiskas metodes.

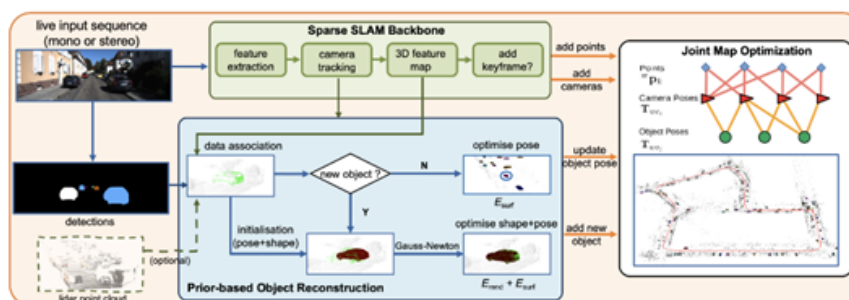
Pētījumā (Teed & Deng, 2022) piedāvā gala-līdz-galam atvasināmu arhitektūru DROID-SLAM, kas pielieto gan klasisku, gan gala-līdz-galam metožu atziņas, padarot sistēmu par noturīgu, taču datus balstītu. DROID-SLAM izmanto neironu tīklu ne tikai, lai izgūtu īpašības un atrastu atbilstības starp kadriem, bet arī kadru pozīciju optimizāciju veic ar apmācāmiem parametriem, tādējādi padarot visu procesu par atvasināmu. Noteiktās pozīciju izmaiņas, balstoties uz secīgu kadru īpašību atbilstībām, izmanto, lai ieglabātu kadrus ar tiem atbilstošajām pozīcijām grafā. Kadru pozīcijas izmaiņa tiek noteikta, izmantojot apmācāmu konvolūciju rekurentu vienību (GRU), kurai ir apslēpts stāvoklis (hidden state), kurš tiek atjaunots starp kadru pāriem. Globālās optimizācijas slānis atbild par to, lai visa grafa kameru pozīcijas tiktu atjaunotas, samazinot kļūdu starp esošajām un patiesajām kadru pozīcijām. Izmantojot dziļās mašīnmācīšanās metodes īpašību izgūšanai, atbilstošu īpašību atrašanai, kadru informācijas glabāšanai apslēptos slāņos, kā arī globālai kadru pozīcijas optimizācijai, DROID-SLAM demonstrē, kā iespējams izveidot gala-līdz-galam datus balstītu SLAM sistēmu, kuru iespējams apmācīt.



6. att. DROID-SLAM mainīgo atjaunināšanas arhitektūras shēma (aizgūts no (Teed & Deng, 2022))

Dziļās mašīnmācīšanās metodes var tikt pielietotas arī semantiskas informācijas izgūšanai no vides, kas var uzlabot SLAM precizitāti, kā arī sniegt papildus informāciju par vidi. DSP-SLAM (J. Wang, Rünz et al., 2021) piedāvā apvienot īpašību izgūšanas metodes ar objektu noteikšanu. Izmantojot īpašību izgūšanu un atbilstību atrašanu starp kadriem, tiek noteikta precīza pozīcijas izmaiņa, savukārt objektu atpazīšana un pielietošana kartēs sniedz globālu semantisku

informāciju. Ar instanču segmentācijas neironu tīklu tiek iegūtas objektu instanču maskas, kuras tiek pielietotas objektu rekonstrukcijai ar dziļuma datiem, ja tādi pieejami, vai arī ar īpašību punktiem, ja to skaits ir pietiekams. Objektu pievienošana notiek, balstoties uz trīsdimensionālo ierobežojošo reģionu noteiktajam objektam ar kartē atrastajiem, ja pieejami dziļuma dati, vai arī atbilstošo īpašību skaitu starp noteikto objektu un kartē atrastajiem. DSP-SLAM veidotais grafs sastāv ne vien no kadru pozīcijām un īpašību punktiem, bet arī objektu instančēm un to pozīcijām. Objekti darbojas kā papildus atskaites punkti, kuri uzlabo kopējo sistēmas precizitāti.



7. att. DSP-SLAM sistēmas shēma (aizgūts no (J. Wang, Rünz et al., 2021))

