

Michał BUGAŁA

ALGORYTMY STOSOWANE W SYSTEMACH POJAZDÓW AUTONOMICZNYCH

Streszczenie. Wiele światowych ośrodków badawczych związanych z wytwarzaniem pojazdów lądowych, zwłaszcza przeznaczonych do transportu i komunikacji w ruchu miejskim, nadal pracuje nad opracowaniem pojazdu wyposażonego w systemy nie wymagające udziału człowieka w procesie kierowania pojazdem. Celami, do których się dąży, jest zapewnienie maksimum bezpieczeństwa (minimalizacja wypadków z udziałem człowieka) oraz optymalizacja kosztów transportu (eliminacja kierowcy z pojazdu, optymalny wybór trasy). W artykule, będącym wynikiem szeroko prowadzonych prac studialnych w OBRUM sp. z o.o., omówiono syntetycznie drogi rozwoju pojazdów autonomicznych oraz kluczowe algorytmy pojazdów autonomicznych opisane szczegółowo w cytowanej - przywołanej literaturze. Przedstawione wyniki stanowią punkt wyjścia do dalszych prac realizowanych w Ośrodku nad pojazdem autonomicznym.

Słowa kluczowe: algorytmika, autonomia, pojazd autonomiczny, system pojazdów autonomicznych.

1. WPROWADZENIE

Artykuł prezentuje i omawia podstawowe algorytmy stosowane w systemach pojazdów autonomicznych (pojazdy lądowe), których wykorzystanie może różnić się w zależności od założonego poziomu autonomii i zastosowanych czujników/sensorów. Najważniejsze zadania systemów pojazdów autonomicznych, bez względu na poziom automatyzacji i rodzaj czujników, to zebranie informacji o otoczeniu, oszacowanie własnej pozycji w otoczeniu, przewidywanie ruchu innych obiektów i przesłanie wykonanych obliczeń do systemów sterujących pojazdem (kierownica, przepustnica, hamulec itd.). Całość materiału przedstawionego w artykule podzielono na dwie części zawierające: część historyczną (rozdział 2) oraz rozwiązania wykorzystywane współcześnie (rozdział 3).

2. RYS HISTORYCZNY

Kwestią dyskusyjną, a nawet sporną może być potrzeba implementacji systemów pojazdów autonomicznych w kontekście samochodów osobowych, ich użytkowników oraz zapowiadanych usług wypożyczania samochodów [1]. Historia pokazuje, że takie próby były już dokonywane wraz z pojawieniem się pierwszych pojazdów obsługiwanych przez człowieka. Powyższe wątpliwości nie dotyczą oczywiście rozwiązań militarnych [2] oraz koncepcji autonomicznych systemów transportu miejskiego [3], [4].

2.1. Pierwsze próby automatyzacji sterowania

Historia pierwszych prób automatyzacji sterowania pojazdami sięga lat dwudziestych, kiedy to w roku 1926 firma Chandler zademonstrowała pojazd sterowany radiowo przez operatora siedzącego w drugim pojeździe podążającym za pojazdem bezzałogowym. Oczywiście nie można w tym przypadku mówić o autonomii pojazdu, ale jest to pierwszy przypadek, kiedy pojazd samochodowy poruszał się po drodze bez załogi [5], [6]. W roku 1956 firma General Motors wyposażyła eksperymentalny model Firebird II w odbiornik sygnałów pochodzących z przewodu elektrycznego znajdującego się w podłożu tak zwanej

autostrady przyszłości, który miał stanowić system naprowadzania pojazdów [7]. W roku 1979 eksperymentalny pojazd The Stanford Cart był zdolny do poruszania się w zamkniętym pomieszczeniu bez ingerencji człowieka i wykorzystywał algorytm przetwarzania obrazu o nazwie The Cart's Vision Algorithm. Algorytm ten był zainspirowany metodą planowania Blocks World i polegał na redukcji obrazu do zestawu krawędzi, jednak nie okazał się odpowiedni dla scen plenerowych, zawierających wiele skomplikowanych kształtów i barw [8]. Metoda Blocks World z 1960 roku to jednak dziś jedna z najbardziej znanych metod w dziedzinie planowania w sztucznej inteligencji [9] i rozpoznawania otoczenia [10], [11].

2.2. Pierwszy pojazd autonomiczny

W 1995 roku eksperymentalny pojazd VaMP przejechał ponad tysiąc kilometrów bez pomocy człowieka i był to pierwszy samochód autonomiczny potrafiący poruszać się po określonym obszarze. Prototyp zdolny był do jazdy w ruchu ulicznym i wyprzedzania pojazdów, a zaimplementowany system autonomii EMS-Vision opierał się na danych pochodzących z dwuogniskowych układów kamer (45 i 15 stopni) zamocowanych na dwuosiowych platformach [12]. Taki dobór kamer (nowsze rozwiązania w systemie EMS-Vision wykorzystują układ trzech kamer [13]) związany był z jazdą przy dużych prędkościach, obserwacją potencjalnych przeszkód na większych i mniejszych odległościach, nierównościami terenu i zdolnościami do interpretacji układu przestrzennego. System EMS-Vision korzystał z:

- mapy sieci dróg;
- obiektów statycznych znajdujących się na mapie, które używane były jako punkty orientacyjne (znaczniki);
- danych statystycznych (np. szerokość pasa ruchu).

Zdobywanie wiedzy dynamicznej, tj. zbieranie informacji o otoczeniu w czasie jazdy, polegało na identyfikacji obiektów (znaczników), obliczaniu położenia obiektów (lokalizacja i orientacja) w przestrzeni z wykorzystaniem algorytmu HTC (ang. *Homogeneous Coordinate Transformation*) oraz tworzeniu drzewka sceny, zawierającego te obiekty (elementami łączącymi obiekty w drzewku sceny były ich względne położenia). System dodatkowo posiadał jednostki decyzyjne do utrzymywania ważnych obiektów w polu widzenia (dla analizy przez specjalistyczne moduły), do sterowania pojazdem oraz centralną jednostkę decydującą. Centralna jednostka posiadała priorytet decyzyjny w razie braku zadań, konfliktów pomiędzy jednostką sterowania a jednostką utrzymywania uwagi, i także w razie innych problemów. Moduły specjalistyczne, w jakie wyposażony był system EMS-Vision, to:

- rozpoznawanie drogi – algorytm nie tylko rozpoznawał drogę, ale tworzył jej model z połączonych segmentów i na tej podstawie ustalał położenie pojazdu;
- kontrola uwagi – odpowiedzialna za komunikację z dwuosiową głowicą obrotową wykonującą manewry skokowe (sakadyczne) związane ze śledzeniem obiektów;
- nawigacja – moduł obliczał kilka tras dla punktu docelowego wraz z czasem rozpoczęcia i zakończenia jazdy, długością trasy i szacowaną możliwością przejazdu (faza planowania). Obliczenia wysyłane były do jednostki centralnej, która decydowała o wyborze trasy. Wybrana trasa dodawana była do listy zadań w planie misji. Plan misji zawierał zadania typu wybrana trasa, obiekty do śledzenia (droga, punkty orientacyjne) i inne. Oryginalny plan misji mógł ulec zmianie w trakcie wykonywania misji, gdy przykładowo nie udało się odszukać oczekiwanego skrzyżowania;

- kontrola pojazdu (sterowanie) – moduł odpowiedzialny był za analizę informacji pochodzących z jednostek decyzyjnych i wykonywanie algorytmów kontroli sprzężenia zwrotnego, kontroli zwrotnej i reguł przejściowych. Moduł oceniał tak powstałe zbiory równań, obliczał zmienne korygujące i sterował siłownikami [14].

Powyższy opis systemu EMS-Vision nie zawiera modyfikacji [13], [15], jakie powstały po 1995 roku, jednak obrazuje stopień skomplikowania oraz liczbę użytych algorytmów, rozumianych jako zdefiniowane czynności, konieczne do wykonywania konkretnych zadań w systemach pojazdów autonomicznych.

2.3. Zawody *DARPA Grand Challenge*

Duże znaczenie w rozwoju systemów pojazdów autonomicznych miały zawody sponsorowane przez DARPA (*Defense Advanced Research Projects Agency*), w którym uczestnicy mieli skonstruować pojazd mogący przejechać 241 kilometrów (150 mil) przez pustynię w stanie Nevada. W 2004 roku żaden zespół nie ukończył zadania, a najlepszy przejechał zaledwie 13 kilometrów (8 mil). Problem związany był z analizą obrazu i trudnością rozpoznawania otoczenia, w którym nie występują wyraźne linie (np. pobocze z chodnikiem), oraz trudnością interpretacji dużej ilości cieni na nierównym terenie pustynnym. Standardowe algorytmy analizy obrazu (tworzenie konturów, dopasowywanie linii itp.), identyfikacja obiektów oraz strategie omijania przeszkód nie sprawdziły się w 2004 roku. W 2005 roku zaczęto używać uczenia maszynowego w procesach analizy obrazu i pięć pojazdów zdołało ukończyć trasę [16].

