

BRUNNO GIRARDELLO DE FARIA MAIA GAPSKI

**UNIDADE DE PROCESSAMENTO GRÁFICO EM VEÍCULOS
AUTÔNOMOS**

**CURITIBA
2021**

BRUNNO GIRARDELLO DE FARIA MAIA GAPSKI

**UNIDADE DE PROCESSAMENTO GRÁFICO EM VEÍCULOS
AUTÔNOMOS**

Monografia apresentada como resultado parcial à obtenção do grau de Especialista em Engenharia Industrial 4.0. Curso de Pós-graduação Lato Sensu, Setor de Tecnologia, Departamento de Engenharia Mecânica, Universidade Federal do Paraná.

Orientador: Prof. Dr. Pablo Deivid Valle

RESUMO

Com a evolução do carro dentro da sociedade, a necessidade por conforto e segurança dentro do veículo aumentaram. A partir disso nasceram as tecnologias de assistência à condução para auxiliar o motorista. Na vanguarda do avanço tecnológico existem os veículos autônomos, que utilizam diversas câmeras, sensores, radares, GPS e lidars para se localizarem. Para processar a quantidade massiva de dados em tempo real necessários a essas tecnologias são necessários vários microprocessadores. Entretanto, descobriu-se que as unidades de processamento gráfico são mais rápidas do que os microprocessadores, consomem menos energia e são mais compactas. Isto é devido à capacidade de processamento em paralelo desses equipamentos. Dessa forma, as unidades de processamento gráfico passaram a ser peça fundamental tanto nos veículos com tecnologias de assistência avançada de condução, aos veículos de direção autônoma. A capacidade de gerar respostas em um curto tempo e ainda gerenciar a grande quantidade de dados em tempo real, principalmente aqueles que compõem imagens e pixels, tornaram as unidades gráficas item chave para alcançar os próximos níveis de direção autônoma.

Palavras-chave: Unidade de processamento gráfico; veículos autônomos; tecnologias avançadas de assistência ao motorista.

LISTA DE ILUSTRAÇÕES

FIGURA 1 - Visão geral da arquitetura do sistema de condução autônoma.....	8
FIGURA 2 - GPU/IMU localization	Erro! Indicador não definido.
FIGURA 3 - Odometria visual	Erro! Indicador não definido.
FIGURA 4 - Fluxograma	Erro! Indicador não definido.
FIGURA 5 - Rastreamento e reconhecimento de objetos	10
FIGURA 6 - Sistema operacional.....	11
FIGURA 7 - zookeeper para o Sistema Operacional.....	12
FIGURA 8 - Multicast x Broadcast.....	13
FIGURA 9 - Plataforma de simulação.....	14
FIGURA 10 - Produção de Mapa em alta definição.....	16

SUMÁRIO

1. INTRODUÇÃO	6
1.1. CONTEXTUALIZAÇÃO	6
1.2. FORMULAÇÃO DO PROBLEMA	7
1.3. JUSTIFICATIVA.....	7
1.4. OBJETIVO	7
2. REVISÃO BIBLIOGRÁFICA	8
2.1. VISÃO GERAL DAS TECNOLOGIAS DE CONDUÇÃO AUTÔNOMA.....	8
2.2. RECONHECIMENTO E RASTREAMENTO DE OBJETOS	12
2.3. SISTEMA DE CONDUÇÃO AUTÔNOMA	15
2.4. PLATAFORMA DE HARDWARE.....	17
2.5. INTRODUÇÃO À CONDUÇÃO AUTÔNOMA.....	19
3. REVISÃO BIBLIOGRÁFICA	20
3.2. GPUs vs CPUs	20
3.3. GPUs para fins genéricos	21
3.4. COMO OS CARROS AUTÔNOMOS TOMAM DECISÕES?.....	21
3.5. DESBRAVADORES.....	22
3.6. DETECÇÃO E CLASSIFICAÇÃO DE OBJETOS	23
3.7. OUTRAS DNNS	23
4. CONCLUSÕES	24
5. REFERÊNCIAS BIBLIOGRÁFICAS	26

1. INTRODUÇÃO

Após o primeiro Ford T em 1908, os automóveis se popularizaram até se tornarem um dos ambientes comuns no cotidiano das pessoas. Dessa forma, essas máquinas evoluíram de simples meios de transporte para verdadeiros centros móveis de entretenimento. Entretanto, para fornecer esse conforto e segurança são necessários recursos eletrônicos que demandam uma grande quantidade de dados para serem processados; e é neste ponto que as GPUs embarcam nos automóveis.

As unidades de processamento gráfico (*graphics processing unit* - GPUs) são chips eletrônicos responsáveis por processar as informações computacionais para gerar vídeos e imagens, que são exibidas em telas e monitores. Para compreender o funcionamento, é possível fazer uma analogia de um computador sendo uma grande empresa; e cada peça, um departamento diferente. Quando é necessário um trabalho de arte visual, para publicidade por exemplo, os funcionários enviam uma requisição para o departamento de design gráfico, a placa de vídeo. Os designers (as GPUs) decidem como criar essa imagem da melhor maneira possível e assim o colocam em um papel. No final, a ideia criada se torna uma imagem real e visível, exibida em uma tela.

1.1. CONTEXTUALIZAÇÃO

A indústria automotiva talvez seja uma das áreas onde o desenvolvimento de tecnologias é mais acelerado e o crescimento de certas demandas mundiais se dá basicamente por resultados obtidos neste setor da indústria. Vivemos uma condição global onde a conectividade é praticamente uma regra para qualquer equipamento ou gadget, desde um simples celular até mesmo um veículo programado para tomar decisões no lugar do ser humano, à partir disso percebe-se a real necessidade de se ter sistemas complexos, confiáveis e muito preciso afim de garantir que não ocorrerão falhas e/ou equívocos em tomadas de decisões, por isso a importância de se alcançar uma condição de perfeição no processamento de informações, a indústria deu um grande passo ao implantar unidades gráficas de processamento para trabalhar nesta questão envolvendo os veículos autônomos, pois elas tem maior poderio de processamento das informações do mundo externo que serão dados importantíssimos para as tomadas de decisões .

1.2. FORMULAÇÃO DO PROBLEMA

Percebemos que o desenvolvimento de veículos autônomos está em um estágio avançado e muito já foi alcançado pela indústria nesse sentido, analisando a tendência mundial de eletrificação dos meios de transporte como um todo e considerando que os combustíveis fósseis são finitos, o desenvolvimento de tecnologias para veículos autônomos tende a ser muito maior do que veículos movidos a combustão, a partir do momento que nos distanciamos da mecânica e nos aproximamos da eletrônica algumas dificuldades podem aparecer, principalmente no quesito confiabilidade, este trabalho tem como foco estudar a problemática envolvendo a questão da confiabilidade nas tomadas de decisões dos veículos autônomos a partir do momento que essas decisões são tomadas com base em dados adquiridos do meio externo por meio de sensores de diversas especificidades e processados internamente por unidades gráficas de processamento para uma rápida e precisa tomada de decisão.