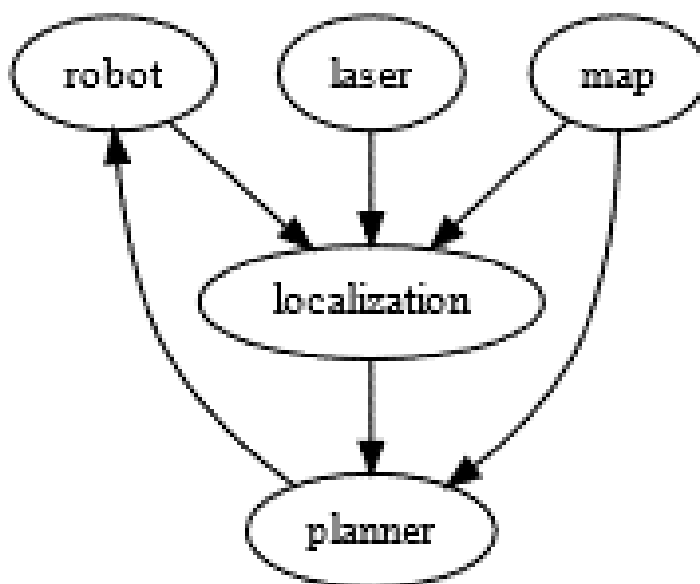
Dziļā mašīnmācīšanās var tikt pielietota arī SLAM, lai ne vien noteiktu precīzu pārvietošanās trajektoriju, bet arī veiktu precīzu telpas rekonstrukciju, kur notika pārvietošanās. GO-SLAM (Zhang, Tosi et al., 2023) piedāvā izmantot apmācāmu SLAM sistēmu, pievienojot arī trīsdimensionālu telpas rekonstrukciju, kas precīzāk apraksta vidi. GO-SLAM pielieto RAFT (Teed & Deng, 2020) arhitektūru optiskās plūsmas noteikšanai, kurā arī tiek izmantota rekurenta vienība stāvokļu iekodēšanai, kā arī veic globālu optimizāciju, izmantojot kadru pozīcijas kļūdas minimizēšanas slāni. Paraleli tiek apmācīts arī trīsdimensionālās telpas rekonstrukcijas tīkls, kurš tiecas samazināt rekonstrukcijas kļūdu ar atjaunotajām pozīcijām. Iegūstot precīzu vides attēlojumu, iespējams precīzāk noteikt telpu izmērus, kā arī tas dod iespēju apstrādāt iegūto trīs dimensiju karti un izgūt semantisku informāciju par to.

Izmantojot datus balstītus dziļās mašīnmācīšanās modeļus iespējams uzlabot, kā arī papildināt SLAM sistēmas. Neironu tīkli var tikt pielietoti īpašību izgūšanai un īpašību atbilstību noteikšanai starp kadriem, lai noteiktu pozīcijas izmaiņu starp kadriem. Tie var tikt izmantoti arī globālai optimizācijai, lai veiktu pozīcijas izlabošanu visiem grafā esošajiem kadriem. Dziļās mašīnmācīšanās modeļi ir piemēroti arī semantiskas informācijas izgūšanai no vides, kas var uzlabot SLAM precizitāti. Visbeidzot, SLAM sistēmas iespējams papildināt ar neironu tīkliem, kurus apmāca precīzi rekonstruēt telpu trīs dimensijās, kas var palīdzēt izmēru noteikšanai un semantiskas informācijas izgūšanai pēc procesa.

1.5. ROS nozīme robotizētu sistēmu izstrādē

ROS (Robot Operating System) (Quigley, 2009) ir atvērtā avota platforma robotizētu sistēmu izstrādei, kura abstrahē zema līmeņa aparatūras procesus, padarot

izstrādi par vieglāk pārvaldāmu. ROS satur jau gatavus risinājumus darbībām ar aparatūru, kurus var pielietot, lai veidotu kompleksas sistēmas, kurās dažādas komponentes komunicē savā starpā. Procesus ROS sauc par mezgliem (node).



8. att. Grafts, kurā attēlota komunikācija starp mezgliem (aizgūts no (Quigley, 2009))

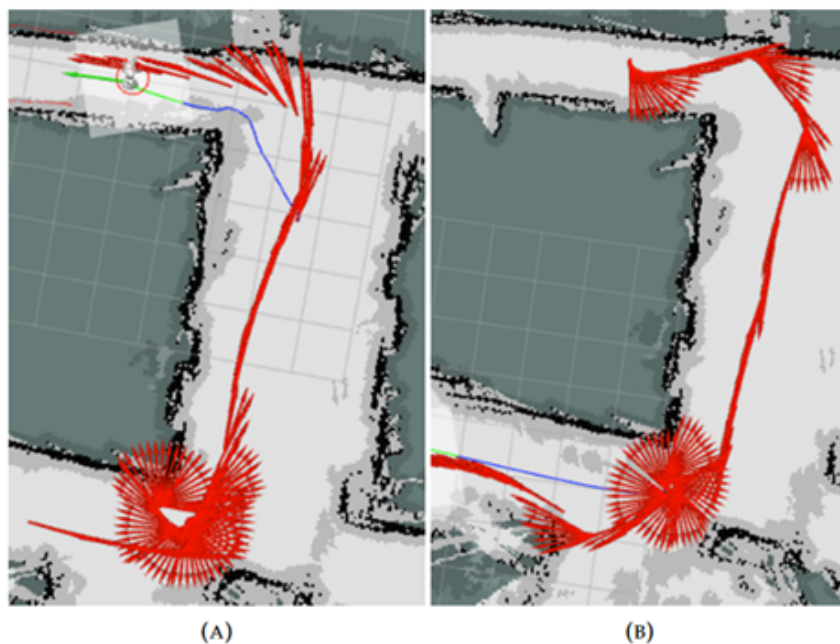
Kompleksu sistēmu darbībai nepieciešams veikt vairākus procesus vienlaikus, no kuriem katrs izmanto citus resursus, komunicē ar citiem procesiem. Komunikācija notiek, izmantojot ziņas (message), kurām ir stingri definēts tips, lai nodrošinātu viennozīmīgu komunikāciju. Procesi var publicēt ziņas tematos (topic), kuriem citi procesi var sekot, tādā veidā nodrošinot informācijas apmaiņu. ROS izmanto standartizētu formu komunikācijā izmantotajiem lielumiem, tādējādi nodrošinot procesu savienojamību un viennozīmīgumu. Izmantojot standartizēta tipa datus, ir zināms, kā veidot procesus un publicēt ziņas, lai citi procesi veiksmīgi darbotos. Standarti arī nodrošina atsevišķu procesu izstrādes neatkarību no pārējās sistēmas, jo iespējams pieņemt, ka sistēma turpinās darboties, kamēr tiks ievērots standarts. ROS ir atvērta avota platforma, kas nozīmē, ka tā ir brīvi pieejama katram lietotājam gan izmantošanai, gan papildināšanai ar saviem risinājumiem. ROS pastāv brīvi pieejamas pakotnes dažādu procesu realizācijai. SLAM kontekstā pieejamas tādas pakotnes kā gmapping, hector_slam, slam_karto. Brīvi pieejami risinājumi dod iespēju pārbaudīt metožu darbību, lai pārbaudītu, vai tās ir piemērotas konkrētajam risinājumam. ROS ir savienojams ar atvērta avota robotikas imitācijas vidi Gazebo. Izmantojot Gazebo, iespējams imitēt robota darbību, ļaujot gūt priekšstatu par izvēlēto algoritmu darbību. Iespējams imitēt sensoru lasījumus, publicēt tos tematos, no kuriem tie var tikt apstrādāti ar izvēlētajām metodēm. ROS piedāvā arī risinājumus robotu navigācijai. Izmantojot