Wykorzystanie uczenia maszynowego w analizie obrazu dało początek nowemu trendowi w budowaniu systemów dla pojazdów autonomicznych. Duże przedsiębiorstwa rozpoczęły budować własne, inteligentne systemy dla pojazdów autonomicznych, np. projekt Waymo rozpoczęty w 2009 roku przez firmę Google (w projekcie wzięli udział członkowie zespołu Stanley, który zwyciężył *DARPA Grand Challenge* w 2005 roku [17]). Oprogramowanie systemu w pojeździe Stanley zawierało 100000 linijek kodu i było odpowiedzialne za interpretację danych z czujników oraz podejmowanie decyzji nawigacyjnych [18].

2.4. Zawody *DARPA Urban Grand Challenge*

W 2007 roku przeprowadzono zawody w terenie miejskim (zurbanizowanym) we Victorville w stanie Kalifornia, w którym wzięły udział 53 zespoły. Do finału dostało się 11 zespołów i 6 z nich zdołało ukończyć trasę. Oprogramowanie zwycięzcy zawodów, zespołu Tartan Racing [19], składało się z 500000 linijek kodu i umożliwiała samodzielną nawigację po mieście w ruchu ulicznym. Opracowany pojazd Boss (model samochodu Chevrolet Tahoe wraz ze sprzętem i oprogramowaniem) używał oprogramowania percepcyjnego, planistycznego i behawioralnego do wnioskowania o ruchu i podejmowania decyzji w czasie przemieszczania się do miejsca docelowego. Pojazd wyposażony był w kilkanaście laserów, kamer i radarów, co umożliwiło mu planowanie trasy z uwzględnieniem przeszkód statycznych i dynamicznych (poruszających się obiektów). Informacje na temat otoczenia były kompletowane przez algorytmy znajdowania i rozpoznawania ograniczeń pasa ruchu, granic parkingów, znaków drogowych i innych. Dodatkowo zaimplementowano algorytmy identyfikujące niebezpieczne zachowania innych kierowców. Najważniejsze cechy technologii opracowanej przez zespół Tartan Racing to:

- jazda zgodna z zasadami ruchu drogowego (uwzględniająca zasady pierwszeństwa przejazdu na skrzyżowaniach);

- detekcja i śledzenie innych pojazdów na dużych odległościach;
- szukanie miejsc do parkowania i parkowanie;
- podążanie za pojazdami w bezpiecznej odległości;
- reakcje na nietypowe zdarzenia (np. zablokowanie drogi) [20].

2.5. Pierwsze pojazdy autonomiczne dopuszczone do ruchu

W 2014 roku firma Tesla wprowadziła na rynek model S z oprogramowaniem o nazwie *Autopilot Firmware 7.0*, który oferował drugi poziom automatyzacji [21]. Opis poziomów automatyzacji według SAE International (poprzednio SAE - *Society of Automotive Engineers*) prezentuje tablica 1.

Tablica 1. Poziomy automatyzacji według SAE [22]

Poziom	Nazwa	Opis
0	Brak automatyzacji (<i>No Automation</i>)	Całkowity brak automatyzacji. Wszystkie decyzje i czynności podejmowane są przez kierowcę.
1	Systemy asystujące (<i>Driver Assistance</i>)	Systemy wsparcia kierowcy na przykład w przyspieszaniu lub zwalnianiu prędkości jazdy w oparciu o informacje zewnętrzne. Ten poziom spełniają między innymi inteligentne tempomaty ACC (<i>Adaptive Cruise Control</i>), systemy wspomaganie parkowania i systemy wspomagające utrzymywanie pasa ruchu LKA (<i>Lane Keep Assist</i>).
2	Automatyzacja częściowa (<i>Partial Automation</i>)	Drugi poziom automatyzacji zakłada autonomiczny charakter systemów z poziomu 1, ale wymaga ciągłej obserwacji otoczenia przez kierowcę.
3	Automatyzacja zależna (<i>Conditional Automation</i>)	Pojazd jest prowadzony przez systemy automatyczne, lecz kierowca ma obowiązek reagowania w sytuacji, gdy decyzję automatu uzna za nietrafną.
4	Automatyzacja wysoka (<i>High Automation</i>)	Pojazd jest prowadzony przez systemy automatyczne nawet wtedy, kiedy kierowca nie zareaguje odpowiednio w sytuacji niepewnej.
5	Automatyzacja pełna (<i>Full Automation</i>)	Pełna autonomia. Pojazd jest w stanie wykonywać wszystkie funkcje jazdy w każdych warunkach. Kierowca nadal może mieć możliwość sterowania pojazdem.

Oczywiście nie był to jeszcze w pełni pojazd autonomiczny (model S), ale ówczesna wersja oprogramowania umożliwiała autonomiczne podążanie pojazdu wzdłuż drogi, zmianę pasa ruchu i parkowanie na żądanie. Model S był jednym z pierwszych samochodów autonomicznych dostępnym na rynku (nie w pełni autonomiczny, jednak spowodował rozpoczęcie wielu dyskusji na temat produkcji i użytkowania pojazdów autonomicznych) [23].

Innymi producentami, jednymi z nielicznych, oferującymi automatyzację w swoich samochodach w tym czasie, byli: Ford (system utrzymywania pasa ruchu, automatyczne parkowanie równoległe, system unikania wypadków) [24], Mercedes [25], Audi i Nissan [26].

Obecnie trwa wyścig pomiędzy producentami samochodów osobowych w dostarczeniu swoim klientom jak najbezpieczniejszych i w miarę możliwości bezobsługowych produktów. Z dużą konkurencyjnością wiąże się ograniczony dostęp do informacji i brak możliwości dokładnej analizy systemów używanych w praktyce przez producentów samochodów.

3. ROZWIĄZANIA WSPÓŁCZESNE

W tej części artykułu przedstawione zostaną podstawowe algorytmy analizy obrazu, magazynowania informacji oraz algorytmy decyzyjne, jakich używa się w konstruowaniu prototypowych systemów autonomicznych (niekoniecznie odbiegających od rozwiązań komercyjnych).

Występująca liczba zdarzeń, jakie mogą pojawić się na drodze, różnorodność obrazu (krajobraz, pora roku, pora dnia), różne zasady ruchu drogowego w różnych krajach (np. ruch lewostronny lub prawostronny, znaki drogowe) powoduje, że opracowanie oprogramowania dla w pełni autonomicznego pojazdu wymaga ogromnej wiedzy i liczby sprawdzonych rozwiązań. Można zaryzykować stwierdzenie, że każdy algorytm przetwarzania i analizy obrazu może mieć zastosowanie w systemach pojazdów autonomicznych. Liczba stosowanych algorytmów jest jednak ograniczona przez możliwości obliczeniowe i zdeterminowana przez rodzaj używanych czujników/sensorów: kamera, LASER, GPS, Infrared, LiDAR itp. Przegląd algorytmów zestawiono według następującego podziału:

- rozpoznawanie otoczenia za pomocą obrazu z kamer (ustęp 3.1):
 - rozpoznawanie pasa ruchu (drogi);
 - segmentacja semantyczna obrazu i identyfikacja obiektów;
 - budowanie mapy otoczenia na podstawie obrazu (wizualny SLAM);
- rozpoznawanie otoczenia za pomocą czujników (ustęp 3.2):
 - budowanie mapy otoczenia na podstawie danych z czujników (SLAM);
- detekcja i śledzenie poruszających się obiektów (DATMO, ustęp 3.3);
- planowanie i decyzyjność (ustęp 3.4):
 - algorytmy planowania trasy przejazdu;
 - metody planowania ruchu pojazdu i sterowania;
 - mechanizmy podejmowania decyzji;
 - sztuczna inteligencja.

3.1. Rozpoznawanie otoczenia za pomocą obrazu z kamer

Obraz z kamer to najbogatsze i niezbędne źródło informacji w procesie identyfikacji otoczenia. Oprócz kamer używa się czujników radarowych, które nie są wrażliwe na pogodę, kurz i zanieczyszczenia oraz droższych czujników laserowych - lidarów [27].

3.1.1. Rozpoznawanie pasa ruchu (drogi)

Zbieganie się równoległych linii w punkcie skupienia w przestrzeni trójwymiarowej to cenna informacja w identyfikacji drogi, ponieważ tworzące pas ruchu linie równoległe

spotykają się w jednym punkcie na obrazie dwuwymiarowym (obrazie z kamery). Identyfikacja pasa ruchu na jezdni polega na znalezieniu linii skupiających się w jednym punkcie.