1.3. JUSTIFICATIVA

A sensação de se ter uma máquina tomando uma decisão por conta própria tem dois vieses, o primeiro é a ideia de que por se tratar de uma máquina com muita tecnologia embarcada esta não é passível de falhas e, portanto, tem plena autonomia e assertividade nas tomadas de decisões, o segundo até que ponto as máquinas tem condições de tomar decisões autônomas com total acerto visto que qualquer máquina ou equipamento não se concebe sozinho, mas são produzidos por seres humanos que são falhos.

1.4. OBJETIVO

O objeto do trabalho é estudar como a utilização de unidades de processamento gráfico (GPU) ao invés de CPU como responsáveis pelo processamento de informações em veículos autônomos, tornou estes mais confiáveis e precisos.

2. REVISÃO BIBLIOGRÁFICA

2.1. VISÃO GERAL DAS TECNOLOGIAS DE CONDUÇÃO AUTÔNOMA

Conforme LIU, Shaoshan os veículos autônomos não são dotados de uma única tecnologia, mas sim de um sistema complexo que consiste de vários subgrupos, mas principalmente 3 grupos bem definidos que são: algoritmos, incluindo sensoriamento, percepção e decisão (o que requer raciocínio para casos complexos); ainda podemos incluir um sistema operacional e uma plataforma de hardware; plataforma de nuvem, incluindo mapa de alta definição (HD), treinamento de modelos de aprendizagem profunda, simulação e armazenamento de dados.

O subgrupo dos algoritmos extrai informações significativas de dados brutos de sensores para entender seu ambiente e tomar decisões sobre suas ações futuras. Os sistemas de clientes se integram esses algoritmos juntos para atender aos requisitos de tempo real e confiabilidade. Por exemplo, se a câmera gera dados a 60 Hz, os sistemas clientes precisam ter certeza de que a etapa mais longa do processamento leva menos de 16 ms para ser concluída. A plataforma em nuvem fornece computação off-line e recursos de armazenamento para carros autônomos. Com a plataforma de nuvem, somos capazes de testar novos algoritmos, atualizar mapas em alta definição e assim poder treinar melhor o reconhecimento, rastreamento e modelos de decisão.

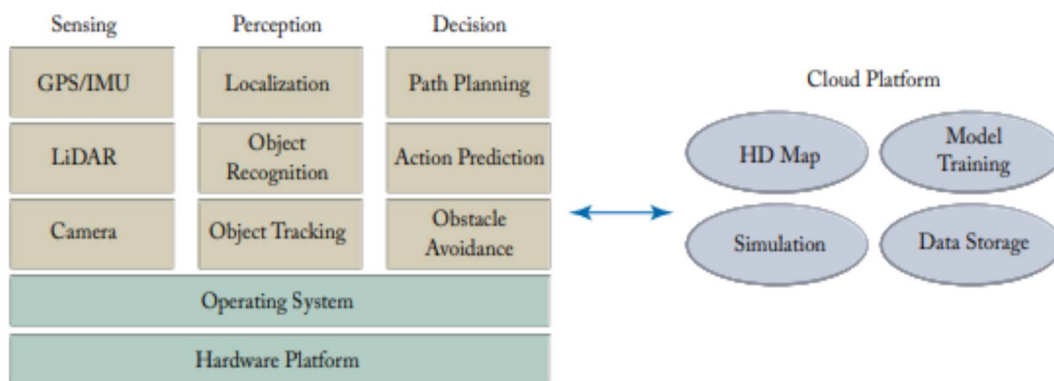


Figure 1.1: Autonomous driving system architecture overview.

Figura 1 - Visão geral da arquitetura do sistema de condução autônoma.

2.1.1 SENSORIAMENTO

Normalmente, um carro autônomo consiste em vários sensores principais. De fato, uma vez que cada tipo de sensor apresenta vantagens e desvantagens, em veículos autônomos, os dados de múltiplos sensores devem ser combinados para maior confiabilidade e segurança, baseado no princípio da redundância.

Podem ser:

- GPS/IMU: O sistema GPS/IMU ajuda o veículo autônomo a se localizar reportando atualizações inerciais e uma estimativa de posição global a uma alta taxa de atualização, por exemplo, 200 Hz. GPS é um sensor de localização bastante preciso, mas sua taxa de atualização é lenta, cerca de 10 Hz, e, portanto, não é capaz de fornecer atualizações em tempo real. No entanto, os erros da IMU se acumulam ao longo do tempo, levando a uma degradação correspondente nas estimativas de posição, propagação do erro.

No entanto, um IMU pode fornecer atualizações com mais frequência, a ou acima de 200 Hz. Isso deve satisfazer o requisito em tempo real. Combinando GPS e IMU, conseguimos fornecer atualizações precisas e em tempo real para localização aeroespacial do veículo.

- LIDAR: O LiDAR é usado para mapeamento, localização e prevenção de obstáculos. Funciona lançando um feixe de raio laser e medindo o tempo de retorno dessa informação para determinar a distância. Devido à sua alta precisão, o LIDAR pode ser usado para produzir mapas HD, para localização de um veículo em movimento, detectar obstáculos à frente e ao redor do veículo etc. Normalmente, uma boa unidade LIDAR consegue efetuar cerca de 1,3 milhões de leituras por segundo.

- Câmeras: As câmeras são usadas principalmente para tarefas de reconhecimento de objetos e rastreamento de objetos como detecção de faixa, detecção de semáforos e detecção de pedestres, etc. Para que se tenha uma segurança de interpretação do veículo geralmente utiliza-se oito ou mais Câmeras 1080p ao redor do mesmo, de tal forma que se possa detectar, reconhecer e rastrear objetos em todas as direções a partir do veículo. Essas câmeras geralmente funcionam a 60 Hz, e, quando combinadas, gerariam cerca de 1,8 GB de dados brutos por segundo.

- Radar e Sonar: O sistema de radar e sonar é usado principalmente para a última linha de defesa na prevenção de obstáculos. Os dados gerados por radar e sonar mostram a distância bem como velocidade do objeto mais próximo na frente do caminho do veículo. Uma vez que não se detecta um objeto próximo, pode haver o perigo de uma colisão, então em veículos autônomos os dados gerados a partir de radares não são processados a fundo e geralmente são usados diretamente para uma tomada de

decisão da unidade de processamento, visto que se faz necessário uma tomada de decisão extremamente rápida.