iegūtās kartes, iespējams pielietot trajektorijas plānošanas algoritmus, lai realizētu robotu autonomu kustību. Nolasot informāciju no nepieciešamajiem tematiem, pakotne `move_base` nodrošina robota navigāciju līdz izraudzītam mērķa punktam, ievērojot gan iepriekš izveidoto karti, gan pārvietošanās laikā sastaptos šķēršļus. Tāpat arī iespējams izmantot navigācijas pakotnes dažādu klasisku trajektorijas plānošanas algoritmu pārbaudei. ROS ļauj modulāri, sistemātiski pārbaudīt dažādu komponentu un procesu darbību, kas padara izstrādi pārskatāmāku, kā arī palielina eksperimentu iterāciju ātrumu.

1.6. Pepper robota platforma

Pepper ir SoftBank Robotics kompānijas izstrādāts mobils sociālais robots. Pepper pārvietošanos nodrošina trīs visvirziena riteņi, kas ļauj robotam rotēt un pārvietoties jebkurā virzienā divdimensionālā plaknē. Robota vadību iespējams realizēt, izmantojot iebūvēto Choregraphe vidi, taču sistēmas pielāgojamība ir ierobežota ar piedāvātajām vides funkcijām. Lai izstrādātu pielāgotus risinājumus, iespējams izmantot standartizētu platformu kā ROS.

Darbā (Groot, 2018) aplūko Pepper robota piemērotību autonomai izpētei un navigācijai. Pepper robota spēju autonomi veikt navigāciju ierobežo tā sensoru precizitāte. Pētījumā aplūkoti ROS SLAM algoritmi izmanto lāzēru datus, taču Pepper robota lāzēru dati nav pietiekamas kvalitātes, lai izveidotu karti un lokalizētu sevi tajā. Lai atdarinātu lāzēru skenējumu, iespējams izmantot Pepper pieejamās dziļuma kartes, kuras pārveidojot punktu mākoņa formā, pēc tam pārprojicējot divdimensionālā grīdas plaknē, iegūst mākslīgu lāzēru skenējumu. Pielāgojot algoritmu parametrus, lai kompensētu Pepper robota odometrijas kļūdu, iespējams panākt karšu izveidi. Karšu kvalitāti noteica robota trajektorija – fokusējot Pepper sensorus uz atšķirīgām īpašībām kā stūri un šķēršļi, iespējams palielināt kartes kvalitāti. Robota navigācijas rezultāti bija atkarīgi no izvēlētās trajektorijas un vides. Kustība taisnā līnijā vai brīvās telpās ar vienkāršas formas objektiem izdevās veiksmīgi, taču problēmas sagādā šauras, kā arī sarežģītākas formas telpas.