Metoda VP (*Vanishing Point*) do znajdowania drogi w terenie pustynnym, zaproponowana przez C. Rasmussen'a [28], polega na obliczeniu dominujących orientacji na segmentach obrazu (obraz w rozdzielczości 640 na 480 pikseli podzielony na 72 segmenty), oszacowaniu położenia punktu skupienia oraz śledzeniu punktu w kolejnych klatkach obrazu. Metoda Rasmussen'a, opracowana w celu rozpoznawania drogi na terenie pustynnym, odgrywa dziś ważną rolę w wykrywaniu pasów ruchu drogowego na obrazie [29] i stanowi podstawę dla bardziej zaawansowanych algorytmów [30].

3.1.2. Segmentacja semantyczna obrazu i identyfikacja obiektów

Współczesne metody analizy obrazu, wykorzystujące uczenie maszynowe (widzenie maszynowe), są w stanie wykryć określone kategorie obiektów, w tym ludzi, znaki drogowe i samochody na złożonych obrazach.

Uczenie maszynowe może posłużyć do segmentacji semantycznej obrazu (np. metoda FRNN – *Full Resolution Residual Networks* [31]) i dokładnego opisu obrazu, tj. segmentacji z dokładnym obrysem sylwetek obiektów, jednak ze względu na duży koszt obliczeniowy i powstające liczne artefakty, nie stosuje się metod segmentacji obrazu w praktyce [27], [32], [33], [34]. Takie podejście może ulec zmianie wraz z rozwojem dedykowanego sprzętu [35], [36] oraz nowszych, szybszych metod służących segmentacji semantycznej [33], [37].

Widzenie maszynowe stosowane jest do wykrywania określonych cech obrazu (predefiniowanych): wykrywania pasów ruchu, przeszkód, poruszających się obiektów i szacowania odległości [27]. Wspecjalizowane systemy wspomaganie kierowcy ADAS (*Advanced Driver Assistance Systems*) potrafią identyfikować znaki drogowe, sygnalizację świetlną i oznaczenia na drodze. Skuteczniejsze metody uczenia maszynowego do rozpoznawania znaków drogowych (oraz innych elementów obrazu) wraz z użytymi algorytmami, jakie pojawiły się w różnych badaniach i zostały opisane w artykułach naukowych, można podzielić na 3 grupy [34]:

- ogólne algorytmy uczenia maszynowego:
 - metoda Viola-Jones z 2001 roku [38];
 - metody wektorów nośnych SVM (*Support Vector Machine*) poprzedzone algorytmami ekstrakcji cech, takimi jak:
 - histogramy zorientowanych gradientów HOG (*Histogram Of Gradients*) [39];
 - piramida histogramów zorientowanych gradientów PHOG (*Pyramid Histogram of Gradients*) [40];
 - metoda umożliwiająca uzyskanie mocnego klasyfikatora z dużej liczby słabych klasyfikatorów o nazwie AdaBoost (*Adaptive Boosting*) [41];
 - klasyfikator cech Haar-like [39];
 - ekstraktor cech MSER (*Maximally Stable Extremal Regions*) [42];
 - HOG wraz z MSER i algorytmami ekstrakcji regionów WaDe [43];
 - HOG z LDA (*Linear Discriminant Analysis*) [44][45];
 - HOG z BCT (*Bilateral Chinese Transform*) i VBT (*Vertex Bisector Transform*) [46];

- metody AdaBoost w kombinacji z:
 - detektorem cech LBP (*Local Binary Pattern*) [47];
 - piramidą obrazów i HOG'iem [48];
 - klasyfikatorem Haar-like [49];
- inne:
 - metody wykorzystujące algorytmy genetyczne (K. Kaplan) [50];
 - BoVW (*Bag of Visual Words*) z algorytmem SIFT (*Scale-Invariant Feature Transform*) [51];
- sztuczne sieci neuronowe:
 - wirtualna generalizacja pamięci masowej z dostępem losowym w sieciach neuronowych VG-RAM-WNN (*Virtual Generalizing Random Access Memory Weightless Neural Networks*) [52];
 - probabilistyczne sieci neuronowe PNN (*Probabilistic Neural Networks*) w kombinacji z CPT (*Central Projection Transformation*) i ACIS (*Adaptive Color Image Segmentation*) [53];
- głębokie uczenie maszynowe:
 - konwolucyjne sieci neuronowe CNN (*Convolutional Neural Network*) [54];
 - CNN poprzedzone przetworzeniem obrazu w skali szarości przez maszynę SVM [55];
 - wykorzystanie dwóch sieci CNN do detekcji znaków na obrazie i identyfikacji znaków [56];
 - CNN z algorytmem RPN (*Region Proposal Network*) [57] w sieci Fast R-CNN (*Fast Region-Based Convolutional Neural Network*) [58];
 - metody detekcji Faster R-CNN (*Faster Region-Based Convolutional Neural Network*) do wykrywania obiektów będących w obszarze zainteresowania, w tym znaków drogowych [59].

Według źródła [34] najbardziej popularnymi i zbadanymi metodami używanymi w rozpoznawaniu znaków drogowych są metody SVM. AdaBoost cechują się szybkim czasem wykonywania, a metody sieci neuronowych działają najwolniej. Metody głębokiego uczenia maszynowego posiadają wysokie wymagania sprzętowe i z tego względu mogą być trudne do zastosowania w systemach ADAS. Metoda Faster R-CNN może być alternatywą dla segmentacji semantycznej obrazu.

3.1.3. Budowanie mapy otoczenia na podstawie obrazu (wizualny SLAM)

Metoda SLAM (*Simultaneous Localization And Mapping*), czyli jednoczesna lokalizacja i budowa mapy otoczenia, umożliwia precyzyjne określenie własnej pozycji względem otoczenia. W algorytmach SLAM trajektoria pojazdu, jak i położenie wszystkich punktów orientacyjnych, muszą być szacowane bardzo szybko (*on-line*) bez znajomości własnej pozycji (lokalizacji) a priori [60]. W przypadku algorytmów wykorzystujących obraz, w którym punkty orientacyjne muszą być identyfikowane i śledzonego w każdej klatce obrazu z kamery, wymagania co do wykorzystywanego sprzętu (mocy obliczeniowej) są bardzo wysokie. Na podstawie punktów orientacyjnych budowana jest trójwymiarowa mapa otoczenia, a sama metoda SLAM występuje pod postacią algorytmów:

- algorytm EKF-SLAM (EKF - *Extended Kalman Filter*) z 1987 roku, który wykorzystuje filtr Kalmana, jako głównego estymatora w szacowaniu struktury otoczenia [61];
- wykorzystujący filtry cząsteczkowe Rao-Blackwellised RBPFs (*Rao-Blackwellised Particle Filters*) algorytm FastSLAM [62], [63];
- FastSLAM 2.0 będący rozwiązaniem hybrydowym, łączącym wykorzystanie filtru Kalmana i filtrów cząsteczkowych [64];
- RGB-D SLAM wykorzystujący obraz i głębie obrazu (Kinect) [65];
- ORB-SLAM i ORB-SLAM2 dla pojedynczych obrazów, obrazów stereo (widzenie stereoskopowe) i kamer RGB-D [66]. Algorytm wykorzystuje deskryptor ORB (*Oriented FAST and Rotated BRIEF*) [67];
- system oparty na modelach obliczeniowych procesów nawigacyjnych w hipokampie (część mózgu ssaków) o nazwie RatSLAM [68];
- LSD-SLAM – algorytm generujący mapy głębokości z pojedynczych klatek obrazu bez wykorzystania punktów orientacyjnych [69];
- L-SLAM – algorytm redukujący wymiarowość algorytmów FastSLAM [70].

Wizualne metody SLAM używane są przede wszystkim w pojazdach o małych gabarytach, np. dronach (przykładowe źródło: [71]).

3.2. Rozpoznawanie otoczenia za pomocą czujników

RADAR (*Radio Detection And Ranging*) i LIDAR (*Light Detection and Ranging*) to podstawowe czujniki używane do rozpoznawania otoczenia w systemach używanych w praktyce [32], [72], [73]. Radar był używany w przemyśle motoryzacyjnym od dawna i może określać prędkość, zasięg i orientację obiektów – nie jest również wrażliwy na warunki pogodowe. Lidaru, który jest cięższy i droższy zaczęto używać w większej skali w 2007 roku (zawody *DARPA Urban Challenge*) [74] z powodu otrzymywania dokładniejszych wyników (pomiarów). Wykorzystanie radaru i lidaru wiąże się z mapowaniem otoczenia i algorytmami SLAM, które nie są obciążone problemem identyfikacji punktów orientacyjnych na obrazie.

Mniej zaawansowaną, i tym samym tańszą grupę urządzeń, stanowią czujniki ultradźwiękowe [75], montowane na przykład na zderzakach samochodów (czujniki parkowania), które pozwalają na pomiar odległości od przeszkody.