2.1.2 PERCEPÇÃO

Os dados do sensor são usados no estágio de percepção do ambiente externo ao redor do veículo. As três principais tarefas na percepção de condução autônoma são localização, detecção de objetos e rastreamento de objetos.

O GPS/IMU pode ser usado para localização e, como mencionado acima, o GPS fornece resultados de localização bastante precisos, mas com uma taxa de atualização relativamente baixa, enquanto um IMU fornece muito rápidas atualizações a um custo de resultados menos precisos. Podemos, assim, usar técnicas de Filtro Kalman para combinar as vantagens dos dois e fornecer atualizações de posição precisas e em tempo real. Como mostrado na Figura 1.2, funciona da seguinte forma: o IMU atualiza a posição do veículo a cada 5 ms, mas o erro se acumula com o tempo. Felizmente, a cada 100 ms, uma atualização gps é recebida, o que ajuda a corrigir o erro do IMU.

Ao executar este modelo de propagação e atualização, a combinação GPS/IMU pode gerar rápido e resultados precisos de localização. No entanto, não podemos confiar apenas nessa combinação de localização por três razões: (1) a precisão é de apenas cerca de um metro; (2) o sinal GPS tem multipata problemas, o que significa que o sinal pode saltar de edifícios, introduzindo mais ruído; e (3) GPS requer uma visão desobstruída do céu e, portanto, não funcionaria em ambientes como túneis

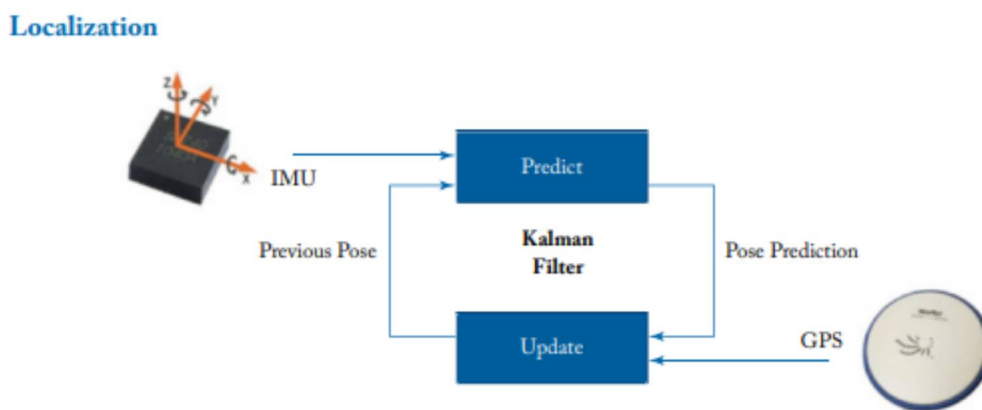


Figure 1.2: GPS/IMU localization.

Figura 2 – GPU/IMU localization

Câmeras também podem ser usadas para localização. A localização baseada em visão pode ser implementada como o seguinte pipeline simplificado: (1) triangulando pares de imagens estéreo, primeiro obtemos uma disparidade mapa que pode ser usado para derivar informações de profundidade para cada ponto; (2) combinando características salientes entre quadros de imagem estéreo sucessivos, podemos estabelecer correlações entre pontos de recurso em quadros diferentes. Poderíamos então estimar o movimento entre os dois últimos quadros; e também, (3) por comparando as características marcantes com as do mapa conhecido, também poderíamos derivar a posição atual do veículo. No entanto, essa abordagem de localização baseada em visão é muito sensível à iluminação condições e, portanto, esta abordagem por si só não seria confiável.

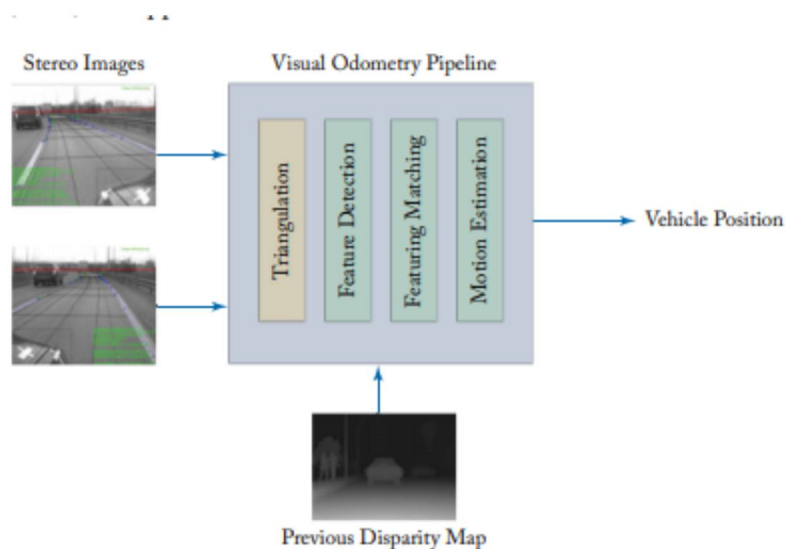


Figure 1.3: Stereo visual odometry.

Figura 3 – Odometria visual

É por isso que as abordagens lidar normalmente têm recurso para técnicas de filtro de partículas. As nuvens de ponto geradas pelo LiDAR fornecem uma "descrição da forma" do ambiente, mas é difícil para diferenciar pontos individuais. Usando um filtro de partículas, o sistema compara uma observada específica forma contra o mapa conhecido para reduzir a incerteza. Para localização de um veículo em movimento em relação a estes mapas, poderíamos aplicar um método de filtro de partículas para correlacionar as medidas LIDAR com o mapa.

O método do filtro de partículas foi demonstrado para alcançar a localização em tempo real com 10 cm precisão e ser eficaz em ambientes urbanos. No entanto, lidar tem seu próprio problema: quando há muitas partículas suspensas no ar, como gotas de chuva e poeira, as medidas podem ser extremamente barulhentas. Portanto, como mostrado na Figura 1.4, para alcançar uma localização confiável e precisa, precisamos

de um processo de fusão de sensores para combinar as vantagens de todos os sensores.