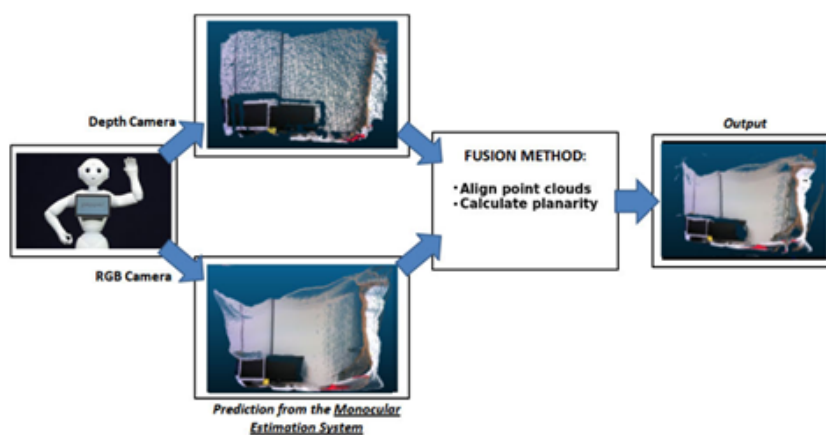


9. att. Pepper robota navigācijas trajektorijas. (A) No apakšējā kreisā līdz augšējam kreisajam, (B) no augšējā kreisā līdz apakšējam kreisajam punktam. Lai gan Pepper izpilda navigāciju, jāizmanto atjaunošanas režīms, lai izlabotu lokalizāciju. Aizgūts no (Groot, 2018)

Visbiežāk navigācija neizdodas kļūdainas lokalizācijas dēļ, kas skaidrojama ar Pepper robota sensoru precizitāti, kā arī ar ierobežoto dziļuma kartes redzes lauku. Tā kā robotam ir iespēja rotēt galvu, tika piedāvāts izmantot šo funkciju, lai aptvertu plašāku daļu vides, tādējādi uzlabojot Pepper lokalizācijas spējas. Tā kā lielākās navigācijas problēmas rodas kļūdainas lokalizācijas dēļ, nepieciešams veltīt īpašu uzmanību visprecīzākās SLAM metodes atrašanai. Autora (Groot, 2018) sniegtie rezultāti parāda to, ka no Gmapping, Hector Mapping un Cartographer metodēm visprecīzākās kartes iespējams iegūt ar Gmapping. Hector Mapping, kurš neizmanto odometrijas datus, nesniedza nevienu pietiekamas kvalitātes karti, lai izmantotu navigācijai. Cartographer arī nesniedza nevienu izmantojamu karti. Gmapping bija vienīgā metode, ar kuru tika iegūta izmantojama karte, kas to padara par vispiemērotāko metodi Pepper robota platformai no trim minētajām (Groot, 2018).

Darbā (Alhmiedat, Marei et al., 2023) salīdzina ROS un Choregraphe pieejamos SLAM algoritmus Pepper robot navigācijas realizācijai. Lai izveidotu telpas karti ar ROS pieejamajām, tika izmantota gmapping pakotne. Karte tika veidota, izmantojot tikai attāluma sensorus, kuru maksimālais darbības rādiuss ir 1.5 metri. Gmapping pakotne ar attālumu sensoru mērījumiem nesniedza pietiekami precīzu karti, lai tiktu veikta navigācija, kas apstiprina nepieciešamību izmantot dziļuma sensoru datus, lai izveidotu mākslīgus lāzeru datus. Lai iegūtu pietiekamas precizitātes karti, lai realizētu navigāciju, tika izmantota Choregraphe vides piedāvātais SLAM algoritms. Izmantojot ROS gmapping pakotni ar lāzeru datiem, iegūst lielāku kartes kļūdu nekā ar Choregraphe algoritmu (Alhmiedat, Marei et al., 2023).

Pētījumā (Bauer, Escalona et al., 2019) piedāvā uzlabot Pepper dziļuma kartes kvalitāti, sapludinot tās ar dziļuma kartēm, kuras iegūst ar konvolūciju neironu tīklu, kurš nosaka dziļuma karti, balstoties uz ieejas attēlu. Dziļuma kartes tiek izteiktas punktu mākoņa formā. Ar neironu tīklu iegūtais punktu mākonis ir blīvs, tādēļ to vispirms retina. Retināto punktu mākonis savieto ar Pepper sensoru punktu mākonis, izmantojot ICP algoritmu. Katram neironu tīkla noteiktajam punktam tiek aprēķināts planaritātes kritērijs. Noteiktā rādiusā no punkta tiek izvēlēti punkti, starp tiem tiek izveidota plakne. Tiek noteikta proporcija tuvu esošajiem punktiem, kuru attālums līdz plaknei ir mazāks par noteiktu sliekšni. Ja proporcija ir pietiekami augsta, dziļuma kartei tiek izvēlēts neironu tīkla noteiktais punkts, citādi izvēlas Pepper sensoru dziļuma kartes punktu.



10. att. Dziļuma noteikšanas sistēmas shēma (Bauer, Escalona et al., 2019))

Pepper robota navigācijā lielākās problēmas rodas lokalizācijas dēļ. Dziļuma kartes var tikt izmantotas, lai iegūtu punktu projekcijas pārvietošanās plaknē, aizstājot vājo attālumu sensoru. Pašu dziļuma karšu kvalitāti iespējams uzlabot, izmantojot neironu tīklus, lai iegūtu dziļuma kartes, kuras var tikt sapludinātas ar Pepper sensoru dziļuma kartēm.

2. METODOLOĢIJA

Lorem ipsum ?? in section...

2.1. Datu kopas

Lorem ipsum...

2.2. Rādītāji

Lorem ipsum...

2.3. Salīdzināšanas protokols

Lorem ipsum...