3.2.1. Budowanie mapy otoczenia na podstawie danych z czujników (SLAM)

Algorytmy SLAM można podzielić ze względu na rodzaj czujnika:

- Radar:
 - działający w trybie *offline* algorytm GraphSLAM [76], który polega na budowaniu grafów i mapowaniu siatki otoczenia [77];
 - polegający na budowaniu grafów i wyszukiwaniu konfiguracji węzłów o najmniejszym błędzie Graph-Based SLAM [78];
 - wykorzystujący FastSLAM i GraphSLAM algorytm Real-Time Radar SLAM, który wykonuje się w czasie 45 milisekund (Intel Core i7-3830K) [79].
- Lidar:
 - bazujący na maksymalizacji prawdopodobieństwa ML-SLAM (ML – *Maximum Likelihood*) [80];

- bazujący na algorytmie TBM (*Transferable Belief Model*) [81] algorytm Credibilist SLAM (lub C-SLAM) [82];
- wykorzystujący metodę rejestracji kształtów trójwymiarowych ICP (*Iterative Closest Point*) [83] algorytm ICP-SLAM [84];
- Cartographer SLAM firmy Google, który oprócz lidarów wykorzystuje IMU (*Inertial Measurement Unit*) i obraz z kamer [85], [86].

Do budowania mapy otoczenia wykorzystuje się również algorytm LOAM (*Lidar Odometry And Mapping*), który polega na wykorzystaniu pomiarów odległości dwuosowego lidarów poruszającego się w sześciu stopniach swobody [87].

3.3. Detekcja i śledzenie poruszających się obiektów (DATMO)

Algorytmy DATMO (*Detection And Tracking of Moving Objects*), czyli detekcja i śledzenie poruszających się obiektów, mają zastosowanie w przypadku wykorzystania obrazu z kamery [88], radarów [89] i lidarów [90].

W przypadku wykorzystania metody DATMO i SLAM korzyści są obustronne. Identyfikacja obiektów dynamicznych w otoczeniu oraz ignorowanie ich przez algorytm SLAM ma fundamentalne znaczenie w budowaniu dokładnych map w terenie miejskim, które charakteryzują się dużą ilością obiektów ruchomych. Z drugiej strony możliwość użycia algorytmu SLAM podczas śledzenia ruchomych obiektów pozwala lepiej oszacować globalne prędkości i pozycje, a tym samym lepiej oszacować trajektorie obiektów ruchomych [91]. Listę rozwiązań dotyczących integracji SLAM z DATMO, na podstawie źródła [92], przedstawia tablica 2.

Tablica 2. Metody SLAM i DATMO

Autor/Autorzy	Metoda SLAM	Metoda DATMO		Środowisko	Obiekty
		Powiązanie danych	Śledzenie		
Wang 2002-2004 [93]	Grid-based (EKF)	MHT (<i>Multiple Hypothesis Tracking</i>) [94]	IMM (<i>Interactive Multiple Model</i>) [95]	Otwarta przestrzeń	Ludzie, samochody, rowery, autobusy
Hähnel, Schulz, Burgard 2003 [96]	Grid-based (filtr Bayesa)	SBJPDA (<i>Sample-based Joint Probabilistic Data Association Filters</i>) [97]		Otwarta przestrzeń	Ludzie
Montesano, Minguez, Montano 2005 [98]	Grid map	NNR (<i>Nearest Neighbor Rule</i>) [99]	EKF	Pomieszczenie	Ludzie, drzwi
Solà 2007 [100]	BiCamSLAM (dwie kamery)	EKF		Pomieszczenie	Ludzie, pudełka, stoły, kosze
Vu 2009 [91]	ML-SLAM i EKF	DDMCMC (<i>Data-Driven Markov Chain Monte Carlo</i>)		Otwarta przestrzeń	Ludzie, samochody, rowery, autobusy
Vu, Burlet, Aycard 2010 [101]	Bayes, ICP	MHT	IMM	Otwarta przestrzeń	Ludzie, samochody

Metody detekcji i śledzenia obiektów DATMO opracowywane samodzielnie, tj. bez kooperacji z metodami SLAM, za pomocą różnych czujników i w różnej konfiguracji są przedmiotem wielu opracowań, np.:

- RoadPlot-DATMO z radarami, lidarami 2D i lidarem 3D [102];
- opisany w 2016 roku przez F. J. Botha [103] algorytm wykorzystujący GIW-PHD (*Gaussian Inverse Wishart – Probability Hypothesis Density*), czujnik radarowy i dwie kamery.

3.4. Planowanie i decyzyjność

Identyfikacja nietypowych zdarzeń na drodze, które mogą pojawić się niemal zawsze i wszędzie, mogą prowadzić do potencjalnych problemów i muszą być rozwiązane w sposób bezpieczny. Rozważaniom etycznym, w przypadkach zagrażającym zdrowiu pasażerów (lub operatora w przypadku niepełnej autonomii) i uczestników ruchu, decyzji autonomicznych, dedykowane są liczne źródła [104], [105], [106].

3.4.1. Algorytmy planowania trasy przejazdu

Jedną z pierwszych decyzji, jaką musi podjąć pojazd autonomiczny, to wybór trasy (ścieżki) przejazdu do wyznaczonego punktu, najczęściej według kryterium odległości, czasu przejazdu lub zakładanego zużycia paliwa. Algorytmy heurystyczne, uwzględniające kryterium przejazdu, czyli koszt na krawędziach łączących węzły (graf ważony) w grafach skierowanych, to:

- algorytm Bellmana-Forda, posiadający ograniczenia w postaci konieczności definiowania krawędzi o nieujemnych wagach [107], [108];
- algorytm Dijkstra, który może być używany w przypadku znanej topologii dróg [109];
- algorytm A* z roku 1968 [110], który wraz z modyfikacjami algorytmu [111] jest najczęściej stosowanym rozwiązaniem [112], [113].

Do lokalizacji własnego pojazdu i wyznaczania trasy na mapie (znanej a priori) używa się przede wszystkim czujników GPS i danych pozycji geograficznych. Istnieją również rozwiązania wykorzystujące mapy zbudowane z danych pochodzących z lidarów 3d [114] i semantyki obrazu [115].

3.4.2. Metody planowania ruchu pojazdu i sterowania

Metody planowania ruchu pojazdu i sterowania można podzielić, według źródła [116] na:

- wykorzystujące zaawansowane metody percepcji i podejmowania decyzji planowanie sekwencyjne z podziałem na:
 - metody klasyczne (tradycyjne):
 - sterowanie nieliniowe (*nonlinear control*) [117] polegające na śledzeniu trajektorii ruchu pojazdu i utrzymywaniu się poprzez sterowanie pojazdem na wyznaczonej trasie;
 - sterowanie predykcyjne MPC (*Model Predictive Control*) [118] polegające na precyzyjnym wyznaczeniu trajektorii pojazdu z wykorzystaniem technik optymalizacji i przewidywania ruchu;
 - sterowanie zwrotne i wyprzedzające (*feedback-feedforward control*) [119], którego celem w przypadku sterowania wyprzedzającego (*feedforward*) jest zapewnienie estymacji kąta skrętu wymaganego do pokonywania drogi o znanej krzywiźnie i profilu prędkości, a w przypadku sterowania

zwrotnego (*feedback*) zminimalizowanie błędu sterowania wyprzedzającego;

- sterowanie równoległe lub autonomia równoległa (*parallel autonomy*), w której system autonomii działa jako *anioł stróż*, zapewniając dodatkowe bezpieczeństwo poprzez zastępowanie kierowcy w niebezpiecznych sytuacjach. Systemy autonomii równoległej mogą również przejmować obsługę pojazdu na życzenie kierowcy i w tym przypadku mówi się o autonomii przeplatanej (*interleaved autonomy*) [116];
- metody *end-to-end* oparte na uczeniu maszynowym, które zastępują klasyczne algorytmy percepcji otoczenia (opisane w ustępach 3.1, 3.1.1, 3.1.2, 3.1.3, 3.2, 3.2.1 i 3.3). Metody *end-to-end* dotyczą pojedynczych komponentów systemu (w odróżnieniu od autonomii *end-to-end* opisanej w dalszej części), np. utrzymywania pojazdu na pasie ruchu [118];
- planowanie uwzględniające zachowania innych kierowców (*behavior-aware planning*) można traktować jako otwarte wyzwanie, ze względu na ilość interakcji pomiędzy różnymi użytkownikami ruchu. Jazda zgodna z zasadami społecznymi oraz współpraca systemów autonomicznych z prawdziwymi kierowcami ma ogromne znaczenie dla bezpieczeństwa w niepewnych środowiskach [116]. Takie sytuacje, jak skręt w lewo przy zakorkowanej drodze, przejazd przez zwężenie bez sygnalizacji świetlnej (ruch wahadłowy), niepoprawne oznakowanie drogi i błędy innych kierowców to tylko przykłady sytuacji, z jakimi musi sobie poradzić w pełni autonomiczny pojazd drogowy i system decyzyjny. Artykuł nie opisuje rozwiązań w przypadku nietypowych sytuacji na drogach, ale istnieje spora liczba źródeł próbujących rozwiązać i opisać te problemy, np. [121], [122], [123];
- planowanie *end-to-end* (autonomia *end-to-end*), które jest rozwiązaniem opartym na uczeniu maszynowym i podejściu całościowym, tj. bez podziału na komponenty odpowiedzialne za jakąś część systemu (np. utrzymywanie pojazdu na pasie ruchu). Istnieją systemy, w których konwolucyjna sieć neuronowa uczy się zdolności postrzegania (percepcji) i generuje trasę tylko na podstawie sekwencji jazdy i danych z lidar [124]. Podobne rozwiązanie wykorzystujące obraz z kamery, lidar i semantyczną segmentację obrazu (model danych KITTI [125]) zostało opracowane w tym samym roku [126].

