UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

VINÍCIUS ARAÚJO SANTOS

SiameseVO-Depth: Odometria Visual através de redes neurais convolucionais siamesas

Goiânia
2018

sistema de bibliotecas ufg

PRPG
PRÓ-REITORIA DE PÓS-GRADUAÇÃO

UFG

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

1. Identificação do material bibliográfico: **[X] Dissertação** **[ ] Tese**

2. Identificação da Tese ou Dissertação:

Nome completo do autor: Vinícius Araújo Santos

Título do trabalho: SiameseVO-Depth: Odometria Visual através de redes neurais convolucionais siamesas

3. Informações de acesso ao documento:

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF da tese ou dissertação.

Vinícius Araújo Santos
Assinatura do(a) autor(a)[2]

Ciente e de acordo:

Assinatura do(a) orientador(a)[2]

Data: 20 / 11 / 2018

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

Casos de embargo:
- Solicitação de registro de patente
- Submissão de artigo em revista científica
- Publicação como capítulo de livro
- Publicação da dissertação/tese em livro

[2] A assinatura deve ser escaneada.

VINÍCIUS ARAÚJO SANTOS

SiameseVO-Depth: Odometria Visual através de redes neurais convolucionais siamesas

Dissertação apresentada ao Programa de Pós–Graduação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Programa de Pós-Graduação em Ciência da Computação.

Área de concentração: Ciência da Computação.

Orientador: Prof. Dr. Gustavo Teodoro Laureano

Goiânia
2018

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Araújo Santos, Vinícius
SiameseVO-Depth: Odometria Visual através de redes neurais convolucionais siamesas [manuscrito] / Vinícius Araújo Santos. - 2018.
73 f.

Orientador: Prof. Dr. Gustavo Teodoro Laureano.
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2018.
Bibliografia.
Inclui siglas, gráfico, tabelas, lista de figuras, lista de tabelas.

1. Odometria Visual. 2. Deep Learning. 3. Redes Convolucionais Siamesas. 4. Visão Computacional. I. Teodoro Laureano, Gustavo, orient. II. Título.

CDU 004

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

UFG

ATA Nº 21/2018

ATA DA SESSÃO DE JULGAMENTO DA DISSERTAÇÃO DE MESTRADO DE VINÍCIUS ARAÚJO SANTOS

Aos onze dias do mês de outubro de dois mil e dezoito, às catorze horas, na sala 150 do Instituto de Informática da Universidade Federal de Goiás, Campus Samambaia, reuniu-se a banca examinadora designada na forma regimental pela Coordenação do Curso para julgar a dissertação de mestrado intitulada **"SiameseVO-Depth: Odometria Visual através de redes neurais convolucionais siamesas"**, apresentada pelo aluno Vinícius Araújo Santos como parte dos requisitos necessários à obtenção do grau de Mestre em Ciência da Computação, área de concentração Ciência da Computação. A banca examinadora foi presidida pelo orientador do trabalho de dissertação, Professor Doutor Gustavo Teodoro Laureano (INF/UFG), tendo como membros os Professores Doutores Anderson da Silva Soares (INF/UFG) e Clarimar José Coelho (PUC-GO). Aberta a sessão, o candidato expôs seu trabalho. Em seguida, o aluno foi arguido pelos membros da banca e:

(X) tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de sua dissertação, a banca concluiu pela **aprovação** do candidato, sem restrições.

( ) não tendo demonstrado suficiência de conhecimento e capacidade de sistematização do tema de sua dissertação, a banca concluiu pela **reprovação** do candidato.

Os trabalhos foram encerrados às 16 horas. Nos termos do Regulamento Geral dos Cursos de Pós-Graduação desta Universidade, lavrou-se a presente ata que, lida e julgada conforme, segue assinada pelos membros da banca examinadora.

Prof. Dr. Gustavo Teodoro Laureano _____

Prof. Dr. Anderson da Silva Soares _____

Prof. Dr. Clarimar José Coelho _____

Vinícius Araújo Santos

Engenheiro de computação pela Universidade Federal de Goiás. Membro do núcleo de pesquisa em robótica Pequi Mecânico desde 2013. Possui experiência com Robótica, Aprendizado de Máquina, Visão Computacional, Processamento Digital de Imagens e Inteligência Computacional. Participa da categoria IEEE Very Small Size Soccer atuando na área de visão computacional robótica. Durante o Mestrado foi bolsista CAPES com pesquisa na área de visão computacional e veículos autônomos voltando-se à técnica de Odometria Visual.

Este trabalho é dedicado a todos aqueles que insistem em sonhar e creem que o melhor ainda está por vir...

Agradecimentos

A Deus, supremo criador e maestro do universo. Sozinho nada poderia fazer mas por sua força posso vencer as barreiras que se ponham diante de mim. Posso tudo Nele, que me fortalece.

Agradeço ao meu orientador Dr. Gustavo Teodoro pelo apoio e direcionamentos que tornaram esta pesquisa possível e pela amizade persistente nos momentos de angústia. Também agradeço ao Dr. Anderson Soares por apresentar-me ao tema de *Deep Learning* que não só têm revolucionado o mundo mas revolucionou a minha vida. Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior que me concedendo o *status* de bolsista permitiu este trabalho ser concluído.

À minha família, minha gratidão pela paciência e força. Agradeço aos meus pais, Josué Santos e Rosineide Araújo por todo o amor e por lutarem sempre para que hoje pudesse ser quem sou. Agradeço aos meus irmãos Jessé Santos e Larissa Araújo pelo amor, amizade e companheirismo ao longo desta caminhada. À minha cunhada, Andrieli pela amizade e meu sobrinho Davi que mal posso esperar para tomar em meus braços.

Agradeço aos amigos que me acompanham em minha trajetória acadêmica. Aos meus companheiros de luta, Vinícius Paulo, Alexandre Cadore e Lucas Assis pela amizade e conselhos. Aos amigos Leonel Filho, Apolo Marton e Alexandre Monteiro pela longa amizade, lealdade e o incentivo frente aos desafios desta jornada.

À minha amiga e psicóloga, Marineide Almeida pelo incentivo e por me levar a enxergar força e alegrias mesmo em situações adversas. Por ensinar que quando acreditamos que na capacidade dada pelo Criador podemos enfrentar até as maiores dificuldades e vencê-las.

Também agradeço aos amigos do Núcleo de Robótica Pequi Mecânico da Universidade Federal por me proporcionarem experiências incríveis e me direcionar ao tema deste trabalho. Por fim, agradeço a todos da empresa Data H pelas oportunidades e apoio na conclusão de mais uma fase em direção à consolidação da minha vida profissional.

"Ó profundidade da riqueza da sabedoria e do conhecimento de Deus! Quão insondáveis são os seus juízos, e inescrutáveis os seus caminhos!"

Bíblia Sagrada,
Romanos 11:33.

Resumo

SANTOS, VINÍCIUS ARAÚJO. **SiameseVO-Depth: Odometria Visual através de redes neurais convolucionais siamesas**. Goiânia, 2018. 72p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

Odometria Visual é um importante processo na navegação de robôs baseada em imagens. Os métodos clássicos deste tema dependem de boas correspondências de características feitas entre imagens sendo que a detecção de características em imagens é um tema amplamente discutido no campo de Visão Computacional. Estas técnicas estão sujeitas a problemas de iluminação, presença de ruído e baixa de acurácia de localização. Nesse contexto, a informação tridimensional de uma cena pode ser uma forma de mitigar as incertezas sobre as características em imagens. Técnicas de *Deep Learning* têm demonstrado bons resultados lidando com problemas comuns em técnicas de OV como insuficiente iluminação e erros na seleção de características. Ainda que já existam trabalhos que relacionam Odometria Visual e *Deep Learning*, não foram encontradas técnicas que utilizem Redes Convolucionais Siamesas com sucesso utilizando informações de profundidade de mapas de disparidade durante esta pesquisa. Este trabalho visa preencher esta lacuna aplicando *Deep Learning* na estimativa do movimento por de mapas de disparidade em uma arquitetura Siamesa. A arquitetura *SiameseVO-Depth* proposta neste trabalho é comparada à técnicas do estado da arte em OV utilizando a base de dados *KITTI Vision Benchmark Suite*. Os resultados demonstram que através da metodologia proposta é possível a estimativa dos valores de uma Odometria Visual ainda que o desempenho não supere técnicas consideradas estado da arte. O trabalho proposto possui menos etapas em comparação com técnicas clássicas de OV por apresentar-se como uma solução fim-a-fim e apresenta nova abordagem no campo de *Deep Learning* aplicado à Odometria Visual.

Palavras–chave

Odometria Visual, Visão Computacional, Deep Learning, Redes Convolucionais Siamesas.

Abstract

SANTOS, VINÍCIUS ARAÚJO. **SiameseVO-Depth: Odometria Visual através de redes neurais convolucionais siamesas**. Goiânia, 2018. 72p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

Visual Odometry is an important process in image based navigation of robots. The standard methods of this field rely on the good feature matching between frames where feature detection on images stands as a well adressed problem within Computer Vision. Such techniques are subject to illumination problems, noise and poor feature localization accuracy. Thus, 3D information on a scene may mitigate the uncertainty of the features on images. Deep Learning techniques show great results when dealing with common difficulties of VO such as low illumination conditions and bad feature selection. While Visual Odometry and Deep Learning have been connected previously, no techniques applying Siamese Convolutional Networks on depth infomation given by disparity maps have been acknowledged as far as this work's researches went. This work aims to fill this gap by applying Deep Learning to estimate egomotion through disparity maps on an Siamese architeture. The SiameseVO-Depth architeture is compared to state of the art techniques on OV by using the KITTI Vision Benchmark Suite. The results reveal that the chosen methodology succeeded on the estimation of Visual Odometry although it doesn't outperform the state-of-the-art techniques. This work presents fewer steps in relation to standard VO techniques for it consists of an end-to-end solution and demonstrates a new approach of Deep Learning applied to Visual Odometry.