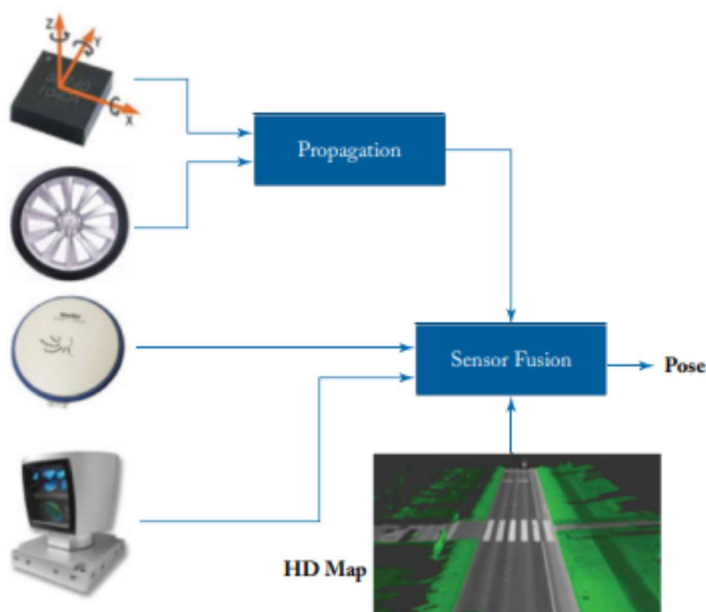


Figure 1.4: Sensor-fusion localization pipeline.

Figura 4 – Fluxograma

2.2. RECONHECIMENTO E RASTREAMENTO DE OBJETOS

Originalmente, o LiDAR era usado principalmente para executar tarefas de detecção e rastreamento de objetos em Autônomos Veículos, uma vez que o LiDAR fornece informações de profundidade muito precisas. Nos últimos anos, no entanto, eles viram o rápido desenvolvimento da tecnologia Deep Learning, que alcança um objeto significativo precisão de detecção e rastreamento. Convolution Neural Network (CNN) é um tipo de Neural Profundo

Rede que é amplamente utilizada em tarefas de reconhecimento de objetos. Um pipeline geral de avaliação da CNN geralmente consiste nas seguintes camadas. (1) A camada de convolução contém diferentes filtros para extrair características diferentes da imagem de entrada. Cada filtro contém um conjunto de parâmetros "aprendendo" que ser derivado após a fase de treinamento. (2) A camada de ativação decide se ativa o alvo neurônio ou não. (3) A camada de agrupamento reduz o tamanho espacial da representação para reduzir o número de parâmetros e, conseqüentemente, a computação na rede. (4) O Totalmente Conectado Camada onde os neurônios têm conexões completas com todas as ativações na camada anterior.

O rastreamento de objetos refere-se à estimativa automática da trajetória de um objeto à medida que ele se move. Depois que o objeto a rastrear é identificado usando técnicas de reconhecimento de objetos, o objetivo do rastreamento de objetos é rastrear automaticamente a trajetória do objeto posteriormente. Esta tecnologia pode ser usada para rastrear veículos em movimento próximos, bem como pessoas atravessando a

estrada para garantir que o veículo atual não colide com esses objetos em movimento. Nos últimos anos, técnicas de aprendizagem profunda têm demonstrado vantagens no rastreamento de objetos em comparação com as técnicas convencionais de visão computacional. Especificamente, usando imagens naturais auxiliares, um Auto-Encoder empilhado pode ser treinado offline para aprender características genéricas de imagem que são mais robustas contra variações em pontos de vista e posições do veículo.

Em seguida, o modelo treinado offline pode ser aplicado para rastreamento online.

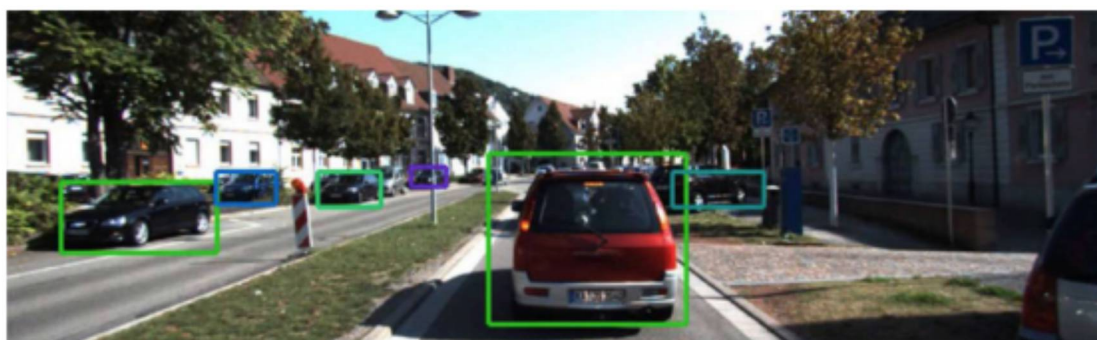


Figure 1.5: Object recognition and tracking [34], used with permission.

Figura 5 – Rastreamento e reconhecimento de objetos

2.2.1 AÇÃO

Com base na compreensão do ambiente do veículo, a fase de decisão pode gerar um cofre e plano de ação eficiente em tempo real.

Previsão de Ação

Um dos principais desafios para os motoristas humanos ao navegar pelo trânsito é lidar com as possíveis ações de outros motoristas que influenciam diretamente em sua própria estratégia de condução. Isso é especialmente verdade quando há várias faixas na estrada ou quando o veículo está em uma mudança de tráfego ponto. Para garantir que o veículo viaje com segurança nesses ambientes, a unidade de decisão gera Previsões de veículos próximos, e decide sobre um plano de ação baseado nessas previsões. Para prever ações de outros veículos, pode-se gerar um modelo estocástico dos conjuntos de posição alcançáveis dos outros participantes de tráfego, e associar esses conjuntos alcançáveis com distribuições de probabilidade

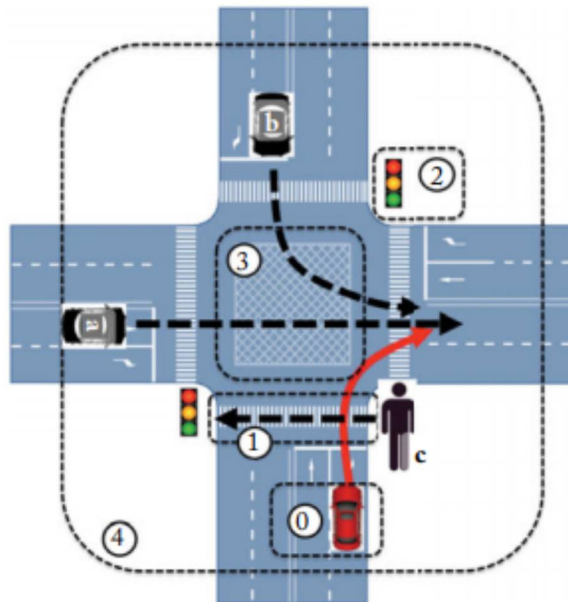


Figure 1.6: Action prediction.