3.4.3. Mechanizmy podejmowania decyzji

Mechanizmy podejmowania decyzji w zakresie kierowania pojazdami DDM (*Driving Decision-making Mechanism*) są uznawane za kluczową technologię zapewniającą bezpieczeństwo jazdy pojazdu autonomicznego. Mechanizmy DDM ustalają strategię poruszania się pojazdów na podstawie zgromadzonych informacji, tj. wyników pracy algorytmów stosowanych w systemach pojazdów autonomicznych (wyżej wymienionych w dokumencie) i danych z czujników.

Autor nie podjął próby dokładnej klasyfikacji mechanizmów podejmowania decyzji ze względu na ogromną liczbę stosowanych rozwiązań zależnych od skali możliwości (lub konieczności) użycia mechanizmów decyzyjnych (np. pojedynczych decyzji związanych z identyfikacją znaków drogowych i wiarygodności oszacowań czy dotyczących szeregu decyzji związanych z przejazdem pojazdu do punktu docelowego), odrębnych komponentów systemu (np. utrzymujący bezpieczną odległość od innego pojazdu aktywny tempomat ACC) i poziomu automatyzacji. Możliwe jest jednak zaprezentowanie pewnych modeli decyzyjnych, które mogą bazować na:

- drzewach decyzyjnych [127], [128] i diagramach [129];
- procesach decyzyjnych POMPD (*Partially Observable Markov Decision Processes*) [130];
- uczeniu maszynowym:
 - SVR (*Support Vector Machine Regression*) z optymalizacją PSO (*Particle Swarm Optimization*) [131];
 - MPD (*Markov Decision Processes*) z RL (*Reinforcement Learning*) [132];
 - DRL (*Deep Reinforcement Learning*) [133].

3.4.4. Sztuczna inteligencja

Planowanie ruchu, sterowanie pojazdem i podejmowanie decyzji tylko za pomocą metod uczenia maszynowego, tj. autonomii *end-to-end*, jest podejściem bardzo uproszczonym, biorąc pod uwagę liczbę algorytmów detekcji, identyfikacji, śledzenia obiektów, budowania mapy otoczenia i reagowania na nietypowe zdarzenia (omówione w punkcie od 3.1 do 3.4.3). Teoretycznie, sztuczna inteligencja może rozwiązać każdy z omówionych problemów, jednak jej skuteczność będzie zależać od liczby i doboru danych treningowych, tj. przykładów potrzebnych do jej nauki. Kolejnym problemem, chociaż na pewno do rozwiązania, może być konieczność przeprowadzenia odpowiednich testów sprawdzających bezpieczeństwo takiego systemu w różnych sytuacjach, w różnych warunkach i w różnej lokalizacji. Obecnie, o czym można dowiedzieć się ze źródła [134], metody testowania systemów pojazdów autonomicznych nie są opracowane w sposób rzetelny i nie są zbyt wymagające.

Metodę autonomii *end-to-end* zaimplementowano już w roku 1989 w systemie ALVINN (*Autonomous Land Vehicle in a Neural Network*) [135]. Firma NVIDIA wykorzystwała głębokie uczenie maszynowe (konwolucyjne sieci neuronowe) w 2016 roku [136]. Obecnie popularne stają się techniki nauczania imitacyjnego (*imitation learning*) [137], w których sieć uczy się na podstawie obrazów imitujących rzeczywistość (symulatorów) [138].

4. PODSUMOWANIE

Systemy pojazdów autonomicznych łączą różnorodne techniki postrzegania otoczenia z wykorzystaniem popularnych czujników [75] i algorytmów. Metody rozwiązujące konkretne zadania (rozpoznawanie otoczenia, planowanie, decyzyjność) wykorzystują algorytmy analityczne, jak i metody sztucznej inteligencji, które w niektórych przypadkach dają o wiele lepsze wyniki. Zastąpienie wszystkich algorytmów przez sztuczną inteligencję (autonomia *end-to-end*) nie wydaje się na dzień dzisiejszy realnym rozwiązaniem ze względu na konieczność zgromadzenia ogromnej i odpowiedniej ilości danych.

Z pewnością minie jeszcze sporo czasu zanim w pełni autonomiczne pojazdy zostaną włączone do ruchu drogowego - wiąże się to ze zmianami w prawie o ruchu drogowym, kwestiami odpowiedzialności odszkodowawczej za wypadki spowodowane przez pojazdy autonomiczne, kwestiami etycznymi (dylemat wagonika) oraz ustaleniem zasad przebiegu egzaminu na prawo jazdy dla tego typu kierowcy. Niemniej jednak, rozwój technologii informatycznych będzie odgrywał ciągły wpływ na dobór algorytmów stosowanych w systemach pojazdów autonomicznych i powstawanie nowych metod i algorytmów.

W podsumowaniu nie proponuje się listy najlepszych algorytmów, jakie powinny być użyte w systemie pojazdów autonomicznych, ponieważ ich dobór zależy od:

- mocy obliczeniowej stosowanego sprzętu, który ma wpływ na szybkość wykonywania algorytmów;
- rodzaju czujników, które zależne są od gabarytów i przeznaczenia pojazdu;
- jakości czujników, która uzależniona jest od zakładanych kosztów (lidar 3D z zasięgiem do 300 metrów to koszt kilkunastu tysięcy dolarów);
- poziomu automatyzacji.

Celem napisania artykułu (na bazie wykonanego przeglądu literatury - studiów literaturowych) było zebranie dotychczas stosowanych algorytmów w systemach pojazdów autonomicznych, w tym algorytmów rozwijanych z myślą o ich zastosowaniu w rozwiązaniach praktycznych. Cytowana, obszerna literatura tematu może posłużyć zainteresowanym czytelnikom jako drogowskaz do szerszego zapoznania się z algorytmiką systemów stosowanych w pojazdach autonomicznych.

5. LITERATURA

- [1] Ziomek O.: Volkswagen będzie wypożyczał samochody elektryczne na minuty, zacznie w 2019 roku. Portal DobreProgramy, 2018.
<https://www.dobreprogramy.pl/Volkswagen-bedzie-wyposzczac-samochody-elektryczne-na-minuty-zacznie-w-2019-roku,News,89124.html> [dostęp: 20.09.2018].
- [2] National Research Council USA: Technology Development for Army Unmanned Ground Vehicles. Washington, 2003.
- [3] Kardasz P., Lyubov O. Kardasz E.: Autonomiczne pojazdy. Biuletyn naukowy Warszawskiej Wyższej Szkoły Informatyki Stosowanej. Informatyka (7). 2017. ISSN: 2082-9892.
- [4] Berrada J.: Modeling Transportation Systems involving Autonomous Vehicles: A State of the Art. Elsevier, 2017.
- [5] Engineering.com: The Road to Driverless Cars: 1925-2025.
<https://www.engineering.com/DesignerEdge/DesignerEdgeArticles/ArticleID/12665/The-Road-to-Driverless-Cars-1925--2025.aspx> [dostęp: 28.08.2018].
- [6] Felton R.: The Man Who Tested The First Driverless Car in 1925 Had A Bizarre Feud With Harry Houdini. <https://jalopnik.com/the-man-who-tested-the-first-driverless-car-in-1925-had-1792312207> [dostęp: 31.08.2018].
- [7] GM Heritage Center: Self-Driving Cars, in 1956?
https://www.gmheritagecenter.com/featured/Autonomous_Vehicles.html [dostęp: 28.08.2018].
- [8] CyberneticZoo.com: 1960 – Stanford Cart.
<http://cyberneticzoo.com/cyberneticanimals/1960-stanford-cart-american/> [dostęp: 28.08.2018].
- [9] Gupta N., S. N. Dana: On the complexity of blocks-world planning. Artificial Intelligence (56). Elsevier, 1992.
<http://www.cs.umd.edu/~nau/papers/gupta1992complexity.pdf> [dostęp: 28.08.2018].
- [10] Gupta A., Efros A., M. Hebert: Blocks World Revisited: Understanding Using Qualitative Geometry and Mechanics. European Conference on Computer Vision. 2010.
<http://www.cs.cmu.edu/~abhinavg/blocksworld/blocksworld.pdf> [dostęp: 28.08.2018].