Keywords

Visual Odometry, Computer Vision, Deep Learning, Siamese Convolutional Neural Network.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Siglas

OV	Odometria Visual
VC	Visão Computacional
RANSAC	Random Sample Consensus
CNN	Convolutional Neural Network
SCNN	Siamese Convolutional Neural Network
GPU	Graphical Processing Unit
LSTM	Long-Short Term Memory
ReLU	Rectified Linear Unit
PCBP	Particle convex belief Propagation
VRAM	Video Random Access Memory
RTK	Real Time Kinematic
INS	Inertial Navigation System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LS-VO	Latent Space Visual Odometry
SLAM	Simultaneous Localization and Mapping
VSLAM	Visual Simultaneous Localization and Mapping
RGB	Red, Green and Blur
ATE	Absolute Trajectory Error

CAPÍTULO 1

Introdução

1.1 Contextualização

Para estimar seu movimento, robôs móveis normalmente estão equipados com dispositivos e técnicas que mensuram seu deslocamento, seja pelo uso de sensores de rotação suas rodas, denominada odometria por rodas, sensores de distância baseados em *laser* e ultrassom, Sistema de Navegação Inercial (*INS, Inertial Navigation System*) ou por Sistema de Posicionamento Global (*GPS*, *Global Positioning System*) [66].

Sensores inerciais utilizam dispositivos como acelerômetros e giroscópios para estimar a posição e orientação de um objeto. Aqel et al. [5] pontuam que sensores *INS* são altamente suscetíveis ao acúmulo de erros considerando que sensores de alta precisão são caros e o que limita aplicações comerciais. Pequenos erros nas medidas de velocidades angulares e aceleração são convertidos em grandes erros de velocidade linear. Consequentemente, os erros de posição tornam-se ainda maiores inviabilizando aplicações de navegação em que longas distâncias são percorridas.

Na odometria com lasers, o movimento de um agente é estimado através de correspondências entre valores medidos pelos lasers. Cada ponto detectado em consecutivos escaneamentos é utilizado para construir uma odometria, ideia similar ao que é realizado em odometria por imagens. [83] e [5] indicam que uma dos fatores que limitam o uso de sensores laser é o alto custo financeiro. É citado que a análise dos dados gerados demandam alto custo computacional limitando o uso em aplicações em tempo real. Estas limitações apresentam-se principalmente na tarefa de encontrar correspondência entre medidas feitas pelos lasers onde erros podem ocorrem muitas vezes em função da composição do material.

A odometria baseada em imagens, conhecida como Odometria Visual (OV), tem se mostrado uma alternativa aos métodos tradicionais devido aos avanços alcançados na área de Visão Computacional (VC) e pelo aumento da capacidade computacional de sistemas embarcados que permite o processamento de imagens em tempo real. Na odometria por imagens, o movimento de um agente é recuperado a partir de uma

sequência de transformações projetivas, calculadas a partir de quadros capturados durante o percurso da câmera [36].

A recuperação do movimento de uma câmera é um problema conhecido na área de *VC* e possui intersecção com os estudos em *Structure from Motion*, conhecido pela sigla SFM, uma área que trata da estimação da estrutura tridimensional de um conjunto de pontos do ambiente a partir de seu movimento. Balazadegan et al. [6] indicam que a OV é um caso específico de SFM onde além da pose da câmera, é feita a reconstrução 3D de um ponto no espaço. Pode-se dizer que a Odometria Visual (OV) é o processo de determinar a orientação e a posição de um agente a partir de uma sequência de imagens consecutivas geradas por uma ou mais câmeras fixadas a ele [23], muitas vezes com restrições de execução em tempo real.

O termo Odometria Visual tornou-se popular na área de navegação robótica por compor uma etapa importante de aplicações de aulo-localização visual e *Visual Simultaneous Localization and Mapping* ou *VSLAM*, [26], comum em aplicações de robôs móveis autônomos.

Em vista dos problemas apresentados pelos métodos tradicionais de odometria, a OV oferece uma opção vantajosa por ter acesso a uma grande quantidade de informações sobre o ambiente na forma de imagens. Câmeras são equipamentos mais acessíveis em comparação a *GPS* e *INS*, apresentando melhor relação entre custo e benefício na implantação de sistemas desse tipo. Além disso, a *OV* possui maior acurácia em relação a odometria de rodas conforme indicado por Scaramuzza et al.[66].

Balazadegan et al. [6] indicam que técnicas de *SFM* em alguns casos são consideradas como sinônimo de *SLAM*. Ao utilizar imagens esta técnica passar a ser denominada *VSLAM* (*Visual SLAM*), realizando o mapeamento e auto-localização de um agente em um determinado ambiente. Neste ponto, métodos de *VSLAM* possuem resultados promissores utilizando *OV* como técnica de localização ao combiná-la com técnicas de *loop closure*. Scaramuzza et al. [66], indicam que a escolha entre *VO* e *VSLAM* é uma troca entre a performance das técnicas de *VO*, que priorizam a aplicações em tempo real, e consistência global da trajetória fornecida pelas técnicas de *VSLAM*.

Em implementações tradicionais de *OV* baseadas em técnicas de extração e correspondência de características de baixo e médio nível como cantos, bordas e fluxo óptico [67] a acurácia da posição das características nas imagens e o tempo de processamento que o processo são variáveis críticas. É importante dizer que VO depende de um sistema pré-calibrado e que os erros de calibração são propagados na estimação do movimento. Além disso, também são dependentes da precisão do extrator de características, onde normalmente são usadas características de baixo nível com imprecisão em decorrência de ruídos e informações redundantes.

Em técnicas clássicas entende-se como principais desafios o custo computaci-

onal, que limita a utilização em aplicações em tempo real. Condições de iluminação e captura das imagens bem como sombras e desfoque são capazes de influenciar negativamente a estimativa de movimento [5]. [67] e [84] indicam que um bom descritor de características deve ser robusto a estas ocorrências. Padrões complexos são a melhor alternativa nestes casos para que os extratores sejam robustos a estas ocorrências. Estes desafios forçam os métodos de *OV* clássica a realizarem sucessivas etapas de filtragem das características definidas. Para realizar a correspondência e seleção das características detectadas há diversas abordagens que influenciam o custo computacional dos sistemas de *OV*.

A correspondência entre características de diferentes imagens é citada comumente como um dos pontos de maior influência computacional no pipeline de uma Odometria Visual [67]. Um método robusto de seleção como o *RANSAC*, discutido nas próximas seções, também é etapa determinante no desempenho final das estimativas de movimento.

Outro ponto que influencia negativamente o desempenho de sistemas monoculares, onde uma única câmera é utilizada, para aplicações de *OV* reside na incerteza da escala do movimento. Aqel et al. [5] indicam que os sistemas de Odometria Visual devem estimar a escala para não falharem por sua decorrência. Ambiguidades de escala não podem ser resolvidas quando não há informações sobre o movimento realizado. Nagatani et al. [61] indicam que flutuações na escala da imagem como mudanças em relação ao terreno podem impedir que boas correspondências sejam feitas e impedem que a estimativa do fator de escala seja corretamente estimada.

Tanto para métodos diretos quanto para métodos baseado em detecção de características escala é estimada para apenas duas imagens. Neste sentido, haverá para cada par de imagens um fator de escala relativo que deve ser utilizado para o cálculo da escala absoluta no trajeto realizado[66]. Como alternativa para este passos, a escala absoluta da imagem pode ser também estimada através de sensores que fornecem informação do movimento como sensores inerciais ou até mesmo estimar o tamanho de objetos detectados na cena. Entretanto, como discutido anteriormente sensores com alta precisão possuem alto custo. A detecção de objetos por si só é um campo diverso e a escolha da técnica para esta função é complexa. Estas técnicas também podem apresentar restrições quanto a mudanças de iluminação.

Métodos diretos demandam que o brilho de um pixel seja constante sobre as superfícies processadas para que seja capaz de fazer predições acuradas [4]. Outra limitação destes métodos consiste em não apresentar bom desempenho em casos onde ocorram movimentos mínimos da câmera não sejam capturados. Em alguns casos, para lidar com este problema utiliza-se câmeras de alta taxa de captura.

Sabe-se que em *Deep Learning* (*DL*), redes neurais são capazes de extrair

padrões em vários níveis de complexidade de imagens e relacioná-los de modo eficaz. Esta abordagem têm sido aplicada em diversos exemplos de sucesso em aplicações de *VC* [80]. Neste ponto, uma técnica construída utilizando *DL* seria independente de processos comuns à *OV* clássica como calibração de câmeras, detecção e correspondência de características e técnicas de seleção de amostras em imagens. Isto permite construir uma abordagem fim-a-fim que com menos etapas é capaz de desempenhar a mesma tarefa.

Também é possível destacar que Redes Neurais Convolucionais atingiram robustez quanto a variações de iluminação através do treinamento em imagens registradas em diferentes níveis de iluminação como relatado por Ghazi et al. em [54]. Assim, problemas de textura causados por inconsistência na iluminação, uma das principais influencias para baixo desempenho de descritores de características clássicos não são tratadas explicitamente.

É comum que as características aprendidas sejam diversas, ou seja, vários descritores de características são utilizados como indicado por Krizhevsky et al. [43]. Passos de correspondências de características não são necessários pois combinação de diversos padrões de menor ordem ocorre obtendo padrões complexos. Destaca-se que uma vez treinada, um rede neural pode ser aplicada com baixo esforço computacional principalmente através de *GPUs* [16].

Em [68], Scherer et al. indicam que camadas que realizem operações do tipo *max pooling*, são capazes gerar propriedades de invariância espacial quando modificam a resolução de *feature maps*. Xu et al. [80] citam que *CNNs* independem do deslocamento de um objeto sendo capazes de localizá-lo independentemente do seu posicionamento na imagem.

As camadas de *pooling* permitem que as *CNNs* sejam robustas a pequenas deformações nas imagens. É indicado ainda que Redes Convolucionais Profundas são capazes de lidar com a escala. Nesta tarefa, o desempenho é inferior à propriedade de invariância espacial de deslocamento. Algumas arquiteturas como as redes *Inception V3* [73] lidam melhor com escala ao possuírem bancos de filtros convolucionais de diferentes dimensões em uma mesma camada.

O método proposto neste trabalho, chamado *SiameseVO-Depth*, utiliza estes conceitos para construir uma arquitetura de Rede Neural Convolucional Siamesa (*SCNN*, *Siamese Convolutional Neural Network*) capaz de estimar a transformação da pose de uma câmera entre consecutivos mapas de disparidade envolvendo na estimativas informações de profundidade. Isto permitirá demonstrar que uma *SCNN* pode ser utilizada para estimar uma *OV* combinando características complexas de duas imagens de forma simultânea.