Figura 5 – Previsão de ação

Planejar o caminho de um veículo autônomo e ágil em um ambiente dinâmico é um ambiente muito complexo problema, especialmente quando o veículo é obrigado a usar suas capacidades de manobra completas. Uma bruta abordagem de força seria para procurar todos os caminhos possíveis e utilizar uma função de custo para identificar o melhor caminho. No entanto, a abordagem da força bruta exigiria enormes recursos de computação e pode ser incapaz de entregar planos de navegação em tempo real. Para contornar a complexidade computacional de algoritmos determinísticos e completos, planejadores probabilísticos têm sido utilizados para fornecer planejamento de caminho em tempo real eficaz.

Prevenção de obstáculos

Como a segurança é a principal preocupação na condução autônoma, geralmente empregamos pelo menos dois níveis de mecanismos de prevenção de obstáculos para garantir que o veículo não colidisse com obstáculos. O primeiro nível é proativo, e é baseado em previsões de tráfego. Em tempo de execução, o mecanismo de previsão de tráfego gera medidas como tempo de colisão ou distância mínima prevista, e com base nessas informações, o mecanismo de prevenção de obstáculos é acionado para realizar o reequivação de caminhos local. Se o mecanismo proativo falha, o segundo nível, o mecanismo reativo, usando dados de radar, seria assumir. Uma vez que o radar detecta um obstáculo à frente do caminho, ele substituiria o controle atual para evitar os obstáculos.

2.3. SISTEMA DE CONDUÇÃO AUTÔNOMA

Os sistemas clientes integram os algoritmos acima mencionados para atender aos requisitos de tempo real e confiabilidade. Alguns dos desafios são os seguintes: o sistema precisa garantir que o pipeline de processamento é rápido o suficiente para processar a enorme quantidade de dados de sensores gerados; se uma parte do sistema falha, deve ser suficientemente robusto para se recuperar da falha; e, além disso, precisa realizar toda a computação sob restrições estritas de energia e recursos.

2.3.1 SISTEMA OPERACIONAL ROBÔ (ROS)

ROS é uma poderosa estrutura de computação distribuída sob medida para aplicações robóticas, e tem sido amplamente utilizado. Como mostrado na Figura 1.7, cada tarefa robótica (como a localização), está hospedada em um Node ROS. Os nós ROS podem se comunicar entre si através de tópicos e serviços. É um grande sistema operacional para condução autônoma, exceto que sofre de vários problemas: (1) confiabilidade: ROS tem um único mestre e nenhum monitor para recuperar nós com falha; (2) desempenho: ao enviar mensagens de transmissão, ele duplica a mensagem várias vezes, levando à degradação do desempenho; e (3) segurança: não possui mecanismos de autenticação e criptografia. Embora ROS 2.0 prometa corrigir esses problemas, o ROS 2.0 em si não foi amplamente testado e muitos recursos ainda não são disponíveis. Portanto, para usar o ROS na condução autônoma, precisamos resolver esses problemas primeiro.

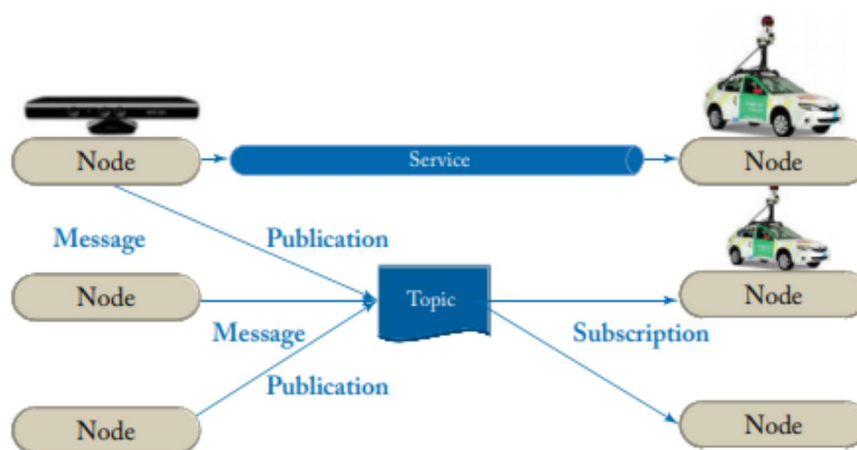


Figure 1.7: Robot operating system (ROS).

Figura 6 – Sistema operacional

A implementação ROS atual tem apenas um nó mestre, quando o nó mestre trava, o todo o sistema iria travar. Isso não atende ao requisito de segurança para condução autônoma. Para corrigir esse problema, implementamos um mecanismo semelhante ao ZooKeeper em ROS. Como mostrado na Figura 1.8, neste design, temos um nó mestre principal e um nó mestre de backup. No caso do nó principal falha, o nó de backup assumiria, certificando-se de que o sistema ainda funciona sem soluços.

Além disso, este mecanismo ZooKeeper monitora e reinicia quaisquer módulos com falha, certificando-se de que o todo o sistema ROS é confiável.

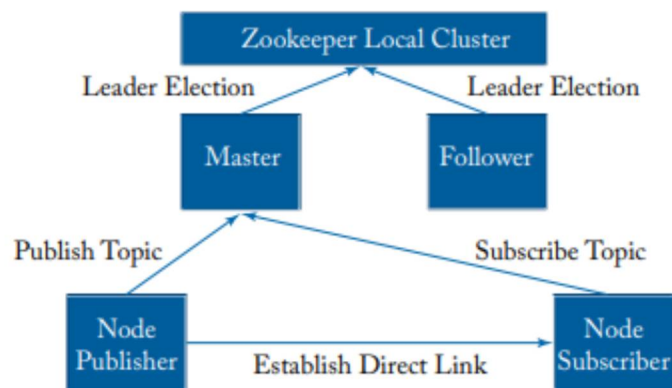


Figure 1.8: Zookeeper for ROS.