3. REZULTĀTI

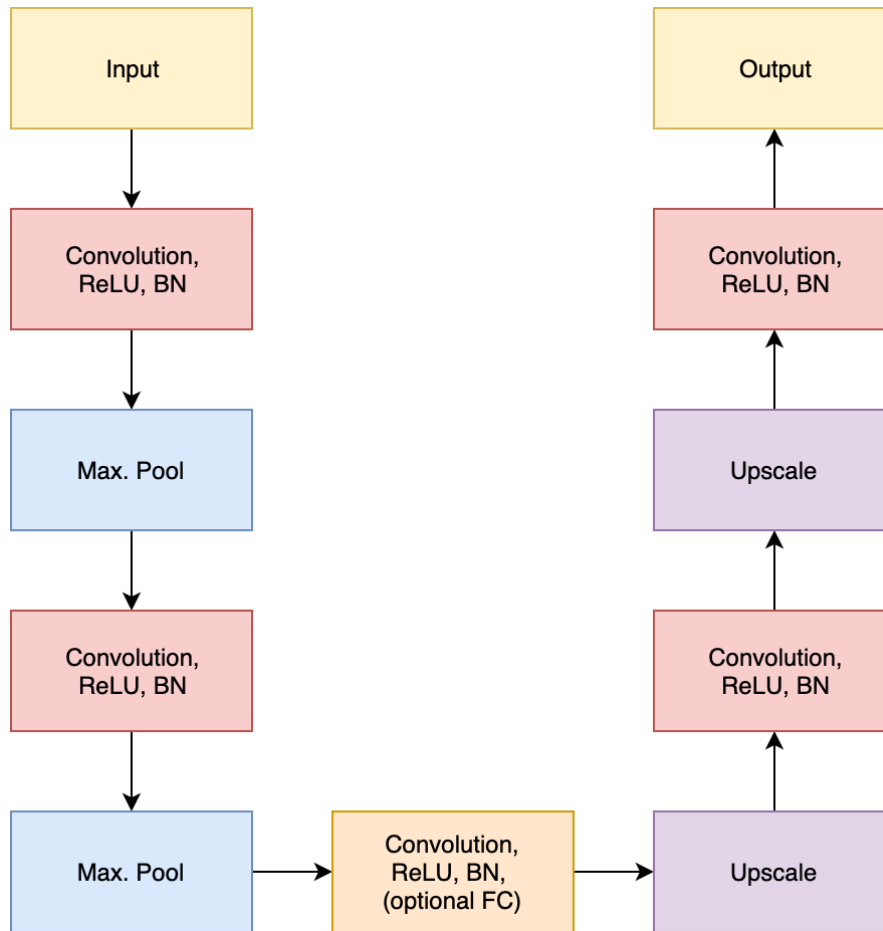
In the past decade, Deep Machine Learning has taken over classical machine learning methods for approximating complex functions using high-dimensional datasets Vaillant, Monrocq et al., 1994 Arulkumaran, Deisenroth et al., 2017 Arik, Chen et al., 2018. Deep Machine Learning models are being used more frequently even to extract functions that describe patterns in the datasets or processes they are observing in an unsupervised or semi-supervised manner Arik, Chen et al., 2018.

3.1. Some first subsection

The objectives of the thesis document are following:

1. Perform review of existing loss functions for functions similar to Deep Q-Learning and Triplet Loss.
2. Develop novel loss functions similar to Deep Q-Learning and Triplet Loss using zero-shot learning.
3. Develop a novel model to learn approximate the Value function in Value Iteration algorithm as shown in 11. attēls.

For each set of methods and loss functions, a full grid search of hyper-parameters has been executed using RTU HPC (High Performance Cluster) that provides access to Nvidia GPUs V100 and K40.



11. att. Simplified activity diagram of the process of novel loss function design in the context of Deep Learning.

In, 2. tabula. information about novel loss functions and their properties regarding DML have been listed. Embedding space refers to normalization or measurement methods between two or more vectors in a latent space. Each of the embedding vectors has been produced by a deep learning based model for the data point. Then two or more embedding vectors have been processed using the loss function and a deep learning based model weights are calculated using the back-propagation algorithm. In addition, for many of these papers uses sample mining methods to select the best training samples to improve the results and speed of the training.

No	Title	Year	Practical application	Dataset / Top-1 Acc.
1	Signature Verification Using A "Siamese" Time Delay Neural Network Vaillant, Monrocq et al., 1994	1993	Signature re-identification	Signatures: 97%
2	Neighbourhood Components Analysis Vaillant, Monrocq et al., 1994	2004	Handwriting identification, Face re-identification	USPS: 85% FERET-B
3	Learning a Similarity Metric Discriminatively, with Application to Face Verification Vaillant, Monrocq et al., 1994	2005	Face re-identification	AT&T: 92.5%