1.2 Objetivos

O principal objetivo deste trabalho é explorar as propriedades de extração e combinação de características de *DL* na solução do problema de *OV*. A construção de uma arquitetura de rede neural profunda, capaz de estimar o movimento e orientação de um agente móvel a partir de informações tridimensionais da cena, obtidas em instantes de tempo consecutivos é tema central das nossas discussões.

Este trabalho visa o desenvolvimento de um novo método que seja independente de detectores de características tradicionais e, consequentemente, menos sensível aos problemas de imprecisão e ruídos. Isto permite lidar com a incerteza inerente ao movimento de câmera de modo automatizado, sem a necessidade de lidar com ambiguidades e correspondência de características. Assim como [31, 23, 13, 4], a metodologia proposta considera o uso de informações tridimensionais, obtidas a partir de mapas de disparidade, como base de informação para a estimação da odometria.

Os resultados apresentados por Konda e Memisevic [42] comprovam a capacidade de uma rede convolucional estimar o movimento a partir de sequências de imagens, mas sem resultados que permitam a equiparação ao estado da arte de *OV*. Constante et al. [19] demonstraram essa ideia ao atingir o estado da arte utilizando uma arquitetura *CNN* como estimador de uma Odometria Visual e indicam ainda que este é um campo promissor a ser explorado.

O trabalho proposto se diferencia dos demais ao propor uma arquitetura de rede profunda de camadas siamesas, cujo treinamento é realizado de modo supervisionado usando o mapas de disparidades de instantes de tempo consecutivos.

1.2.1 Objetivos específicos

Para atingir o objetivo apresentado é necessário percorrer etapas intermediárias que envolvem os conceitos de *OV*, *DL* e escolha da forma de otimização para a rede neural proposta. A essência do problema que relaciona transformações entre *frames* em uma sequência e a ideia de coerência temporal será utilizada como meio de atingir este objetivo.

Os seguintes tópicos serão investigados para a construção de uma técnica capaz de resolver problemas de *OV* utilizando *DL*:

1. Definição da função de custo para treinamento;
2. Estudo e definição de uma arquitetura de rede neural convolucional siamesa que usa mapas de disparidade;
3. Construção da base de treinamento ao utilizar mapas de disparidade para extrair informação tridimensional de pares de imagens;

4. Treinamento da rede aplicando os conceitos propostos;
5. Comparação dos resultados com resultados de outras técnicas que utilizem o *KITTI Vision Benchmark Suite*, explicado na Seção 4.3;
6. Implementação e documentação dos experimentos realizados.

Considerando que a rede *SCNN* proposta é um exemplo ainda não testado, a configuração de camadas e operações para convergência é feita experimentalmente para atingirmos a arquitetura que apresente melhor desempenho. Com estas ações será possível descrever em uma Seção mais detalhada os tipos de camadas utilizadas e os parâmetros de treinamento.

Uma possível limitação deste trabalho deverá ocorrer para movimentos em dimensões não consideradas neste trabalho. Isto corre pelo fato de a arquitetura treinada se ajustar a apenas um único eixo de rotação. Portanto, realizar estimativas de movimentos em outros eixos demandará uma base de dados mais completa. No entanto, compreende-se que para a validação da abordagem proposta, a base escolhida é suficiente.

De acordo com nosso conhecimento, esta é a primeira pesquisa em *OV* que relaciona a redes *SCNN* a informações de profundidade por mapas de disparidade. Será abordada também, a comparação do desempenho de *SCNN* com arquiteturas não-siamesas.

O resultado do trabalho irá compor um módulo de auto-localização e mapeamento de robôs móveis em desenvolvimento no Laboratório de Visão Computacional (PixelLab) do Instituto de Informática - UFG.

1.3 Organização deste trabalho

A organização deste trabalho pode ser descrita da seguinte forma: O Capítulo 2 abrigará discussões sobre o campo de *OV* clássica e principais limitações destas técnicas. Em sequência, as seções 2.3 e 2.4 fornecerão fundamentos de Redes Neurais Convolucionais e também redes siamesas.

Trabalhos do campo de Odometria Visual que aplicam técnicas de *Deep Learning* para estimar movimentos são discutidos no Capítulo 3 denominado Trabalhos Relacionados e a Seção 3.0.2 apresenta os trabalhos utilizados para avaliar o desempenho da proposta.

O Capítulo 4 apresentada a Metodologia Proposta definindo aspectos específicos relacionados a este trabalho apresentando detalhes sobre a base de dados, a arquitetura *SCNN* escolhida e o método de avaliação. Os experimentos realizados esta metodologia serão discutidos no Capítulo 5 com a comparação dos resultados a outras técnicas de Odometria Visual.

Por fim, na Seção de 6 encerrará este trabalho com as conclusões feitas a partir dos experimentos e também são discutidos trabalhos futuros.

CAPÍTULO 2

Fundamentos Teóricos

Neste capítulo discutiremos aspectos da Odometria Visual Clássica, discutindo as etapas envolvidas fazendo referências a trabalhos relacionados. A formulação matemática do problema de *OV* clássica é apresentada. Por fim, CNNs serão discutidas e também arquiteturas siamesas.

2.1 Odometria Visual Clássica

O termo *"Visual Odometry"* foi usado pela primeira vez por Hagnelius et al. [34] sendo uma aplicação para um sistema de navegação. Em Odometria Visual, o movimento de um agente é determinado através de uma representação tridimensional analisando-se unicamente características obtidas a partir de uma sequência de imagens.

Scaramuzza et al. em [66] e [67] apresentam um estudo sobre a história e a evolução de *OV* e uma revisão sobre os fundamentos dessa área. Yousif [83] apresenta um resumo das aplicações de *OV* em robótica móvel traçando um paralelo com as técnicas de *VSLAM*.

Lemaire et al. [49] pontuam que ao utilizar a visão monocular é necessário que várias observações de uma característica seja feita ao longo do tempo para que a coordenada *3D* de um ponto seja recuperada com maior precisão. Em aplicações de visão estéreo é possível recuperar esta informação através de correspondências entre pares de imagens.

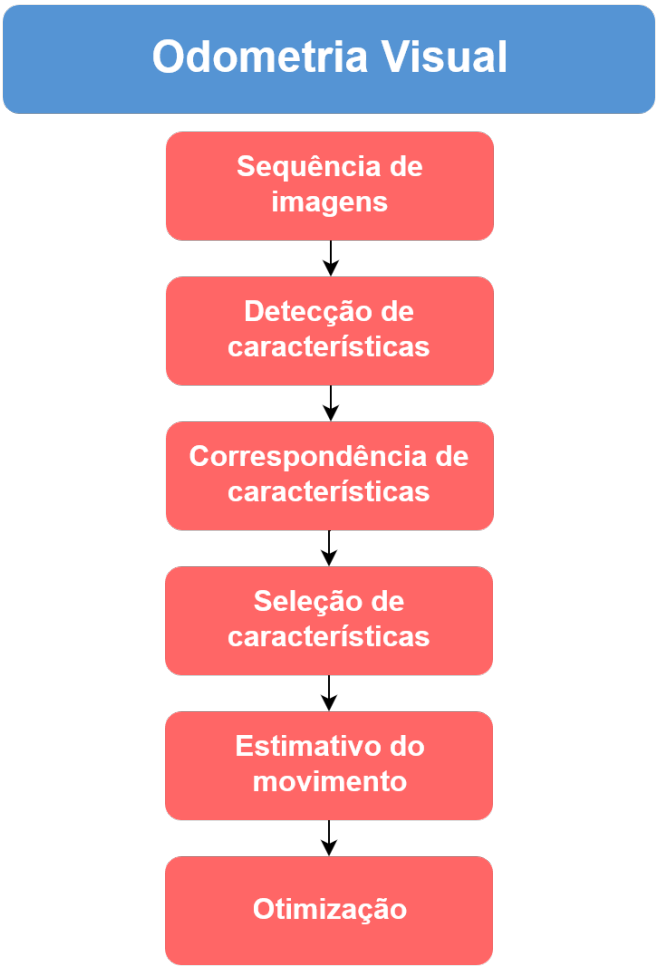


Figura 2.1: *Pipeline básico de um problema de Odometria Visual.*

Inicialmente, a captura de imagens em posições diferentes fornece a informação necessária sobre o movimento transcorrido. Em seguida, as imagens passam por um processo de detecção de características onde são encontrados pontos de interesse com estruturas de fácil rastreamento e boa repetibilidade na sequência de imagens.

Há duas principais abordagem para a detecção de características em Odometria Visual: Detectar características em uma imagem e rastreá-las em imagens consecutivas de uma sequência, ou métodos diretos baseados em intensidades de *pixels* e informações associadas [66].

Pode-se citar dentre as principais técnicas de detecção de características utilizadas na estimativa da *OV* os operadores de Moravec [22], de Harris [35], detector de cantos de Shi-Tomasi [38], FAST [65], SIFT [51] e SURF [8].

Métodos que trabalham diretamente sobre intensidades de *pixels* estão relacionados a técnicas de fluxo ótico, tarefa de avaliar posições de *pixels* em imagens de uma sequência. Através destas técnicas são registrados movimentos verticais e horizontais [60].

Métodos como SAD [7] e PLK [52] também operam com intensidades de *pixels*.

Em [55], Milella et al. apresentam um sistema de visão estéreo que avalia a correlação e encontra correspondências entre regiões de uma imagem. Em [74], um método similar à seleção de características é descrito onde descritores de regiões rastreáveis é proposto.

A Figura 2.2 exibe uma estimativa de uma *OV* antes e após da remoção de equivalências incorretas, em azul e vermelho respectivamente. Um registo mais preciso de odometria pode então ser construído em um determinado espaço discreto de tempo ao aplicar esta estratégia. Estas equivalências que não seguem um consenso global de movimento são denominadas *outliers*.

Figura 2.2: *Efeito da remoção de* outliers *na estimação da odometria. A cor vermelha representa a trajetória livre de* outliers *e a azul uma trajetória sem este refinamento.*

A estimativa do movimento é obtida iterativamente por um método linear. O principal método para estimativa de transformações entre imagens para *OV* é o algoritmo *The Gold Standard method* [36], que demanda pelo menos 8 pontos correspondentes. Porém, nesse processo as incertezas de cada estimação são propagadas para as próximas operações. Ocorrerá como consequência, a propagação de erros levando a trajetória estimada a divergir da trajetória real.