Figura 7 – zookeeper para o Sistema Operacional

Desempenho

O desempenho é outro problema com a implementação atual do ROS. Os nós ROS se comunicam com frequência, e é imperativo garantir que a comunicação entre nós seja eficiente. Primeiro, ele passa pelo mecanismo de loop-back quando os módulos locais se comunicam entre si. Cada vez ele passa pelo pipeline loop-back, uma sobrecarga de 20 ms é introduzida. Para eliminar essa sobrecarga, para sobrecarga de comunicação de nó local, usamos mecanismo de memória compartilhada de tal forma que a mensagem não precisa passar pela pilha TCPIP para chegar ao nó de destino. Segundo, quando um nó ROS transmite uma mensagem, a mensagem é copiada várias vezes, consumindo significativo largura de banda no sistema. Como mostrado na Figura 1.9, mudando para mecanismo multicast, que muito melhorou o throughput do sistema

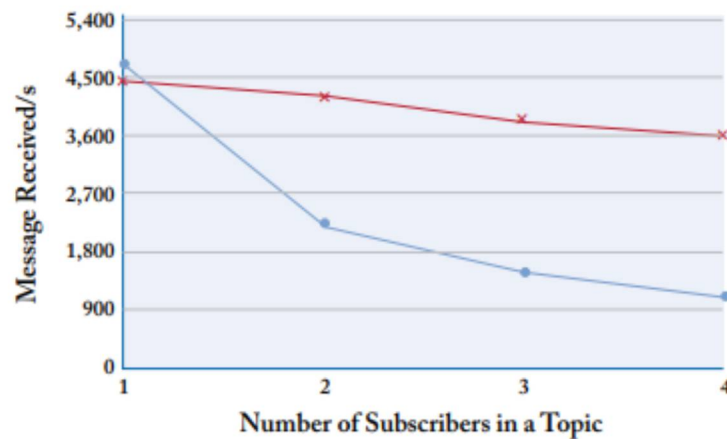


Figure 1.9: Multicast vs. broadcast in ROS.

Figura 8 – Multicast x Broadcast

Segurança

Como sabemos, a segurança é a preocupação mais crítica para a ROS. Agora imagine dois cenários: no primeiro cenário, um nó ROS é sequestrado e continuamente aloca memória até que o sistema se esgota de memória e começa a matar outros nódulos ROS. Neste cenário, o hacker trava com sucesso o sistema. No segundo cenário, uma vez que por padrão as mensagens ROS não são criptografadas, um hacker pode escutar facilmente a mensagem entre nós e aplica ataques man-in-the-middle. Para corrigir o primeiro problema, podemos usar o Linux Container (LXC) para restringir a quantidade de recursos usados por cada nó, e também para fornecer um mecanismo de caixa de areia para proteger o nó uns dos outros, portanto efetivamente impedindo o vazamento de recursos. Para corrigir o segundo problema, podemos criptografar mensagens na comunicação, impedindo que mensagens sejam escutadas.

2.4. PLATAFORMA DE HARDWARE

Para entender os desafios na concepção de plataforma de hardware para condução autônoma, vamos examinar a implementação da plataforma de computação de uma empresa líder em direção autônoma. Ela consiste em duas caixas de computação, cada uma equipada com um processador Intel Xeon E5 e quatro a oito Aceleradores de GPU Nvidia K80. A segunda caixa de computação executa exatamente as mesmas tarefas e é usada para confiabilidade: caso a primeira caixa falhe, a segunda caixa pode assumir imediatamente. Na pior das hipóteses, quando ambas as caixas funcionam em seu pico, isso significaria mais de 5000 W de consumo de energia que consequentemente geraria uma enorme quantidade de calor. Além disso, cada caixa custa \$20.000-\$30.000, tornando toda a solução inacessível para os consumidores médios. A energia, a dissipação de calor e os requisitos de custo deste projeto impedem

que a condução autônoma chegue ao público em geral. Para explorar as bordas do envelope e entender o quão bem um sistema de condução autônoma poderia funcionar em um SOC móvel ARM, implementamos um simplificado, sistema de condução autônoma baseado em visão em um SoC móvel baseado em ARM com consumo máximo de energia de 15 W. Como se vê, o desempenho foi próximo aos nossos requisitos: a localização o pipeline era capaz de processar 25 imagens por segundo, quase acompanhando uma taxa de geração de imagens de 30 imagens por segundo. O oleoduto de aprendizagem profunda era capaz de executar 2-3 objetos tarefas de reconhecimento por segundo. O oleoduto de planejamento e controle poderia planejar um caminho dentro de 6 ms. Com este sistema, conseguimos dirigir o veículo a cerca de 5 mph sem qualquer perda de localização

2.4.1 PLATAFORMA DE NUVEM DE CONDUÇÃO AUTÔNOMA

Os veículos autônomos são sistemas móveis e, portanto, precisam de uma plataforma em nuvem para fornecer suportes. As duas principais funções fornecidas pela nuvem incluem computação distribuída e armazenamento distribuído. Possui várias aplicações, incluindo simulação, que é usada para verificar novos algoritmos; Produção de mapas HD; e treinamento de modelo de aprendizagem profunda. Para construir tal plataforma, usamos a Spark para computação distribuída, OpenCL para computação heterogênea e Alluxio para armazenamento na memória. Conseguimos fornecer uma condução autônoma confiável, de baixa latência

2.4.2 SIMULAÇÃO

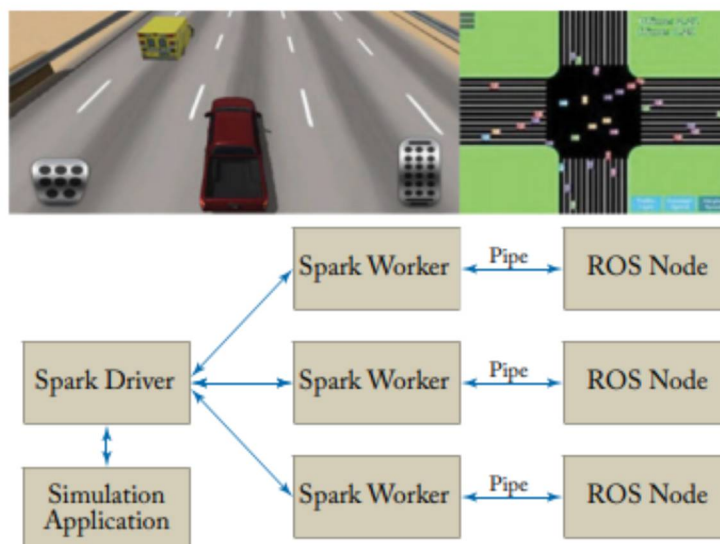


Figure 1.10: Spark and ROS-based simulation platform.

Figura 9 – Plataforma de simulação

2.5. INTRODUÇÃO À CONDUÇÃO AUTÔNOMA

A primeira aplicação de tal sistema é a simulação. Quando desenvolvemos um novo algoritmo, e precisamos para testá-lo completamente antes que possamos implantá-lo em carros. Se fôssemos testá-lo em carros de verdade, o custo seria ser enorme e o tempo de volta seria muito longo. Portanto, geralmente testamos em simuladores, como reproduzir dados através de nós ROS. No entanto, se fôssemos testar o novo algoritmo em uma única máquina, ou vai demorar muito, ou não temos cobertura de teste suficiente. Como mostrado na Figura 1.10, para resolver este problema, desenvolvemos uma plataforma de simulação distribuída. De tal forma que usamos spark para gerenciar nós de computação distribuída, e em cada nó, executamos um ROS exemplo de repetição. Em um conjunto de teste de reconhecimento de objeto de condução autônoma que usamos, levou 3 horas para executado em um único servidor; usando o sistema distribuído que desenvolvemos, o teste terminou dentro de 25 min quando escalamos para 8 máquinas.