4	Distance metric learning for large margin nearest neighbor classification Vaillant, Monrocq et al., 1994	2005	Handwriting identification, text classification	MNIST: 98.8% Letters: 96.3% 20news: 92% Isolet: 96.6% YaleFaces: 93.9%
5	Large scale metric learning from equivalence constraints Vaillant, Monrocq et al., 1994	2012	Face Re-identification, Image Re-identification	LFW: 80.5% VIPeR: 22%
6	Quadruplet-Wise Image Similarity Learning Vaillant, Monrocq et al., 1994	2013	Product or image retrieval	OSR: 74.6% Pubfig: 77.6%
7	Reidentification by Relative Distance Comparison Vaillant, Monrocq et al., 1994	2013	Face Re-identification	ETHZ: 61.58% i-LIDS: 32.60% VIPeR: 9.12%
8	Deep Metric Learning for Practical Person Re-Identification Vaillant, Monrocq et al., 1994	2014	Face Re-identification	VIPER: 34.49%
9	FaceNet: A unified embedding for face recognition and clustering Vaillant, Monrocq et al., 1994	2015	Face Re-identification	LFW: 99.63% YTF: 95.12%
10	Improved Deep Metric Learning with Multi-class N-pair Loss Objective Vaillant, Monrocq et al., 1994	2016	Product image retrieval, Face Re-identification	LFW: 98.33% SOP: 28.19% CAR-196: 33.5% CUB-200: 27.24%
11	A Discriminative Feature Learning Approach for Deep Face Recognition Vaillant, Monrocq et al., 1994	2016	Face Re-identification	LFW: 99.28% YTF: 94.9% MegaFace: 76.5%
12	Deep Metric Learning via Lifted Structured Feature Embedding Vaillant, Monrocq et al., 1994	2016	Product or image retrieval	CUB200: 55%, CARS196: 48%, SOP: 62%
13	Deep clustering: Discriminative embeddings for segmentation and separation Vaillant, Monrocq et al., 1994	2016	Speaker diarization, separation	WSJ0: 2.74 dB (SDR)
14	Learning Deep Embeddings with Histogram Loss Vaillant, Monrocq et al., 1994	2016	Product or image retrieval	CUHK03: 65.7% CUB-200: 51% Market-1501: 59.47% SOP: 65%
15	Local Similarity-Aware Deep Feature Embedding Vaillant, Monrocq et al., 1994	2016	Product or image retrieval	CARS196: 57.4% CUB-200: 58.3% ImageNet: 48.2%
16	Metric Learning with Adaptive Density Discrimination Vaillant, Monrocq et al., 1994	2016	Image classification, Face Re-identification	Stanford Dogs: 75.1% Flowers-102: 91.4% Oxford-IIIT Pet: 89.4% ImageNet: 84.1%
17	L2-constrained Softmax Loss for Discriminative Face Verification Vaillant, Monrocq et al., 1994	2017	Image classification, Face Re-identification	LFW: 99.33% YTF: 99.78% MNIST: 99.05% IJB-A: 97.5%
18	In Defense of the Triplet Loss for Person Re-Identification Vaillant, Monrocq et al., 1994	2017	Product image retrieval, Face Re-identification	MARS: 90.53%, Market-1501: 79.8%, CUHK03: 87.58%
19	Deep Metric Learning with Angular Loss Vaillant, Monrocq et al., 1994	2017	Product or image retrieval	CAR-196: 71.4%, CUB-200: 54.7%, SOP: 70.9%
20	No Fuss Distance Metric Learning Using Proxies Vaillant, Monrocq et al., 1994	2017	Product or image retrieval	CARS196: 73.22% CUB200: 73.22% SOP: 73.73%
21	Sampling Matters in Deep Embedding Learning Vaillant, Monrocq et al., 1994	2017	Product or image retrieval, Face Re-identification	CARS196: 86.9% CUB200: 63.9% SOP: 72.7%
22	Deep Metric Learning via Facility Location Vaillant, Monrocq et al., 1994	2017	Product or image retrieval	CARS196: 58.11% CUB200: 48.18% SOP: 67.02%

23	Deep spectral clustering learning Vaillant, Monrocq et al., 1994	2017	Product or image retrieval	CARS196: 73.07% CUB200: 43.22% SOP: 67.59%
24	Hard-Aware Deeply Cascaded Embedding Vaillant, Monrocq et al., 1994	2017	Product or image retrieval	CARS196: 83.8% CUB-200: 60.7% In-shop: 62.1 % SOP: 70.1%
25	PPFNet: Global Context Aware Local Features for Robust 3D Point Matching Vaillant, Monrocq et al., 1994	2018	3D Point Cloud matching	SUN3D: 71%
26	Ranked List Loss for Deep Metric Learning Vaillant, Monrocq et al., 1994	2019	Product or image retrieval	CARS196: 82.1% CUB-200: 61.3% SOP: 79.8%
27	Multi-Similarity Loss with General Pair Weighting for Deep Metric Learning Vaillant, Monrocq et al., 1994	2019	Product or image retrieval	CARS196: 77.3% CUB-200: 65.7% In-Shop: 78.2%
28	A Simple and Effective Framework for Pairwise Deep Metric Learning Vaillant, Monrocq et al., 1994	2019	Product or image retrieval	In-shop: 91.3% CARS-196: 86.2% CUB-200: 68.1%
29	Deep Metric Learning Meets Deep Clustering: An Novel Unsupervised Approach for Feature Embedding Vaillant, Monrocq et al., 1994	2020	Product or image retrieval	CUB200: 47.5%, Car196: 42.6%
30	Exponential triplet loss Vaillant, Monrocq et al., 1994	2020	Face Re-identification, Image Re-identification	VGGFace2: 85.7% EMNIST: 86% FMNIST: 93.1% CIFAR10: 87.3% MNIST: 99.6%

2. tabula. Authors and conferences of studies regarding DML.