Ao obter esta estimativa, um processo de otimização não-linear que encontra os melhores valores de translação e rotação da câmera é feita através de métodos como *Bundle Adjustment* (BA) e fechamento de *loop* [6]. O *Bundle Adjustment* é um processo realiza a otimização dos parâmetros da câmera e, simultaneamente, realiza uma otimização de parâmetros tridimensionais através de marcos visuais. Esta técnica só é aplicável para situações onde correspondências entre características em mais do que duas imagens são feita [67]. .

O processo de fechamento de *loop* ocorre com a detecção de re-observância de marcos visuais ao longo de uma trajetória. Através de reconhecimento de similaridades entre novas imagens registradas e imagens anteriores um retorno a uma região previamente é detectado. Utilizando a região anteriormente mapeada maior consistência global

é atingida para reduzir a derivação das estimativas de trajetória causada pelo acúmulo de erros [83].

O escopo de aplicações de *OV* abrangem desde aplicações em veículos aéreos não-tripulados [25] a aplicações subaquáticas como descritos em [79] com alta efetividade. Maimone et al. descrevem uma das mais notórias aplicações de odometria visual nas explorações ao planeta Marte com o programa *Mars Exploration Rovers* [12, 53].

Fiala e Ufkes [23] apresentam uma aplicação de *OV* com informações de profundidades. Utilizando como sensor unicamente uma câmera *Microsoft Kinect* informações tridimensionais através de lasers infravermelhos são registradas em uma imagem bidimensional. Também são usadas imagens de espectro visível para que cada pixel seja mapeado em um espaço tridimensional ao associá-las com as imagens de profundidade. Assim, através de descritores de características *2D* os pontos detectados são mapeados em termos de coordenadas tridimensionais. Essa combinação de informações indica que utilizar informações de profundidade pode auxiliar na estimativa de uma odometria visual.

Em [4], Alismail e Browning utilizam informações de disparidade entre imagens para obter mapas de disparidade que contenham informações de profundidade semelhante às amostra de Fiala e Ufkes. Isto é feito utilizando apenas uma câmera estéreo e a ainda utiliza descritores de características próprios inspirados pelo descritor de característica proposto por Níster et al. [34]. Estes descritores são aplicados aos mapas de disparidade para estimar uma *OV* com baixos erros de translação e rotação.

2.2 Odometria Visual como problema

Supondo um agente em movimento inserido em determinado ambiente com duas câmeras fixadas a ele, a captura é realizada em um intervalo de tempo discreto k. A Figura 2.3 exibe a formulação de um problema de *OV* em um intervalo de tempo $t = 0{:}k$.

Denota-se o conjunto de imagens capturadas em cada instante de tempo como $I_{0:k} = \{I_0, ..., I_k\}$, para uma câmera monocular. As expressões $I_{e,0:k} = \{I_{e,0}, ..., I_{e,k}\}$ e $I_{e,0:k} = \{I_{e,0}, ..., I_{e,k}\}$, são usadas para sistemas de câmera estéreo, separando-as por câmera à esquerda ou à direita.

A partir desta notação, pode-se descrever a transformação $T_{k,k-1}$ como a transformação que modela o movimento ocorrido em dois instantes de tempo k e $k-1$ sendo denotada por:

$$T_{k,k-1} = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix} \tag{2-1}$$

O termo $R_{k,k-1}$ denota a matriz de rotação e o termo $t_{k,k-1}$, a matriz de translação associada ao movimento realizado pelo agente. Assim, pode-se obter a posição do agente

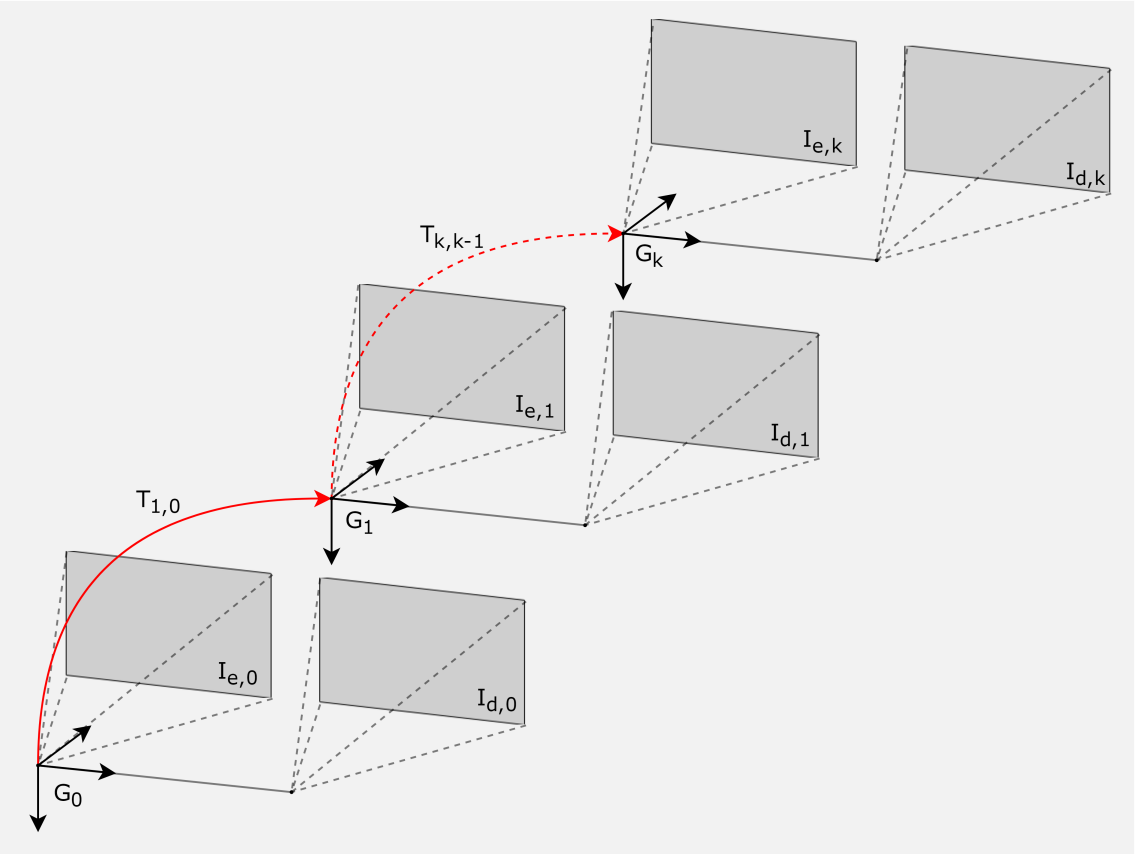


Figura 2.3: *Representação do Problema de Odometria Visual Estéreo. Uma sequência de transformações ao longo do tempo permite a construção de uma odometria.*

no espaço como uma combinação de transformações ocorrida a cada intervalo de tempo onde é realizada uma captura.

Desta forma, pode-se modelar a posição atual de um agente no espaço como a concatenação das transformações ocorridas em diferentes períodos de tempo k. Considera-se então que o conjunto de poses assumidas pela câmera em um intervalo de tempo k pode ser expressa por $G_{0:k} = \{G_0, G_1, ..., G_k\}$. Considera-se que G_0 será a pose assumida pelas coordenadas da câmera para o instante $k = 0$.

Por comodidade, a transformação $T_{k,k-1}$ (ocorrida em um intervalo de tempo k e $k-1$) passará a ser expressa simplesmente pela forma T_k. Então $G_k = \{G_{k-1}T_k\}$ será a Transformação Global, expressão que possibilita a triangulação da posição global da câmera para um determinado instante de tempo k recursivamente em relação à pose assumida em $k = 0$. O sistema de coordenadas neste caso, $k = 0$ será definido pelo usuário.

Pode-se obter descrições detalhadas sobre o funcionamento geral de um sistema de *OV* em [79], [67] e [83].

2.3 Redes Neurais Convolucionais

Redes neurais têm sido tema de estudos desde a década de 1950 influenciados por estudos como o de Von Neumann [77] que relacionavam máquinas de computação e organismos vivos. O Perceptron, proposto por Rosenblatt [64] em 1957 foi um dos primeiros algoritmos capazes de aprender a partir de dados após treinamento. Um perceptron é composto por um único neurônio que recebe entradas ponderas por um fator denominado peso sináptico, que por sua vez são somadas. A saída do neurônio que pode ser chamada de ativação é uma representação do potencial de ativação de um neurônio biológico e é obtida através de uma função não-linear chamada de função de ativação. Um único neurônio é capaz de dividir o espaço de solução em duas classes distintas através de um modelo aprendido no treinamento.

Posteriormente, esse conceito foi expandido ao utilizar-se o conceito de camadas onde múltiplos Perceptrons são conectados. Estas redes são denominadas redes Perceptron Multi-camadas (MLP, *Multi-layer Perceptron*). A saída de um neurônio será então entrada ponderada para neurônios de uma camada seguinte em processo denominado *Feedfoward*. Gardner e Dorling [27] pontuam que a sobreposição de várias unidades não-lineares possibilitam que *MLPs* sejam capazes de aproximar complexas funções não-lineares. O treinamento em redes Perceptron Multi-camadas é feito pelo algoritmo de aprendizado *Backpropagation* proposta em [48]. Este algoritmo utiliza o gradiente do erro entre a saída esperada e a saída predita para atualizar os pesos da rede neural.

Técnicas baseadas em *RNPs*, também conhecidas como *Deep Learning* (*DL*), têm apresentado novas possibilidades para a solução de problemas em reconhecimento de padrões e aprendizado de máquina. Essas técnicas têm vencido desafios científicos, com especial destaque nas áreas de *VC* e Linguagem natural [69].

. Ao expandir o número de camadas desta arquitetura ele atingiu à época o estado da arte para a classificação de imagens no *ImageNet*, uma competição que contabiliza os erros de classificação. Em comparação ao resultado do ano anterior, sua arquitetura conseguiu 37,5% de erro em relação ao segundo colocado com 45,7%. O treinamento desta arquitetura utilizou aproximadamente 1,2 milhão de imagens de alta resolução redimensionadas para uma resolução de 256×256 pixels.

Entretanto, esta não foi a primeira vez que o conceito de *CNNs* apareceu na literatura. Em 1989, LeCun et al. [45] associaram o conceito de convolução do campo de Visão Computacional a redes neurais. Uma rede convolucional toma como entrada imagens e realiza sucessivamente a extração de características de imagens através de camadas convolucionais. Ao contrário de outras aplicações de *VC*, *CNNs* não demandam que filtros convolucionais sejam previamente definidos considerando que no treinamento a rede aprenderá filtros necessários para desempenhar a tarefa de aprendizado.