2.5.1 PRODUÇÃO DE MAPA HD

Como mostrado na Figura 1.11, a produção de mapas HD é um processo complexo que envolve muitas etapas, incluindo processamento de dados brutos, produção de nuvem de pontos, alinhamento de nuvem de pontos, mapa de reflectância 2D geração, rotulagem de mapa HD, bem como a geração final de mapas. Usando a Centelha, conectamos todas essas etapas juntos em um trabalho spark. Mais importante, o Spark fornece um mecanismo de computação na memória, de tal forma que não temos que armazenar os dados intermediários em disco rígido, portanto, muito reduzindo o desempenho do processo de produção do mapa.

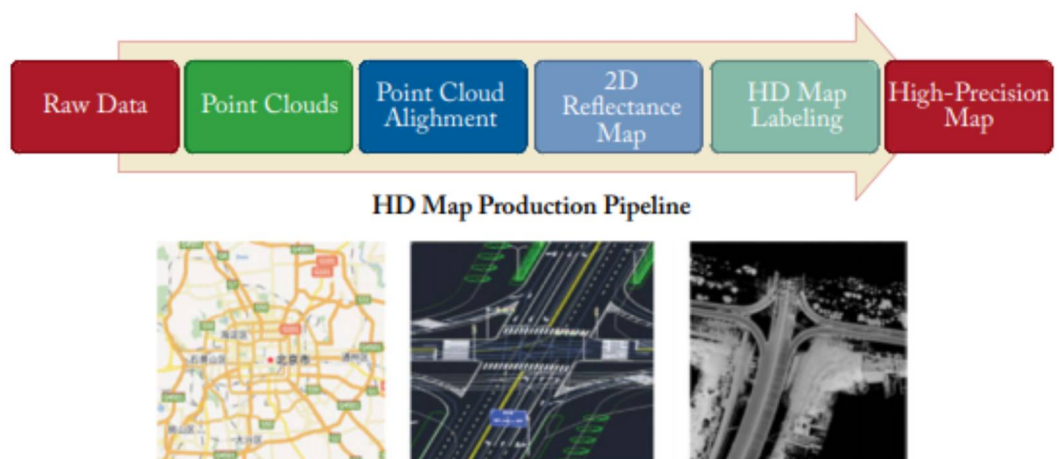


Figure 1.11: Cloud-based HD Map production.

3. REVISÃO BIBLIOGRÁFICA

3.1. Conforme McLeish a utilização de GPU's em veículos se dá pela sua capacidade de processar rapidamente cálculos matemáticos associados a imagens 3D, as GPUs são capazes de processar dados gerais de forma paralela, ao invés de sequencial como as CPUs. Essa característica reduz o tempo de execução de algoritmos paralelos e aumenta a velocidade de resposta em tempo real para os sistemas avançados de assistência ao motorista (*advanced driver-assistance systems-ADAS*).

3.2. GPUs vs CPUs

Ainda, As unidades de processamento são o cérebro dos computadores, pois são dentro desses itens os cálculos e comandos são realizados. As capacidades lógicas e configurações determinam se o item é uma unidade central de processamento (CPU) ou uma unidade de processamento gráfico (GPU).

As CPUs são lineares, pois processam as informações em uma simples metodologia de instrução em um cenário de “se/então” (if/then). Normalmente são alojadas em ambientes estáticos, sem exposição a elementos externos, variações de temperaturas e vibrações.

As GPUs são normalmente utilizadas para funcionalidades avançadas de físicas, texturas, sombras e múltiplas cores. Os processadores gráficos operam em computação paralela através do método SIMD (*single instruction, multiple data*), em que a mesma instrução é aplicada simultaneamente a diversos dados e assim produz mais resultados. Dessa forma, as GPUs conseguem processar os dados em paralelo mais rapidamente do que as operações lineares das CPUs.

3.3. GPUs para fins genéricos

Dessa forma, como as GPUs são inerentemente rápidas em gerenciar pixels e imagens, as informações em pixels recebidas pelas câmeras e sensores (dados que são naturalmente paralelos) podem ser enviadas à GPUs para serem processadas. Assim, as unidades de processamento gráfico passam a ser utilizadas para gerenciar uma quantidade massiva de dados que podem ser processados paralelamente para otimizar o tempo de resposta – requisito crucial para os veículos autônomos.

Porém, esses dados não são necessariamente gráficos, e assim, as GPUs passam a ser utilizadas para fins genéricos (general purpose graphics processing unit

– GPGPU). Alguns exemplos são: dados de posicionamento de GPS, acelerômetros, radares, sensores de câmeras e LIDARs para guiar os carros autônomos em segurança.

3.4. COMO OS CARROS AUTÔNOMOS TOMAM DECISÕES?

Katie Burke fala que os carros autônomos enxergam o mundo através de sensores, e a chave para essa interpretação desses dados recebidos é a percepção, o termo do setor para a capacidade, durante a condução, de processar e identificar dados rodoviários, placas, detectar pedestres ou animais e o tráfego circundante. Com o poder da inteligência artificial os veículos sem motorista podem reconhecer e reagir ao ambiente em tempo real, permitindo uma navegação segura.

Eles fazem isso usando uma série de algoritmos conhecidos como redes neurais profundas ou DNNs, ao invés de exigir um conjunto de regras e condições de programação feito por um humano para o carro tomar decisões, como “pare se vir vermelho”, os DNNs permitem que os veículos aprendam a navegar pelo mundo por conta própria usando dados de sensores. Esses modelos matemáticos são inspirados no cérebro humano - eles aprendem com a experiência. Se um DNN for mostrado várias imagens de sinais de parada

em condições variadas, ele pode aprender a identificar os sinais de parada por conta própria.

Apenas um algoritmo não pode fazer o trabalho sozinho, um conjunto completo de DNNs, cada um dedicado a uma tarefa específica, é necessário para uma direção autônoma segura, essas redes são diversas, cobrindo tudo, desde a leitura de sinais até a identificação de cruzamentos e a detecção de caminhos para carros. Eles também são redundantes, com recursos sobrepostos para minimizar as chances de falha. Não há um número definido de DNNs necessários para a direção autônoma. E novos recursos surgem com frequência, de modo que a lista está crescendo e mudando constantemente, é uma constante atualização, para realmente dirigir o carro, os sinais gerados pelos DNNs individuais devem ser processados em tempo real. Isso requer uma plataforma de computação centralizada e de alto desempenho, alcançada por GPU's dedicadas de última geração.