- [11] Kuipers B. J.: An Hypothesis-Driven Recognition System for the Blocks World. MIT, 1974.
- [12] Dickmanns E. D.: Vehicles Capable of Dynamic Vision. Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence. Nagoya, 1997.
- [13] Dickmanns E. D.: Expectation-Based, Multi-Focal, Saccadic (EMS) Vision for Dynamic Scene Understanding. Control Engineering Practise (10), nr 8. Elsevier, 2012.
- [14] Gregor R., Lutzeler M., Pellkofer M. Siedersberg K.-H, Dickmanns E. D.: EMS-Vision: A Receptual System for Autonomous Vehicles. Transactions on Intelligent Transportation Systems (3), nr 1. IEEE, 2002.
- [15] Hofmann U., Rieder A., Dickmanns E. D.: EMS-Vision: Application to Hybrid Adaptive Cruise Control. Proceedings of the IEEE Intelligent Vehicles Symposium 2000. IEEE, 2000. ISBN: 0-7803-6363-9.
- [16] Launchbury J.: A Darpa Perspective on Artificial Intelligence. 2017. <https://www.youtube.com/watch?v=-O01G3tSYpU> [dostęp: 29.08.2018].
- [17] Markoff J.: Google Cars Drive Themselves, in Traffic. 2010. <https://www.nytimes.com/2010/10/10/science/10google.html> [dostęp: 29.08.2018].
- [18] Russel S.: DARPA Grand Challenge Winner, Popular Mechanics. 2006. <https://www.popularmechanics.com/technology/robots/a393/2169012/> [dostęp: 29.08.2018].
- [19] Cargegie Mellon University: Carnegie Mellon's Tartan Racing Wins \$2M DARPA Urban Challenge. <https://www.cmu.edu/homepage/practical/2007/fall/victory.shtml> [dostęp: 30.08.2018].
- [20] Cargegie Mellon University: Tartan Racing Technology. <http://www.cs.cmu.edu/~tartanrace/tech.html> [dostęp: 30.08.2018].
- [21] Tesla Motors: Enhancing Safety and Convenience with Summon. 2016. <https://www.tesla.com/blog/enhancing-safety-and-convenience-summon> [dostęp: 30.08.2018].
- [22] NHTS: Preliminary Statement of Policy Concerning Automated Vehicles. 2017. <https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety> [dostęp: 28.08.2018].
- [23] Zhang B.: ELON MUSK: In 2 years your Tesla will be able to drive from New York to LA and find you. 2016. <https://finance.yahoo.com/news/elon-musk-two-years-car-202858960.html> [dostęp: 30.08.2018].
- [24] Cunningham W.: 2013 Ford Fusion review. 2013. <https://www.cnet.com/roadshow/reviews/2013-ford-fusion-review/> [dostęp: 30.08.2018].
- [25] English A.: New Car Tech: 2014 Mercedes-Benz S-Class. 2013. <https://www.roadandtrack.com/new-cars/news/a3689/new-car-tech-2014-mercedes-benz-s-class/>.
- [26] Tregaskis S.: The first generation of self-driving cars – in pictures. 2014. <https://www.theguardian.com/technology/gallery/2014/may/29/first-generation-self-driving-cars-google-in-pictures>.
- [27] Hirz M., Walzel B.: Sensor and object recognition technologies for self-driving cars. Computer-Aided Design and Application. 2018.

- [28] Rasmussen C.: Grouping dominant orientations or Ill-structured road following. International Conference on Computer Vision and Pattern Recognition. IEEE, 2004.
- [29] Miksik O.: Rapid Vanishing Point Estimation for General Road Detection. Proceedings – IEEE International Conference on Robotics and Automation. IEEE, 2012.
- [30] Moon Y. Y., Geem Z. W., Han G. T.: Vanishing point detection for self-driving car using harmony search algorithms. Swarm and Evolutionary Computation. 2018.
- [31] Pohlen T., Hermans A., Mathias M., Leibe B.: Full Resolution Residual Networks. 2017. http://openaccess.thecvf.com/content_cvpr_2017/papers/Pohlen_Full-Resolution_Residual_Networks_CVPR_2017_paper.pdf [dostęp: 3.09.2018].
- [32] Petrovskaya A., Thrun S.: Model based vehicle detection and tracking for autonomous urban driving. Autonomous Robots (26). Springer, 2009.
- [33] Trempl M., Arjona-Medina J., Unterthiner T.: Speeding up Semantic Segmentation for Autonomous Driving. 2016. <https://openreview.net/pdf?id=S1uHiFyyg> [dostęp: 3.09.2018].
- [34] Mukhometzianov R., Wang Y.: Review. Machine learning techniques for traffic sign detection. 2017. <https://arxiv.org/ftp/arxiv/papers/1712/1712.04391.pdf> [dostęp: 4.09.2018].
- [35] Nvidia: Nvidia Drive. <https://www.nvidia.com/en-us/self-driving-cars/drive-platform/> [dostęp: 3.09.2018].
- [36] Hruska J.: Tesla Working with AMD on Self-Driving Car Processor. <https://www.extremetech.com/extreme/256171-tesla-working-amd-self-driving-car> [dostęp: 3.09.2018].
- [37] Zhang H., Qi X., Shen X., Jia J.: ICNet for Real-Time Semantic Segmentation on High-Resolution Images. 2018. <https://arxiv.org/pdf/1704.08545.pdf> [dostęp: 3.09.2018].
- [38] K. Brkić: An overview of traffic sign detection methods. <https://pdfs.semanticscholar.org/74a1/336f1fbc8b7bb3b6e159711af1a91336ce22.pdf> [dostęp: 4.09.2018].
- [39] T. Radu, Z. Karel, L. V. Gool: Multi-view traffic sign detection, recognition, and 3D localisation. 2009 Workshop on Applications of Computer Vision (WACV). 2009.
- [40] Sugiharto A., Harjoko A.: Traffic sign detection based on HOG and PHOG using binary SVM and k-NN. 3rd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE). 2016.
- [41] Fleyeh H., Biswas R., Davami E.: Traffic sign detection based on AdaBoost color segmentation and SVM classification. IEEE, 2013.
- [42] Gurel C. S.: Traffic Sign Recognition using multi-class SVM. Perception for Autonomous Robots. 2018.
- [43] Samuele S., Alioscia P., Federico T., Nicola F., Luigi S.: A traffic sign detection pipeline based on interest region extraction. The 2013 International Joint Conference on Neural Networks (IJCNN). 2013.
- [44] Martinez A. Kak A., PCA versus LDA. IEEE Trans. Pattern Analysis and Machine Intelligence (23), nr 2. IEEE, 2001.
- [45] Wang G., Ren G, Wu Z., Zhao Y., JiangL.: A robust, coarse-to-fine traffic sign detection method. The 2013 International Joint Conference on Neural Networks (IJCNN). 2013.