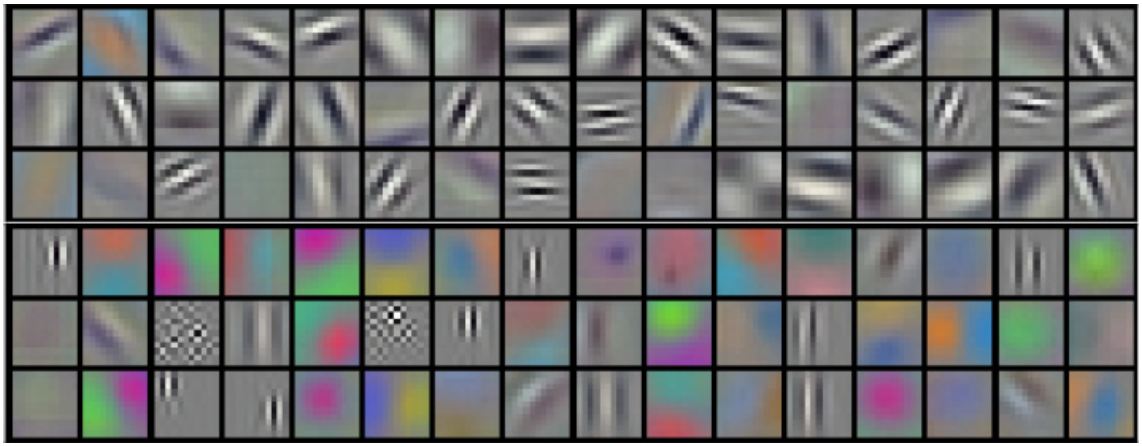


Figura 2.4: *Filtros convolucionais aprendidos pela primeira camada da rede Alexnet como descrito por Krizhevsky e Sutskever [43].*

Cada camada convolucional extrai um conjunto de características chamados *Feature Maps* que serão a resposta dos filtros da camada à entrada considerando a posição espacial. Inspirado também pelo conceito de células complexas do córtex visual, as camadas de *Pooling* realizam a combinação de padrões de menor ordem em padrões mais complexos [68]. Isto permite que uma arquitetura *CNN* seja robusta quanto a problemas de iluminação dentre outros problemas.

LeCun et al. [47] à época indicavam que cada bloco de uma rede era composto por um banco de filtros convolucionais, uma camada de *pooling* e uma camada de ativação. Atualmente, existem muitas variações para este conceito. A Figura 2.5 exibe uma típica arquitetura de uma *CNN*.

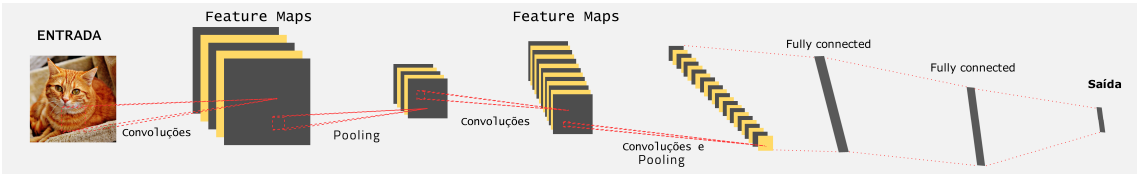


Figura 2.5: *Uma arquitetura CNN típica é composta por camadas de convolução,* pooling *e de estimação* fully connected*. Por fim, camadas de ativação são aplicadas às saídas das camadas citadas.*

Dentre as arquiteturas convolucionais mais populares podemos citar como exemplos mais famosos as arquiteturas *Inception V3* [73], *VGG16* [70] e *ResNet* [37]. Cada uma possui particularidades em suas arquiteturas que mostram que não há consenso quanto a uma arquitetura ótima para resolver os problemas de classificação e regressão em *DL*. Independente de suas particularidades, estas arquiteturas demonstraram no momento de sua publicação serem estado da arte para *CNNs*.

A etapa de otimização é passo que integra o treinamento. Como exemplo, a arquitetura Alexnet proposta por [43] foi treinada utilizando o algoritmo de otimização *Stochastic Gradient Descent* (*SGD*) [9] . No caso específico das redes convolucionais os pesos sinápticos aprendidos serão os próprios filtros convolucionais.

Atualmente existem diversos algoritmos otimizadores para o treinamento de *CNNs*. O anteriormente citado *SGD* possui popular variação com aceleração através do momento *Nesterov* [62]. Ainda existem outros como os algoritmos *RMSprop* (algoritmo sem publicação em conferências ou periódicos comumente sendo referido apenas através da apresentação de Hilton e Tieleman [75]), *Adagrad* [21], *Adam* e *Adamax* [40] utilizados para o treinamento e diferem-se com relação a particularidades de convergência. Esta otimização ocorre utilizando alguma função de energia associado ao erro medido denominada Função de Perdas.

2.4 Rede Neural Convolucional Siamesa

Redes neurais siamesas foram inicialmente propostas por Bromley et al. [10] como um novo tipo de rede neural artificial. Este tipo de rede neural é denominada siamesa pela sua arquitetura composta por suas sub-redes idênticas que têm suas saídas reunidas por uma função de energia.

Estas sub-redes utilizam a técnica de compartilhamento de pesos sinápticos treinados a partir de uma versão modificada do algoritmo de treinamento *Backpropagation* [48]. O compartilhamento neste caso é obtido ao forçar os pesos das duas sub-redes serem iguais ao longo do treinamento.

Redes neurais siamesas têm sido utilizadas em problemas em que duas amostras devem ser comparadas a partir de alguma métrica de similaridade. Koch et al. [41] pontuam que o compartilhamento de pesos permitem que duas imagens similares sejam mapeadas para posições também similares no espaço de características.

Este tipo de rede neural foi utilizada no reconhecimento de assinaturas com treinamento a partir da minimização de medidas de distância entre vetores de características extraídos. Observou-se que esta estratégia permitiu a utilização de menos parâmetros a serem otimizados e mais rápida convergência da rede neural [10].

A Figura 2.6 exibe um exemplo de uma arquitetura de rede neural siamesa para um problema onde métricas de distância são calculadas entre saídas de neurônios. Na camada de distância estes cálculos são realizados entre os valores de saída de dois neurônios da camada oculta anterior.

Esta situação é comum em problemas de processamento de vídeos em que informações sequenciais são importantes. Considerando que as informações serão analisadas

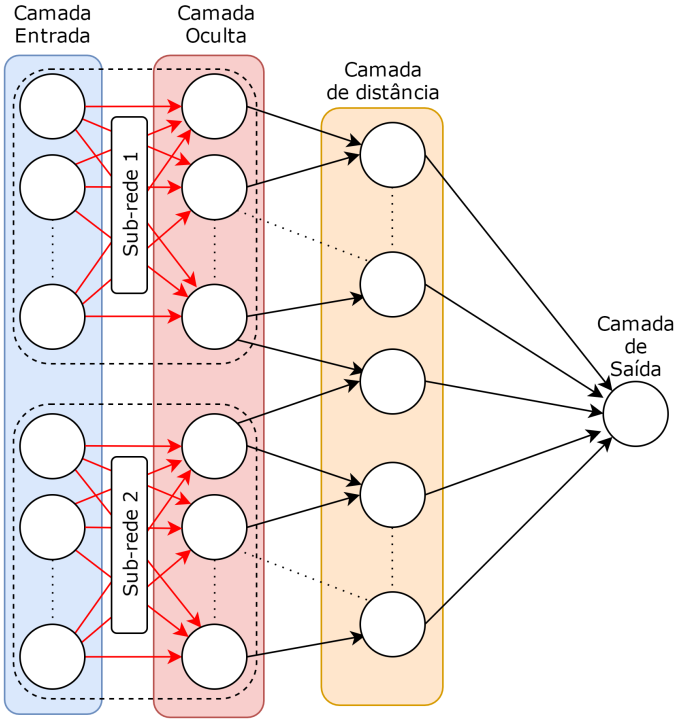


Figura 2.6: *Representação de uma rede neural siamesa para problemas que utilize métricas de distâncias para classificação ou regressão.*

de forma sequenciais a utilização de recorrência em uma rede neural convolucional siamesa pode trazer benefícios.

A Figura 2.7 exibe uma *SCNN* com diferentes tipos de camada e diferentes tipos de compartilhamento de pesos entre sub-redes. Observa-se que diferentes tipos de camadas podem ser utilizadas (camadas convolucionais, *LSTM*, *pooling*, *dense*, *dropout*, dentre outras) [43, 73, 57].

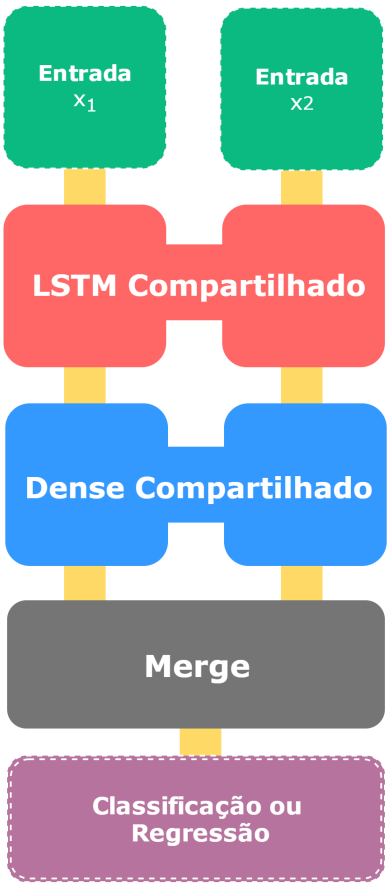


Figura 2.7: *Exemplo de uma SCNN com diferentes tipos de camadas. Diferentes tipos de camadas podem ser utilizadas permitindo a extração de características complexas pela rede neural.*

CAPÍTULO 3

Trabalhos relacionados

Neste capítulo, serão discutidos os trabalhos que buscam estimar pose de uma câmera através de técnicas de *Deep Learning* onde serão citadas as particularidades de cada abordagem. Em seguida serão descritas as técnicas de Odometria Visual utilizadas como base de comparação ao método proposto por este trabalho.