Alguns exemplos dos principais DNNs que a NVIDIA usa para percepção de veículos autônomos.

3.5. DESBRAVADORES

DNNs que ajudam o carro a determinar onde ele pode dirigir e planejar com segurança o caminho a seguir:

O OpenRoadNet identifica todo o espaço dirigível ao redor do veículo, independentemente de ser na pista do carro ou em pistas vizinhas.

PathNet destaca o caminho dirigível à frente do veículo, mesmo se não houver marcadores de faixa.

LaneNet detecta linhas de faixa e outros marcadores que definem o caminho do carro.

O MapNet também identifica pistas e pontos de referência que podem ser usados para criar e atualizar mapas de alta definição.

DNNs de localização de caminhos trabalham juntos para identificar uma rota de direção segura para um veículo autônomo.

3.6. DETECÇÃO E CLASSIFICAÇÃO DE OBJETOS

DNNs que detectam possíveis obstáculos, bem como semáforos e sinais:

O DriveNet percebe outros carros na estrada, pedestres, semáforos e sinais, mas não lê a cor do sinal ou tipo de sinal.

O LightNet classifica o estado de um semáforo - vermelho, amarelo ou verde.

SignNet distingue o tipo de sinal - parar, rendimento, mão única, etc.

WaitNet detecta condições em que o veículo deve parar e esperar, como cruzamentos.

3.7. OUTRAS DNNS

DNNs que podem detectar o estado das partes do veículo e da cabine, bem como facilitar manobras como estacionamento:

Clear Sight Net monitora quão bem as câmeras do veículo podem ver, detectando condições que limitam a visão, como chuva, neblina e luz solar direta.

ParkNet identifica vagas disponíveis para estacionamento.

Essas redes são apenas uma amostra das DNNs que constituem a camada de percepção do DRIVE Software redundante responsável pelas decisões.

4. CONCLUSÕES

Para dirigir melhor do que os humanos, os veículos autônomos devem primeiro ver melhor do que os humanos tudo que está ocorrendo a sua volta e para isso devem possuir sistemas confiáveis e este tem sido o maior obstáculo dos desenvolvedores. Há a necessidade de combinar diversas tecnologias e desta forma criar um complexo sistema que acima de tudo seja a prova de falhas.

A grande chave para essa tecnologia está na escolha dos tipos de sensores e a capacidade de processamento do todo o conjunto. A resposta está na diversidade, ou seja, um sistema capaz de fazer detecções por diferentes métodos, lançando mão até de redundância de sinal fazendo sobreposição de sensores a fim de garantir que a informação que chegará até a GPU seja fidedigna e confiável, e assim, o veículo tomará uma decisão correta sem ocasionar um eventual acidente.

Entendemos que o futuro é das máquinas autônomas. Porém, quando se fala em veículos a situação é muito delicada, pois é um ambiente onde não se permite falhas, visto que uma falha pode custar uma vida humana. Isto posto, para que essa realidade do veículo autônomo ganhe corpo e seja uma condição normal na humanidade ainda precisamos de desenvolvimento na área, principalmente no que diz respeito a confiabilidade, assertividade e velocidade na tomada de decisões por parte de todo conjunto envolvido, ou seja, sensores, atuadores, unidades de processamento gráfico, mecanismos eletrônicos, mecânicos e hidráulicos, por fim todo o conjunto de um veículo autônomo.

Uma vez que não se tem plena convicção e segurança a ponto de permitir que um veículo autônomo faça uma viagem sozinho sem um humano supervisionando, será que em algum momento por mais avançada que esteja a tecnologia, o veículo autônomo terá essa Inteligência artificial e capacidade gráfica de suas GPU's apurada e calibrada ao ponto de fazer o papel do motorista sem restrições?

Em cima do exposto podemos enfim concluir que quanto mais aprimorado for o desenvolvimento das Gpu's, maior será o grau de confiabilidade e segurança que poderemos depositar nos veículos autônomos, o foco está justamente em ter total garantia dos seus movimentos e tomada de decisões, que sempre estarão baseados em informações adquiridas do meio ao seu redor, por intermédio de sensores já citados e com o processamento dessas informações por uma GPU dedicada, ou seja tudo está baseado no processamento gráfico.

5. REFERÊNCIAS BIBLIOGRÁFICAS

LIU, Shaoshan. **Creating Autonomous Vehicle Systems**. 2018. Morgan & Claypool Publishers

CORPORATION, Nvidia. <https://resources.nvidia.com/en-us-drive-overview/how-a-self-driving-car-makes-decisions-blog>. Acesso em 05.04.2021

CORPORATION, Nvidia. <https://images.nvidia.com/content/self-driving-cars/safety-report/auto-print-self-driving-safety-report-2021-update.pdf>
acesso em 03.04.2021

CROW, Thomas Scott. **Evolution of the Graphical Processing Unit**. 2004. 53 f. Tese (Doutorado) - Curso de Master Of Science With A Major In Computer Science, University Of Nevada, Reno, 2004.

CORPORATION, Nvidia. **NVIDIA Introduces DRIVE AGX Orin: advanced, software- defined platform for autonomous machines**. Advanced, Software-Defined Platform for Autonomous Machines. 2019. Disponível em: <https://nvidianews.nvidia.com/news/nvidia-introduces-drive-agx-orin-advanced-software-defined-platform-for-autonomous-machines>. Acesso em: 22 março 2021.

MCLEISH, James. **Are GPUs Reliable Enough To Be An Autonomous Vehicle's Brain?** 2016. Disponível em: <https://www.dfrsolutions.com/blog/are-gpus-reliable-enough-to-be-an-autonomous-vehicles-brain>. Acesso em: 18 março 2021.

TAO, Benson. **The Importance of Graphics and GPGPU in ADAS and Other Automotive Applications**. 2013. Disponível em: https://bensontao.wordpress.com/2013/06/11/gpu_in_automotive/. Acesso em: 20 março 2021.

SWISSA, Lior. **Nvidia and the GPU: contribution to the AI world of self-driving cars**. 2017. Disponível em: <https://becominghuman.ai/nvidia-and->

the-gpu- contribution-to-the-ai-world-of-self-driving-cars-1f00e3212508.
Acesso em: 19 março de 2021.

TYSON, Jeff; WILSON, Tracy V. **How Graphics Cards Work**. Disponível em: <https://computer.howstuffworks.com/graphicsard.htm>. Acesso em: 20 março 2021.