3.2. Some second subsection

Skip connections are shown in 7. vienādojums. and 10. vienādojums. uses a concatenation operation, but for segmentation tasks it is often also used as an addition operation as in ResNet Vaillant, Monrocq et al., 1994. Final output y that is limited by a sigmoid function σ that is then scaled to the range of values for every cell in the map.

$$o_1 = ReLU(Conv(x)) \quad (1)$$

$$o_2 = MaxPool(o_1) \quad (2)$$

$$o_3 = ReLU(Conv(o_2)) \quad (3)$$

$$o_4 = MaxPool(o_3) \quad (4)$$

$$o_5 = ReLU(Conv(o_4)) \quad (5)$$

$$o_6 = DeConv(o_5) \quad (6)$$

$$o_7 = (o_6, o_3) \quad (7)$$

$$o_8 = Dropout(ReLU(Conv(o_7))) \quad (8)$$

$$o_9 = DeConv(o_8) \quad (9)$$

$$o_{10} = (o_9, o_1) \quad (10)$$

$$o_{11} = ReLU(Conv(o_{10})) \quad (11)$$

$$y = \sigma(o_{11}) \quad (12)$$

A small discount value γ prioritizes short-term rewards, but a larger value prioritizes rewards and positive events that happen at later stages of an episode. Discount factor usually is a hyper-parameter set at a value in the range 0.9-0.99, but for environments where the reward comes later in an episode it might have a lower value.

$$R = \sum_{t=0}^n \gamma^t r_t \quad (13)$$

The research also established that for the task of VI algorithm, UNet models outperform convolutional AE models, as shown in 3. tabula.

Model / Map Size	32	64	128	256
VI	2.95	24.873	195.902	1473.108
VIN	0.031	0.071	0.236	0.833

3. tabula. Comparison of ConvNet based and UNet based models for VI problem.

4. TĀLĀKIE PĒTĪJUMI

Lorem ipsum...

5. SECINĀJUMI

Lorem ipsum...

IZMANTOTIE INFORMĀCIJAS AVOTI

- Alhmiedat, Tareq Ali, Ashraf M. Marei et al., “A SLAM-Based Localization and Navigation System for Social Robots: The Pepper Robot Case”. *Machines* (2023). Pieejams: <https://api.semanticscholar.org/CorpusID:256266025>.
- Arik, Sercan Ö., Jitong Chen et al., “Neural Voice Cloning with a Few Samples”. *ArXiv* abs/1802.06006 (2018).
- Arulkumaran, Kai, M. Deisenroth et al., “A Brief Survey of Deep Reinforcement Learning”. *ArXiv* abs/1708.05866 (2017).
- Ba, Jimmy, J. Kiros et al., “Layer Normalization”. *ArXiv* abs/1607.06450 (2016).
- Bauer, Z., Félix Escalona et al., “Refining the Fusion of Pepper Robot and Estimated Depth Maps Method for Improved 3D Perception”. *IEEE Access* 7 (2019), 185076.—185085. lpp. Pieejams: <https://api.semanticscholar.org/CorpusID:209853461>.
- Bruno, Hudson Martins Silva & Esther Luna Colombini. “LIFT-SLAM: A deep-learning feature-based monocular visual SLAM method”. *Neurocomputing* 455 (2021. g. sept.), 97.—110. lpp. ISSN: 0925-2312. Pieejams: doi: 10.1016/j.neucom.2021.05.027. Pieejams: <http://dx.doi.org/10.1016/j.neucom.2021.05.027>.
- Chaplot, Devendra Singh, Dhiraj Gandhi et al., *Object Goal Navigation using Goal-Oriented Semantic Exploration*. 2020. arXiv: 2007.00643 [cs.CV].
- Cimurs, Reinis, Il Hong Suh et al., *Goal-Driven Autonomous Exploration Through Deep Reinforcement Learning*. 2021. arXiv: 2103.07119 [cs.R0].
- Groot, Robert J. W. de. “Autonomous Exploration and Navigation with the Pepper robot”. 2018. Pieejams: <https://api.semanticscholar.org/CorpusID:59514899>.
- Leonard, J.J. & H.F. Durrant-Whyte. “Simultaneous map building and localization for an autonomous mobile robot”. *Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*. 1991, 1442—1447 vol.3. Pieejams: doi: 10.1109/IROS.1991.174711.
- Mur-Artal, Raúl, J. M. M. Montiel et al., “ORB-SLAM: A Versatile and Accurate Monocular SLAM System”. *IEEE Transactions on Robotics* 31.5 (2015), 1147.—1163. lpp. Pieejams: doi: 10.1109/TR0.2015.2463671.
- Partsey, Ruslan, Erik Wijmans et al., *Is Mapping Necessary for Realistic Point Goal Navigation?* 2022. arXiv: 2206.00997 [cs.CV].
- Qi, Qi, Yan Yan et al., “A Simple and Effective Framework for Pairwise Deep Metric Learning”. *Computer Vision – ECCV 2020*. Izdevis Andrea Vedaldi, Horst Bischof et al., Cham: Springer International Publishing, 2020, 375.—391. lpp. ISBN: 978-3-030-58583-9.
- Quigley, Morgan. “ROS: an open-source Robot Operating System”. *IEEE International Conference on Robotics and Automation*. 2009. Pieejams: <https://api.semanticscholar.org/CorpusID:6324125>.
- Sarlin, Paul-Edouard, Daniel DeTone et al., *SuperGlue: Learning Feature Matching with Graph Neural Networks*. 2020. arXiv: 1911.11763 [cs.CV].
- Shah, Dhruv, Ajay Sridhar et al., *ViNT: A Foundation Model for Visual Navigation*. 2023. arXiv: 2306.14846 [cs.R0].