3.0.1 Deep Learning e Odometria Visual

O uso de *DL* em problemas de odometria visual é uma abordagem recente, com poucos trabalhos relacionados. Enquanto as aplicações mais comuns de *DL* têm o objetivo de classificação de padrões, onde amostras são categorizadas em um conjunto discreto e finito de classes, o problema de odometria visual requer a estimação de valores contínuos que expressam a pose de um agente. Isso exige que a rede neural funcione como um estimador da função de movimento [60].

Konda e Memisevic [42] foram uns dos primeiros autores a abordarem o problema de *OV* usando *DL*. Em 2015, os autores conseguiram prever a velocidade e direção de uma câmera, extraindo e classificando características ao utilizar redes convolucionais distintas especializadas em cada tarefa.

Agrawal et al. [3] apresentam um trabalho que avalia a utilização de informações de movimento como supervisão em problemas de aprendizado e extração de características em imagens para classificação. Os autores usaram uma versão transformada da base de dados de reconhecimento de dígitos *MNIST* [46], com diferentes operações de rotações e transação, para treinar uma rede neural capaz de estimar as transformações aplicadas. O conceito de Redes Neurais Convolucionais Siamesas [33, 76] foi utilizado na arquitetura proposta como meio de relacionar duas imagens.

Neste mesmo trabalho, o problema de Odometria Visual também foi abordado ao utilizar a base de dados extraída do *Google Street View* denominado *SF Dataset* [11] e também a base de dados KITTI [28]. O foco de Agrawal et al. não era diretamente estimar uma *OV* mas verificar a possibilidade de utilizar informações de movimento como sinal supervisório de treinamento de uma arquitetura *CNN*. Desta forma, o *ground truth* foi

transformado para conter intervalos de ângulos e translações pré-definidos e a otimização passou a ser tratado como um problema de classificação.

Ao fim de 2016, Mohanty et al. [56] apresentaram uma aplicação de *OV* em sistemas monoculares. O problema de estimar a odometria foi encarado como um problema de regressão a partir de camadas *fully-connected*. Sua arquitetura denominada *DeepVO* baseia-se em duas redes convolucionais semelhantes à *AlexNet* [43] que avaliam dois frames consecutivos que são concatenadas após etapas de extração de características utilizando funções de perda para a distância L2. Estas duas redes foram pré-treinadas na base de dados ImageNet [20]. Os resultados apresentados melhoram à medida que a base de treinamento é formada com frames que são mais próximos temporalmente, isto é, em frames consecutivos. De forma semelhante ao problema clássico de OV, os resultados tenderam a acumular erros de estimação ao longo da trajetória.

Muller [59] apresentou uma técnica de Odometria Visual monocular baseada em DL utilizando imagens de fluxo ótico extraídas pela arquitetura *Flownet* [24]. A rede convolucional, posteriormente denominada *Flowdometry* em [58], foi treinada utilizando o fluxo ótico e obteve valores de erro comparáveis ao estado da arte em *OV* ao ser aplicada à base de dados KITTI. Esta arquitetura propôs a representação por ângulo e deslocamento únicos utilizada na proposta deste trabalho considerando que deslocamentos verticais são pequenos em relação a outros deslocamentos. Foi considerado também que ângulos de rotação relacionado a movimentos verticais também são considerados desprezíveis.

Em [50], Li et al. propõem a arquitetura *UnDeepVO* capaz de estimar o movimento de uma câmera monocular. Simultaneamente à tarefa supervisionada de regressão dos valores de pose, esta *CNN* realiza também a estimativa não-supervisionada de profundidade. Sua função de perdas envolve portanto, informações de movimento e de profundidade como sinal de otimização no treinamento realizado. Nos experimentos realizados é indicado que a arquitetura *UnDeepVO* apresenta resultados competitivos em relação ao estado da arte.

De forma semelhante à técnica anterior, a arquitetura *Latent Space Visual Odometry* (*LS-VO*) realiza duas tarefas de otimização no treinamento. Porém, o trabalho de Constante e Ciarfuglia [18] realiza a estimativa da representação de espaço latende do campo de fluxo ótico conforme estudos realizados por Roberts [63] e Goodfellow et al. [30]. A tarefa de estimativa de fluxo ótico é também é realizada de forma não-supervisionada ao minimizar a função de Raiz do erro Quadrático Médio (*RMSLE*, *Root Mean Squared Log Error*). Esta técnica utiliza a mesma função de perdas utilizada na proposta do trabalho proposta por Kendal et al. [39]. A arquitetura apresentou resultados que superam o desempenho de outras *CNNs* que não utilizam múltiplas tarefas e mostrou-se mais robusta que as técnicas clássicas comparadas.

Estes trabalhos ao aplicar arquiteturas de Redes Convolucionais Profundas a

problemas de Odometria Visual, evidenciam a diversidade de estratégias possíveis.

3.0.2 Trabalhos utilizados como base de comparação

Os resultados demonstrados pela arquitetura *SiameseVO-Depth* são comparados a três técnicas que representam abordagens de estimadores de pose que uma câmera sofre:

1. A técnica *VISO2-S*, baseada em características;
2. *SVR VO*, técnica aprendizado de máquina *SVM*;
3. A técnica *P-CNN VO*, representante da classe de arquiteturas de redes *CNN* especializadas em estimar Odometria Visual.

A técnica VISO2-S é um método de *OV* estéreo capaz de realizar a estimativa a cada par de frames utilizando informações de escala para melhor desempenho. Em seu trabalho, Geiger et al. [29] realizam a estimativa de correspondências estéreo para se obter informações tridimensionais. Em seguida, um sistema de estimação de *OV* foi desenvolvido para realizar a reconstrução 3D do ambiente por onde uma câmera se move. A Odometria Visual é estimada ao minimizar uma função de erro de re-projeção de pontos de interesse e correção da trajetória com um filtro de *Kalman* que estima a velocidade da câmera. A otimização é feita através de uma função *Gauss-Newton* a ser minimizada em relação à rotação e translação. Além disso, utiliza-se de um extrator de características SAD [7] e seleção de características através do algoritmo *RANSAC*.

Em seu trabalho, Ciarfuglia et al. [15] avaliaram a aplicação de métodos baseados em aprendizado supervisionado para a estimativa de uma Odometria Visual Monocular. Utilizando as técnicas de regressão *SVM* (*Support Vector Machines*) e também modelos estatísticos da classe de Processos Gaussianos foi a primeira aplicação de algoritmos *SVM* para estimar uma *OV*. Em decorrência destes pontos, sua técnica é comumente chamada de *SVR VO*, por ser uma técnica de regressão através de algoritmos *SVM*. Experimentalmente, foi demonstrado que o desempenho da técnica *SVM* foi superior aos processos gaussianos ao utilizar o fluxo ótico em pares de imagens quantizado através da técnica Histograma de Fluxo Ótico (*HOF*, Histogram of Optical Flow).

O sistema de predição de Odometria Visual *P-CNN VO*, proposta por Constante et al. [19], realiza a estimativa frame a frame por um conjunto de Redes Convolucionais Profundas. Sua abordagem é indicada como robusta a aspectos de captura da imagem como mudanças de iluminação e de contraste e supera técnicas consideradas estado da arte em *OV*. É considerado que esta invariância é obtida através do carácter intrínseco de *CNNs* capaz de selecionar as informações contidas nos dados de treinamento necessárias para estimar o movimento de uma câmera.

Este sistema também utiliza como entrada imagens de fluxo ótico que são aplicadas a duas arquiteturas treinadas para a tarefa de estimar uma *OV*. A primeira destas

arquiteturas denominada *CNN-1b VO* utiliza toda a imagem de fluxo ótico para estimar diretamente o movimento. A arquitetura *CNN-4b VO* é a segunda arquitetura, que explora as informações locais da imagem ao dividir a imagem de fluxo-ótico em 4 sub-imagens e camadas *fully connected* realizam uma segunda estimativa do movimento sofrido pela câmera. Por fim, a arquitetura denominada *P-CNN* utiliza ambas as seções convolucionais das duas arquiteturas citadas como entrada de uma rede *fully connected* que estima a Odometria Visual com alta acurácia.

CAPÍTULO 4

Metodologia Proposta

Neste capítulo é apresentada a arquitetura *SiameseVO-Depth* treinada para estimação de poses a partir de informações extraídas de imagens estéreo bem como a forma de treinamento escolhida a partir da literatura. Também serão discutidos conceitos básicos relacionados à base de dados *KITTI* construída a partir de sensores inerciais e câmeras montadas sobre um veículo. As métricas de comparação deste trabalho com os métodos considerados como estado da arte serão apresentados ao fim desta Seção.

4.1 SiameseVO-Depth

Conforme descrito na Seção 2.4, as redes convolucionais siamesas são compostas por sub-redes idênticas base que extraem de duas entradas apresentadas as mesmas características de duas imagens. Isto permite que no treinamento a rede possa aprender a usar ambas as imagens para uma tarefa de classificação ou de regressão. As camadas de *pooling* são utilizadas para redução de dimensão e as camadas *fully connected* para estimação da função de movimento.

Existem diversas plataformas de desenvolvimento e treinamento de técnicas de *Deep Learning* disponíveis. Podemos citar os *frameworks* de desenvolvimento *Caffe*, *PyTorch*, *MXNet*, *Deeplearning4j* e *TensorFlow* como opções largamente utilizadas. Estes *frameworks* fornecem diferentes opções de linguagens de programação como *C++*, *Java*, *R* e *Python*. Os algoritmos desenvolvidos neste trabalho foram codificados e documentados na linguagem Python com utilização do *framework Keras* [14] que opera sobre o *Tensorflow* [2].

4.1.1 A arquitetura

Algumas arquiteturas utilizadas para tarefas em *DL* utilizam outras arquiteturas como extratores de características pré-treinado conectados posteriormente a camadas *fully connected* como pré-treinamento. Redes disponíveis como *VGG16* [70], *Inception V3* [73] e *ResNet* [37] têm sido utilizadas para tal finalidades com sucesso. Porém, para que

este procedimento fosse possível era necessário encontrar arquiteturas pré-treinadas para o mesmo domínio da arquitetura *SiameseVO-Depth* isto, mapas de disparidade.

Não foram encontrados modelos pré-treinados que utilizem imagens de disparidades. Nestes casos em que os domínio do problema é desconhecido é comum treinar estas arquiteturas utilizando apenas os dados novos. Como a arquitetura neste caso é siamesa, a ideia inicial foi usar uma versão inspirada na arquitetura *VGG16* para a rede base (conjunto de sub-redes que compõe a porção siamesa). Por fim, chegou-se à arquitetura proposta para esta rede base descrita a seguir.