- [46] Shi J., Lin H.-Y.: A vision system for traffic sign detection and recognition. 26th International Symposium on Industrial Electronics (ISIE). IEEE, 2017.
- [47] Fukazawa Y.: Traffic Sign Recognition system on Android devices. 2003. https://www.researchgate.net/publication/311540973_Traffic_Sign_Recognition_system_on_Android_devices [dostęp: 4.09.2018].
- [48] Jagannathan S., Desappan K., Swami P.: Efficient Object Detection and Classification on Low Power Embedded Systems. International Conference on Consumer Electronics (ICCE). IEEE, 2017.
- [49] Li Z., Dong C., Zheng L., Liu L.: Traffic Signs Detection Based on Haar-Like Features and Adaboost Classifier. Second International Conference on Transportation Information and Safety. 2013.
- [50] Kaplan K., Kurtul C., Akin H. L.: Real-Time Traffic Sign Detection and Classification Method for Intelligent Vehicles. International Conference on Vehicular Electronics and Safety. IEEE, 2012.
- [51] Nguyen K.-D., Le D.-D, Duc D. A.: Efficient Traffic Sign Detection Using Bag of Visual Words and Multi-scales SIFT. International Conference on Neural Information Processing, 2013.
- [52] De Souza A., Fontana C., Mutz F.: Traffic sign detection with VG-RAM weightless neural networks. The 2013 International Joint Conference on Neural Networks (IJCNN). 2013.
- [53] Zhang K., Sheng Y., Wang P.: Automatic recognition of traffic signs in natural scene image based on central projection transformation. http://www.isprs.org/proceedings/XXXVII/congress/3b_pdf/26.pdf [dostęp: 4.09.2018].
- [54] Shustanov A., Yakimov P.: CNN Design for Real-Time Traffic Sign Recognition. *Procedia Engineering* (201). ELSEVIER, 2017.
- [55] Wu Y., Liu Y., Li J., Liu H., Hu X.: Traffic sign detection based on convolutional neural networks. The 2013 International Joint Conference on Neural Networks (IJCNN). 2013.
- [56] Zhu Z., Liang D., Zhang S.: Traffic-Sign Detection and Classification in the Wild. https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Zhu_Traffic-Sign_Detection_and_CVPR_2016_paper.pdf [dostęp: 4.09.2018].
- [57] Bappy J. H., Roy-Chowdhury K.: CNN based region proposals for efficient object detection. http://intra.ece.ucr.edu/~mbappy/pubs/ICIP_2016.pdf [dostęp: 4.09.2018].
- [58] Xiong C., Wang C., Ma W., Shan Y.: A traffic sign detection algorithm based on deep convolutional neural network. International Conference on Signal and Image Processing (ICSIP). IEEE, 2016.
- [59] Zuo Z., Yu K., Zhou Q., Wang X., Li T.: Traffic Signs Detection Based on Faster R-CNN. 37th International Conference on Distributed Computing Systems Workshops (ICDCSW). 2017.
- [60] Durrant-Whyte H., Bailey T.: Simultaneous localisation and mapping (slam): Part I The essential algorithms. *Robotics and Automation Magazine*. 2006.
- [61] Smith R., Cheeseman P.: On the representation and estimation of spatial uncertainty. *Int. Journal of Robotics Research*. 1987.
- [62] Montemerlo M., Thrun S., Koller D., Wegbreit B.: FastSLAM: A factored solution to the simultaneous localization and mapping problem. *AAAI National Conference on Artificial Intelligence*, 2002.

- [63] Doucet A., De Freitas N., Murphy K., Russel S.: Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. *Uncertainty in Artificial Intelligence Proceedings*. 2000.
- [64] Montemerlo M., Thrun S.: FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. *Proceedings of IJCAI*. 2003.
- [65] Ebdres F., Hess J., Engelhard N.: An evaluation of the RGB-D SLAM system. *International Conference on Robotics and Automation*. IEEE, 2012.
- [66] Mur-Artal E., Tardos J.D.: ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *Transactions on Robotics PP (99)*. IEEE, 2016.
- [67] Rublee E., Rabaud V., Konolige K., Bradski G.: ORB: an efficient alternative to SIFT or SURF. http://www.willowgarage.com/sites/default/files/orb_final.pdf, [dostęp: 5.09.2018].
- [68] Ball D., Heath S., Wiles J.: OpenRatSLAM: an open source brain-based SLAM system. *Autonomous Robots*. 2013.
- [69] Engel J., Shöps T., Cremers D.: LSD-SLAM: Large-Scale Direct Monocular SLAM. *Computer Vision*. ECCV, 2014.
- [70] Petridis V., Zikos N.: L-SLAM: „Reduced dimensionality FastSLAM algorithms. *The 2010 International Joint Conference on Neural Networks (IJCNN)*. 2010.
- [71] Yang S., Shrerer S. A., Zell A.: Visual SLAM for autonomous MAVs with dual cameras. *International Conference on Robotics and Automation*. IEEE, 2014.
- [72] Levinson J., Askeland J., Becker J.: Towards Fully Autonomous Driving: Systems and Algorithms. *Intelligent Vehicles Symposium (IV)*. IEEE, 2011.
- [73] Teichman A., Thrun S.: Practical object recognition in autonomous driving and beyond. *Advanced Robotics and its Social Impacts*. IEEE, 2012.
- [74] Ors A. O.: Radar, camera, LiDAR and V2X for autonomous cars. 2017. <https://blog.nxp.com/automotive/radar-camera-and-lidar-for-autonomous-cars> [dostęp: 03.09.2018].
- [75] Rainer M.: Indoor positioning technologies. ETH Zürich, 2012. <https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/54888/eth-5659-01.pdf> [dostęp: 27-08-2018]
- [76] Thrun S., Montemerlo M.: The GraphSLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. *The International Journal of Robotics Research*. 2006.
- [77] Colleens T., Colleens J. J., Ryan C.: Occupancy grid mapping: An empirical evaluation. *Control & Automation*. IEEE, 2007.
- [78] Shuster F., Keller C., Rapp M. Haueis M.: Curio C.: Landmark based radar slam using graph optimization. *Intelligent Transportation Systems (ITSC)*. IEEE, 2016.
- [79] Schoen M., Horn M., Hahn M., Dickmann J.: Real-Time Radar SLAM. *FAS*, 2017. <https://www.uni-das.de/images/pdf/veroeffentlichungen/2017/01.pdf> [dostęp: 6.09.2018].
- [80] Thrun S., Burgard W., Fox D.: A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. *International Conference on Robotics and Automation*. IEEE, 2000.
- [81] Smets P., Kennes R.: The transferable belief model. *Artificial Intelligence (66)*, nr 2. Elsevier, 1994.

- [82] Trehard C., Pollard E., Bradai B., Nashashibi F.: Credibilist SLAM Performances with Different Laser Set-ups. 13th International Conference on Control, Automation, Robotics and Vision (ICARCV). 2014.
- [83] Besl P. J., McKay H. D.: A method for registration of 3-D shapes,. IEEE PAMI (14), nr 2. IEEE, 1992.
- [84] Mendes E., Koch P., Lacroix S.: ICP-based pose-graph SLAM. International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2016.
- [85] Datta A.: Google releases SLAM tool Cartographer to open source community. Geospatial World, 2016. <https://www.geospatialworld.net/blogs/google-open-sources-slam-tool-cartographer/> [dostęp: 10.09.2018].
- [86] Nuchter A., Bleier M., Schauer J., Janotta P.: Improving Google's Cartographer 3D Mapping by Continuous-Time SLAM. <https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W3/543/2017/isprs-archives-XLII-2-W3-543-2017.pdf> [dostęp: 10.09.2018].
- [87] Zhang J., Singh S.: LOAM: Lidar Odometry and Mapping in Real-time. Conference: Robotics: Science and Systems Conference. 2014.
- [88] Rodriguez-Canosa R. G., Thomas S., del Cerro J.: A Real-Time Method to Detect and Track Moving Objects (DATMO) from Unmanned Aerial Vehicles (UAVs) Using a Single Camera. Remote Sensing. 2012.
- [89] Vivet D., Checchin P., Chapuis R.: A Mobile Ground-based Radar Sensor of Detection And Tracking of Moving Objects. EURASIP Journal on Advances in Signal Processing 2012 (1). 2012.
- [90] Magnier V., Gruyer D., Godelle J.: Automotive LIDAR objects Detection and Classification Algorithm Using the Belief Theory. Intelligent Vehicles Symposium. IEEE, 2017.
- [91] Vu T.-D.: Vehicle perception: Localization, mapping with detection, classification and tracking of moving objects. Ph.D. dissertation. Institut National Polytechnique de Grenoble (INPG), 2009.
- [92] Pancham A., Tlale N., Bright G.: Literature Review of SLAM and DATMO. 2011. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.886.7941&rep=rep1&type=pdf> [dostęp: 10.09.2018].
- [93] Wang C.-C., Thorpe C., Thrun S.: Online Simultaneous Localization and Mapping with Detection and Tracking of Moving Objects: Theory and Results from a Ground Vehicle in Crowded Urban Areas. International Conference on Robotics and Automation (ICRA). IEEE, 2003.
- [94] Kim C., Li F., Ciptadi A., Rehg J. M.: Multiple Hypothesis Tracing Revisited. International Conference on Computer Vision (ICCV). IEEE, 2015.
- [95] Watson G. A., Blair W. D.: IMM Algorithm for Tracking Targets that Maneuver through Coordinated Turns. Proceedings of SPIE. 1992.
- [96] Hähnel D., Schulz D., Burgard W.: Mobile robot mapping in populated environments. <http://ais.informatik.uni-freiburg.de/publications/papers/haehnel-populated.pdf> [dostęp: 11.09.2018].
- [97] Rezatofighi S. H., Milan A., Zhang Z.: Joint Probabilistic Data Association Revisited. https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Rezatofighi_Joint_Probabilistic_Data_ICCV_2015_paper.pdf [dostęp: 11.09.2018].
- [98] Montesano L., Minguez J., Montano L.: Modeling the static and the dynamic parts of the environment to improve sensor-based navigation. International Conference on Robotics and Automation (ICRA). IEEE, 2005.