- Teed, Zachary & Jia Deng. *DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras*. 2022. arXiv: 2108.10869 [cs.CV].
- *RAFT: Recurrent All-Pairs Field Transforms for Optical Flow*. 2020. arXiv: 2003.12039 [cs.CV].
- Thrun, Sebastian, Wolfram Burgard et al., *Probabilistic robotics*. Cambridge, Mass.: MIT Press, 2005. ISBN: 0262201623 9780262201629. Pieejams: <http://www.amazon.de/gp/product/0262201623/102-8479661-9831324?v=glance&n=283155&n=507846&s=books&v=glance>.
- Vaillant, R., C. Monrocq et al., “Original Approach for the Localisation of Objects in Images”. *IEE Proceedings - Vision, Image and Signal Processing* 141.4 (1994), 245.—250. lpp.
- Wang, Jingwen, Martin Rünz et al., *DSP-SLAM: Object Oriented SLAM with Deep Shape Priors*. 2021. arXiv: 2108.09481 [cs.CV].
- Wang, Sen, Ronald Clark et al., “DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks”. *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017. g. maijs. Pieejams: doi: 10.1109/icra.2017.7989236. Pieejams: <http://dx.doi.org/10.1109/ICRA.2017.7989236>.
- Zhang, Youmin, Fabio Tosi et al., “GO-SLAM: Global Optimization for Consistent 3D Instant Reconstruction”. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023. g. okt.
- Zhao, Jianwei, Shengyi Liu et al., “Research and Implementation of Autonomous Navigation for Mobile Robots Based on SLAM Algorithm under ROS”. *Sensors* 22.11 (2022). ISSN: 1424-8220. Pieejams: doi: 10.3390/s22114172. Pieejams: <https://www.mdpi.com/1424-8220/22/11/4172>.

PIELIKUMI

1. Pielikumi

Tabula parāda visus ievāktos datus katram algoritmam, visu procentu jaukto klašu un iezīmju eksperimentos.

precision	recall	f1	accuracy	tp	fp	fn	tn	dispersija	std	algoritms	% jauktas klases	% jauktas iezīmes
95.34%	85.71%	90.27%	90.76%	696	34	116	778	0.105724	0.325152	VI	0%	0%
89.71%	86.95%	88.31%	88.49%	706	81	106	731	0.168235	0.410165	VI	10%	10%
94.47%	79.93%	86.59%	87.62%	649	38	163	774	0.201928	0.449364	VI	20%	20%
88.77%	81.77%	85.13%	85.71%	664	84	148	728	0.229843	0.479419	VI	30%	30%
82.63%	67.36%	74.22%	76.60%	547	115	265	697	0.242078	0.492015	VI	40%	40%
52.53%	24.26%	33.19%	51.17%	197	178	615	634	0.245685	0.495666	VI	50%	50%
100.00%	99.42%	99.71%	99.70%	807	0	5	812	0.002664	0.051611	BBB	0%	0%
99.29%	97.41%	98.34%	98.35%	791	6	21	806	0.003512	0.059261	BBB	10%	10%
98.48%	97.59%	98.02%	98.05%	792	12	20	800	0.003335	0.057746	BBB	20%	20%
99.49%	89.60%	94.23%	94.60%	728	4	84	808	0.0054	0.073486	BBB	30%	30%
98.33%	87.11%	92.32%	92.93%	709	12	103	800	0.01185	0.108856	BBB	40%	40%
56.37%	49.18%	52.24%	55.29%	399	310	413	502	0.012961	0.113846	BBB	50%	50%
100.00%	100.00%	100.00%	100.00%	812	0	0	812	1.07E-05	0.003275	MCD	0%	0%
100.00%	99.88%	99.94%	99.94%	811	0	1	812	0.000907	0.030111	MCD	10%	10%
99.82%	99.76%	99.79%	99.79%	810	1	2	811	0.002085	0.04566	MCD	20%	20%
98.23%	98.01%	98.12%	98.18%	797	14	15	798	0.0101	0.1005	MCD	30%	30%
86.52%	96.44%	91.18%	90.67%	782	123	30	689	0.039816	0.199539	MCD	40%	40%
60.40%	56.33%	58.17%	59.49%	457	305	355	507	0.073562	0.271223	MCD	50%	50%