Rede Base

A quantidade de camadas foi definida empiricamente a partir de testes de diferentes profundidades de arquitetura. Observou-se que a profundidade da arquitetura têm influência direta no desempenho da tarefa final de estimação da *OV*. Arquiteturas de pouca profundidade apresentaram baixo desempenho. Com maiores profundidade o desempenho demonstrado foi superior chegando à rede base que compõe a Seção siamesa da arquitetura proposta na Figura 4.1.

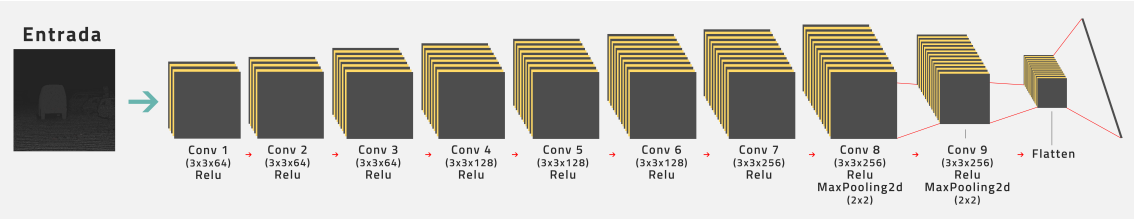


Figura 4.1: *Arquitetura base proposta para a etapa de extração de características para que camadas* fully connected *estimem a pose relativa entre* frames.

Combinando padrões de menor ordem a arquitetura é capaz de extrair características comparáveis para a tarefa de *OV* [68]. As duas últimas camadas convolucionais foram seguidas de camadas de *Pooling* do tipo *Max-Pooling* bidimensional de forma a extrair padrões robustos a distorções causadas por translações, destacando padrões mais significativos nos mapas de ativação.

Após as camadas convolucionais e camadas de *pooling*, a saída da camada *Conv 9* é aplicada a uma camada do tipo *Flatten*. Este tipo de camada recebe informações bidimensionais de cada mapa de disparidade e os transforma em um vetor unidimensional. Em seguida, cada um destes vetores que representam os mapas de ativação são concatenados obtendo um vetor unidimensional que combina todas as características extraídas das imagens de entradas.

Regressão dos valores de OV

Para este projeto o resultado do cálculo da Norma *L1*, também conhecida como Função de Diferença Absoluta, entre $z_\theta^k(x_1)$ e $z_\theta^k(x_2)$ foi utilizado como camada de mesclagem dos padrões extraídos pela rede base. A escolha foi feita empiricamente apresentando desempenho superior ao de outras técnicas de mesclagem nos testes experimentais. Dadas duas imagens consecutivas a transformação da saída da rede base é feita através do cálculo da Norma *L1* pela Equação (4-1):

$$saida_{base} = \| z_\theta^k(x_1) - z_\theta^k(x_2) \|_1 \tag{4-1}$$

Para regressão, são utilizadas 3 camadas *Fully Connected* com diferentes números de neurônios. A primeira destas camadas possui 64 neurônios, e portanto o mesmo número de saídas, a segunda camada é composta por 32 neurônios e a camada de saída da arquitetura possui 2 conforme descrito na Seção 4.1.2.

Na arquitetura siamesa proposta, estão presentes camadas de ativação do tipo *ReLU* e *LeakyReLU* [17]. Camadas do tipo *LeakyReLU* são primordiais para a obtenção de uma *OV* pois essas camadas permitem valores negativos sejam estimados representando diferentes sentidos de movimento.

A função de ativação do tipo *ReLU* pode ser descrita através da seguinte relação:

$$saida = \begin{cases} x & \text{se } x > 0 \\ 0 & \text{se } x < 0 \end{cases} \tag{4-2}$$

Uma função de ativação do tipo *LeakyReLU* funciona de forma semelhante porém o parâmetro α permite a ocorrência de valores negativos conforme descrito na relação 4-3.

$$saida = \begin{cases} x & \text{se } x > 0 \\ x \cdot \alpha & \text{se } x < 0 \end{cases} \tag{4-3}$$

Pode-se obter uma função *ReLU* a partir de uma função *LeakyReLU* simplesmente aplicando o valor de α igual a zero. A Figura 4.2 exibe o resultado da aplicação a uma entrada para estes tipos de funções de ativações. Os melhores resultados obtidos neste trabalho ocorreram para α igual 0,1, valor definido experimentalmente.

O treinamento é realizado por uma etapa supervisionada de minimização da função de perdas proposta em [39]. A Equação ((4-4)) exibe a função de perdas utilizada para este treinamento:

$$\mathcal{L} = \sum_i \|\hat{\tau} - \tau\|_2^2 + \beta \left\|\hat{\theta} - \theta\right\|_2^2 \tag{4-4}$$

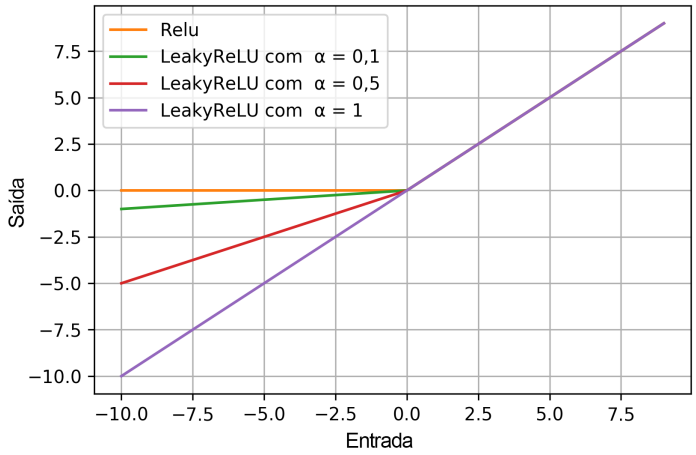


Figura 4.2: *Exemplos de diferentes valores para o parâmetro* α *escolhidos para uma função de ativação do tipo* Leaky-RelU.

onde τ e $\hat{\tau}$ são respectivamente as componentes de translação preditos e esperados, θ e $\hat{\theta}$ são os valores de rotação preditos e esperados no plano $\overline{XZ}$. β é um fator de balanceamento entre os erros angular e translacional para que os valores sejam aproximadamente iguais. O valore de β foi estipulado conforme sugerido em [39], realizando uma busca para os melhores valores entre 150 e 2000. O melhor desempenho acontece para $\beta = 950$, sendo que este foi definido experimentalmente.

A Figura 4.3 exibe a arquitetura completa da rede neural proposta por este trabalho.

Figura 4.3: *Arquitetura completa da SCNN proposta para estimação dos parâmetros da* OV. *A entrada é composta por dois mapas de disparidade no intervalo de tempo* $t = k\text{-}1{:}k$. *Operações de convolução e sub-sampling são aplicadas extraindo padrões que serão utilizados na estimativa da Rotação e Translação sofrida pela câmera. Por fim, três camadas* Fully Connected *realizam a estimativa da OV.*

4.1.2 Preparação da base de treinamento

Os dados são organizados de modo que, dada a sequência de mapas de disparidades d_k, onde k e $k+1$ são *frames* consecutivos, a base de treinamento é formada pela composição de todos os pares d_i e d_j, para $i, j \in [1, N]$, onde N é a quantidade de *frames* da sequência.

Os valores estimados estão definidos na forma de translações e rotações em um sistemas de coordenadas *3D*. As translações são denotadas por T_x, T_y e T_z relacionadas às translação nos eixos x, y e z respectivamente. De mesmo modo, θ_x, θ_y e θ_z denotam os ângulos em cada eixo.

Considerando a natureza deste problema, considera-se que a translação no eixo Y nula por ser pelo menos uma ordem de magnitude menor que o movimento nos demais eixos. Neste caso, serão desconsideradas conforme relatado por Muller [60].

As rotações estimadas são expressas como ângulos de Euler em que há três possíveis tipos de rotação: *Rolagem*, que ocorre quando o eixo z é fixo e a rotação ocorre em torno dele denotada por θ_z; *Arfagem* que é a rotação em torno de um eixo x fixo, denotada por θ_x; e *Guinada*, rotação denotada por θ_y em torno do y fixado. A Figura 4.4 exibe cada tipo de rotação possível ao utilizar ângulos de Euler sobre um veículo.

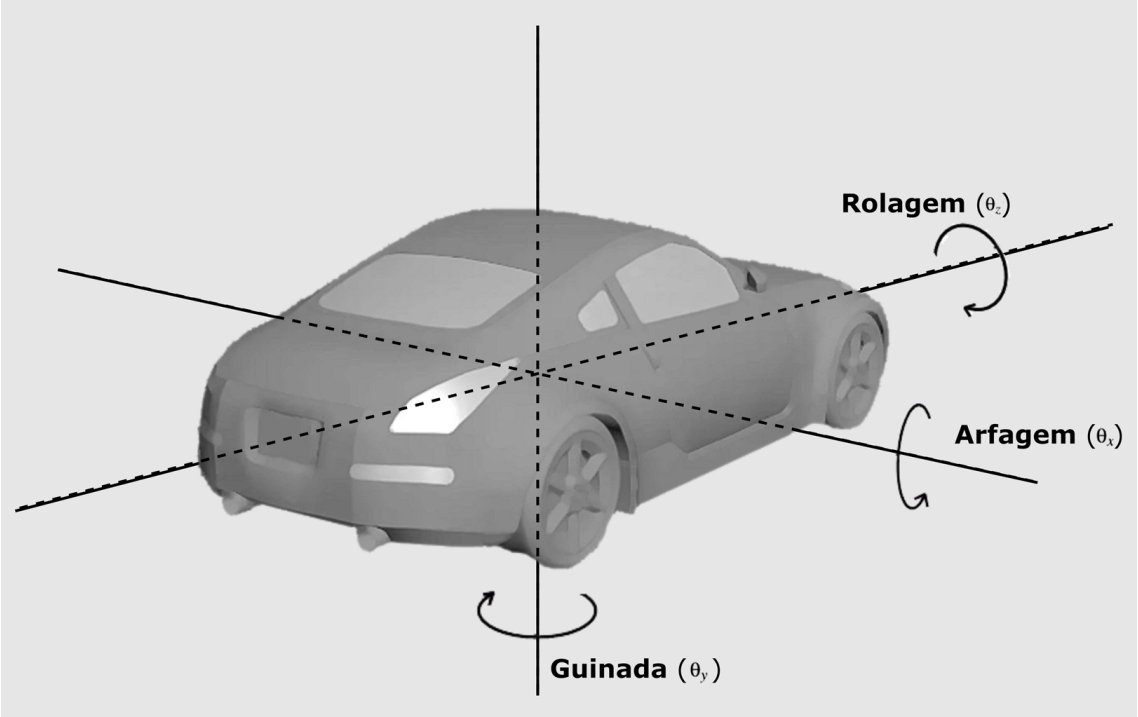


Figura 4.4: *Exemplo da disposição dos ângulos de Euler sobre um veículo. Rotações de rolagem e arfagem são consideradas desprezíveis assim como a translação T_y no eixo y.*

A exemplo das translações, considera-se que não ocorrerão consideráveis varia-

ções de rotação em determinados eixos. As rotações de *rolagem* e *arfagem* serão de menor magnitude e assim, será considerada apenas as rotações de guinada. A notação $\Delta\Theta_y$, será utilizada para as variações de rotação no eixo y.