- [99] Cover T. M.: Estimation by the Nearest Neighbor Rule. 1968.
<https://pdfs.semanticscholar.org/32ba/0664f51c88960fc9b783a9c0adb6b0546e73.pdf>
[dostęp: 11.09.2018].
- [100] Solà J.: Towards Visual Localization, Mapping and Moving Objects Tracking by a Mobile Robot: a Geometric and Probabilistic Approach. 2007.
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.125.7858&rep=rep1&type=pdf> [dostęp: 11.09.2018].
- [101] Vu T. -D., Burlet J., Aycard O.: Grid-based localization and local mapping with moving object detection and tracking. *Information Fusion* 12, 2011.
- [102] Na K., Byun J., Roh M., Seo B.: RoadPlot-DATMO: Moving object tracking and track fusion system using multiple sensors. 2015 International Conference on Connected Vehicles and Expo (ICCVE), 2015.
- [103] Botha F. J.: Data Fusion of Radar and Stereo Vision for Detection and Tracking of Moving Objects. PRASA-RobMech International Conference. 2016.
- [104] Holstein T., Crnkovic G. D., Pelliccione P.: Ethical and Social Aspects of Self-Driving Cars. ARXIV. 2018.
- [105] Holstein T.: The Misconception of Ethical Dilemmas in Self-Driving Cars. *Proceedings*. 2017.
- [106] Goodall H. J.: Ethical Decision Making During Automated Vehicle Crashes. *Transportation Research Record Journal of the Transportation Research Board*. 2014.
- [107] Bellman R.: On a routing problem. 1956.
<http://www.dtic.mil/dtic/tr/fulltext/u2/606258.pdf> [dostęp: 19.09.2018].
- [108] Ford L. R.: Network Flow Theory. 1958.
<http://www.dtic.mil/dtic/tr/fulltext/u2/422842.pdf> [dostęp: 19.09.2018]
- [109] Dijkstra E. W.: A note on two problems in connexion with graphs. 1959. <http://www-m3.ma.tum.de/foswiki/pub/MN0506/WebHome/dijkstra.pdf> [dostęp: 19.09.2018]
- [110] Hart P. E., Nilsson N. J., Raphael B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *Transactions on Systems Science and Cybernetics* (4). IEEE, 1968.
- [111] Stentz A., Mellon I. C.: Optimal and efficient path planning for unknown and dynamic environments. *International Journal of Robotics and Automation*. 1993.
- [112] Dolgov D., Thrun S., Montemerlo M.: Practical Search Techniques in Path Planning for Autonomous Driving. 2008.
https://ai.stanford.edu/~ddolgov/papers/dolgov_gpp_stair08.pdf [dostęp: 19.09.2018].
- [113] Chu K., Kim j., Jo K., Sunwoo M.: Real-time path planning of autonomous vehicles for unstructured road navigation. *International Journal of Automotive Technology*. 2015.
- [114] Wang L., Zhang Y, Wang J.: Map-Based Localization Method for Autonomous Vehicles Using 3D-LIDAR. *The 25th International Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD)*. 2017.
- [115] Xiao Z., Jiang K., Xie S.: Monocular Vehicle Self-localization method based on Compact Semantic Map. 2018.
- [116] Schwarting W., Mora J. A., Rus D.: Planning and Decision-Making for Autonomous Vehicles. *Annual Reviews*. 2018.

- [117] Hoffmann G., Tomlin C. J., Montemerlo M., Thrun S.: Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing. Proceedings of the American Control Conference. IEEE, 2007.
- [118] Keviczky T., Falcone O., Borrelli F.: Predictive Control Approach to Autonomous Vehicle Steering. Transactions on Control Systems Technology. IEEE, 2007.
- [119] Kapania N. R., Gerdes J. C.: Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling. 2015. <https://www.tandfonline.com/doi/abs/10.1080/00423114.2015.1055279> [dostęp: 11.09.2018].
- [120] Sallab A. E., Abdou M., Perot E., Yogamani S.: End-to-End Deep Reinforcement Learning for Lane Keeping Assist. 2016. <https://openreview.net/pdf?id=ByjDCQgke> [dostęp: 12.09.2018].
- [121] Aoki S., Rajkumar E.: Dynamic Intersections and Self-Driving Vehicles. 2018. https://www.researchgate.net/publication/326145557_Dynamic_Intersections_and_Self-Driving_Vehicles [dostęp: 12.09.2018].
- [122] Krishnan S., Govind A. R., Ramakrishnan R.: A Look at Motion Planning for Avs at an Intersection. 2018. <https://arxiv.org/pdf/1806.07834.pdf> [dostęp: 12.09.2018].
- [123] Isele D., Rahimi R., Cosgun A.: Navigation Occluded Intersections with Autonomous Vehicles using Deep Reinforcement Learning. 2018. <https://arxiv.org/pdf/1705.01196.pdf> [dostęp: 12.09.2018].
- [124] Caltagirone L., Bellone M., Svensson M.: LIDAR-based driving path generation using fully convolutional neural networks. 2017. https://www.researchgate.net/publication/315666984_LIDAR-based_Driving_Path_Generation_Using_Fully_Convolutional_Neural_Networks [dostęp: 12.09.2018].
- [125] The KITTI Vision Benchmark Suite: Semantic and Instance Segmentation Evaluation. <http://www.cvlibs.net/datasets/kitti/> [dostęp: 12.09.2018].
- [126] Barnes D., Maddern W., Posner W.: Find your own way: weakly-supervised segmentation of path proposals for urban autonomy. International Conference on Robotics and Automation (ICRA). IEEE, 2017.
- [127] Olsson M.: Behavior Trees for decision-making in Autonomous Driving. 2016. <http://www.diva-portal.org/smash/get/diva2:907048/FULLTEXT01.pdf> [dostęp: 13.09.2018].
- [128] Hu M., Liao Y., Wang W.: Decision Tree-Based Maneuver Prediction for Driver Rear-End Risk-Avoidance Behaviors in Cut-In Scenarios. Journal of Advanced Transportation. 2017.
- [129] Claussmann L., Carvalho A., Schildbach G.: A path planner for autonomous driving on highways using a human mimicry approach with binary decision diagrams. European Control Conference. 2015.
- [130] Ulbrich S., Mauerer M.: Probabilistic Online POMDP Decision Making for Lane Changes in Fully Automated Driving. 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). IEEE, 2013.
- [131] Zhang J., Liao Y., Wang S., Han J.: Study on Driving Decision-Making Mechanism of Autonomous Vehicle Based on an Optimized Support Vector Machine Regression. Applied Sciences. 2017.
- [132] Mueller M. A.: Reinforcement Learning: MDP Applied to Autonomous Navigation. 2017. <http://aircconline.com/mlaij/V4N4/4417mlaij01.pdf> [dostęp: 13.09.2018].

- [133] Legrand N.: Deep Reinforcement Learning for Autonomous Vehicle Control among Human Drivers. 2017. https://ai.vub.ac.be/sites/default/files/thesis_legrand.pdf [dostęp: 13.09.2018].
- [134] Harris M.: How Google's Autonomous Car Passed the First U.S. State Self-Driving Test. IEEE Spectrum. IEEE, 2014. <https://spectrum.ieee.org/transportation/advanced-cars/how-googles-autonomous-car-passed-the-first-us-state-selfdriving-test> [dostęp: 12.09.2018].
- [135] Pomerleau D. A.: ALVINN: An Autonomous Land Vehicle in a Neural Network. 1989. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.830.2188&rep=rep1&type=pdf> [dostęp: 12.09.2018].
- [136] Bojarski M., Del Testa D., Dworakowski D.: End to end learning for self-driving cars. <https://arxiv.org/abs/1604.07316> [dostęp: 12.09.2018].
- [137] Hussein A., Gaber M. M., Elyan E.: Imitation Learning: A Survey of Learning Methods. ACM Computer Surveys. 2017.
- [138] Dosovitskiy A, Ros G., Codevilla F.: CARLA: An Open Urban Driving Simulator. 2017. <http://proceedings.mlr.press/v78/dosovitskiy17a/dosovitskiy17a.pdf> [dostęp: 12.09.2018].

ALGORITHMS APPLIED IN AUTONOMOUS VEHICLE SYSTEMS

Abstract. Many research centres in the world that deal with the problems of the manufacture of land vehicles, especially those intended for transport and communication in urban traffic, are still working on the development of a vehicle equipped with systems that do not require human participation in the process of driving a vehicle. The goals to be pursued are to ensure maximum safety (minimize accidents involving people) and to optimize transport costs (eliminating the driver from the vehicle, optimal route selection). This article, which is the result of broad studies conducted at OBRUM, discusses the development paths of autonomous vehicles and key algorithms of autonomous vehicles described in detail in the literature cited. The presented results constitute a starting point for further work on an autonomous vehicle to be carried out at OBRUM.

Keywords: algorithmics, autonomy, autonomous vehicle, autonomous vehicle system.