A exemplo de [60], a matriz T_k é extraída diretamente do *ground truth* disponível na base de dados *KITTI* para *OV*. São disponibilizadas poses absolutas de cada trajetória sendo necessário primeiramente realizar a conversão para poses relativas entre cada *frame*. Considera-se que cada pose absoluta em uma sequência pode ser definida como uma combinação entre poses conforme a Equação ((4-5)):

$$P_k = T_k \cdot P_{k-1} \tag{4-5}$$

onde P_k e P_{k-1} são respectivamente, as poses absolutas consecutivas $t = k$ e $t = k-1$. Então é possível calcular a matriz de transformação T_k:

$$T_k = P_k \cdot (P_{k-1})^{-1} \tag{4-6}$$

Os valores de treinamento são calculados decompondo o vetor de translação τ e a matriz de rotação R_k, extraídos da matriz de transformação entre poses T_k seguindo a Equação ((4-7)):

$$T_k = \begin{bmatrix} \mathbf{R}_k & \tau_k \\ 0 & 1 \end{bmatrix} \tag{4-7}$$

As equações (4-8) e (4-9) exibem como a matriz de rotação R_k e o vetor τ_k são representados:

$$\mathbf{R}_k = \begin{bmatrix} R_{k,1} & R_{k,2} & R_{k,3} \\ R_{k,4} & R_{k,5} & R_{k,6} \\ R_{k,7} & R_{k,8} & R_{k,9} \end{bmatrix} \tag{4-8}$$

$$\tau_k = \begin{bmatrix} Tx_k \\ Ty_k \\ Tz_k \end{bmatrix} \tag{4-9}$$

[18] e [78] utilizam a representação de transformações da pose da câmera através de 6 graus de liberdade. A partir da matriz de rotação os ângulos de Euler são calculados pelas equações 4-10, 4-11 e 4-12:

$$\theta_x = \tan^{-1}(R_{k,8}, R_{k,9}) \tag{4-10}$$

$$\theta_y = \tan^{-1}(-R_{k,7}, \sqrt{R_{k,8}^2 + R_{k,9}^2}) \tag{4-11}$$

$$\theta_z = \tan^{-1}(R_{k,1}, R_{k,4}) \tag{4-12}$$

As translações a serem estimadas são extraídas diretamente do vetor de translação τ_i na Equação (4-9).

Outra representação possível foi proposta por [58] em que a transformação do *Ground Truth* é considerada apenas no plano $\overline{XZ}$. A rotação é estimada através da diferença entre ângulos do plano $\overline{XZ}$, $\Delta\Theta_{y,k}$ conforme a Equação ((4-13)).

$$\Delta\Theta_{y,k} = \tan^{-1}(R_{k,3}, R_{k,1}) - \tan^{-1}(R_{k-1,3}, R_{k-1,1}) \tag{4-13}$$

Para estimar a translação é utilizada a distância euclidiana entre os vetores de translação τ_k e τ_{k-1} conforme a Equação ((4-14)) e a translação Ty_k é assumida como nula:

$$\Delta d_k = \sqrt{\sum(\tau_k - \tau_{k-1})^2} \tag{4-14}$$

Experimentalmente os valores obtidos através das duas representações foram verificados e a segunda representação, proposta por Muller [58] foi utilizada durante o treinamento supervisionado por apresentar melhor desempenho.

4.2 Avaliação da abordagem proposta

Para efeito de comparação entre abordagens do estado da arte, serão utilizadas as mesmas métricas definidas no projeto *KITTI Benchmark*. O site do projeto, [1], concentra as avaliações das melhores técnicas para odometria visual. As métricas utilizadas são variações da metodologia de análise de erros de trajetória propostas por [44]. As medidas de erro são calculadas entre poses relativas, ou seja, entre poses equivalentes em instantes de tempo consecutivos.

A biblioteca de avaliação de técnicas de Odometria e também de *SLAM evo* [32], permite avaliar técnicas descritas em diversas representações do *ground truth* incluindo o padrão de trajetória da base de dados *KITTI*. Ferramentas de análise e plotagem de trajetórias são utilizadas neste projeto para gerar trajetórias e também os erros discutidos nas seções 5.2 e 5.3. A Seção 5.2 em especial, relaciona-se ao Erro Absoluto de Trajetória ou *ATE* (*Absolute Trajectory Error*) descrito em [72].

Conforme descrito [28], a métrica proposta em [44] tem a vantagem de não combinar rotação e translação em uma única medida, permitindo que sejam avaliadas de forma independente. O que permite percepções mais completas das qualidades e falhas de abordagens para cada tipo de transformação. As métricas de erro podem ser definidas como:

$$E_{rot}(F) = \frac{1}{|F|} \sum_{(i,j)\in F} \angle[(\hat{\mathbf{p}}_j \ominus \hat{\mathbf{p}}_i) \ominus (\mathbf{p}_j \ominus \mathbf{p}_i)] \tag{4-15}$$

$$E_{trans}(F) = \frac{1}{|F|} \sum_{(i,j)\in F} \left\| (\hat{\mathbf{p}}_j \ominus \hat{\mathbf{p}}_i) \ominus (\mathbf{p}_j \ominus \mathbf{p}_i) \right\|_2 \tag{4-16}$$

onde F é um conjunto de *frames* (i, j), $\mathbf{p}$ e $\hat{\mathbf{p}}$ são as poses reais e estimadas, respectivamente. Os operadores $\ominus$ e $\angle$ denotam o operador composicional inverso descrito em [71] e o ângulo de rotação.

É necessário realizar a conversão dos valores estimados para o cálculo dos erros médios de translação e rotação. Para tal, considera-se os conceitos de matriz de rotação e vetor translação. A Equação (4-17) exibe a forma da equação de rotação em 3D e a Equação ((4-18)) exibe o vetor translação dos movimentos entre um par de *frames* ambos pertencentes ao *ground truth* e as predições devem estar representados nesta forma.

$$\mathbf{R}_{pred,k} = \begin{bmatrix} R_1 & R_2 & R_3 \\ R_4 & R_5 & R_6 \\ R_7 & R_8 & R_9 \end{bmatrix} \tag{4-17}$$

$$\tau_{pred,k} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{4-18}$$

Para a conversão, da representação proposta por [58] utiliza-se a Equação (4-19), onde Δd_k e $\Delta\Theta_{y,k}$ são as saídas da *SCNN* treinada a serem transformados na mesma forma da base de dados *KITTI* e a Equação (4-20), que exibe a forma com que cada entrada é calculada. Neste caso, $\hat{\mathbf{p}}$ é obtido através da expressão 4-19.

$$\hat{\mathbf{p}}_\mathbf{k} = \begin{bmatrix} R_1 & R_2 & R_3 & T_x \\ R_4 & R_5 & R_6 & T_y \\ R_7 & R_8 & R_9 & T_z \end{bmatrix} \tag{4-19}$$

$$\hat{\mathbf{p}}_\mathbf{k} = \begin{bmatrix} cos(\Delta\Theta_{y,k} - 90^\circ) & 0 & -sen(\Delta\Theta_{y,k} - 90^\circ) & T_x \\ 0 & 1 & 0 & 0 \\ sen(\Delta\Theta_{y,k} - 90^\circ) & 0 & cos(\Delta\Theta_{y,k} - 90^\circ) & T_z \end{bmatrix} \tag{4-20}$$

Por fim, concatena-se cada linha da matriz em um vetor para comparação ao *ground truth* fornecido da base de dados. É importante destacar que estes valores representam os valores de poses relativas entre frames. É necessária a adição de uma linha na base da matriz $\hat{\mathbf{p}}_\mathbf{k}$ para que esta esteja na representação homogênea. Para de fato realizar a comparação com o *Ground Truth* é necessário portanto obter-se a *OV* em termos pose absoluta $\hat{\mathbf{P}}_\mathbf{t}$. Para isto basta realizar a multiplicação sucessiva das transformações

estimadas:

$$\hat{P} = p_{k-1} \cdot p_k \tag{4-21}$$

onde P_0 é uma matriz identidade de dimensão 4×4.

$$\hat{\mathbf{P}}_\mathbf{t} = [R_1\ R_2\ R_3\ T_x\ R_4\ R_5\ R_6\ T_y\ R_7\ R_8\ R_9\ T_z] \tag{4-22}$$

A aplicação do operador $\ominus$ é definido como a relação espacial inversa entre diferentes poses em aplicações de robótica conforme em [71] a partir das rotações sofridas entre frames pela câmera conforme a Equação (4-23):

$$F_k \ominus F_{k-1} = \begin{bmatrix} T_x' \\ T_y' \\ T_z' \\ \varphi' \\ \theta' \\ \psi' \end{bmatrix} = \begin{bmatrix} -(R_1T_x + R_2T_y + R_3T_z) \\ -(R_4T_x + R_5T_y + R_6T_z) \\ -(R_7T_x + R_8T_y + R_9T_z) \\ -\varphi \\ -\theta \\ -\psi \end{bmatrix} \tag{4-23}$$

Uma vez encontrada a relação entre transformação da poses estimadas entre diferentes *frames*, o mesmo operador $\ominus$ é aplicado entre as poses que compõem o *ground truth* para os mesmos *frames* avaliados. Desse modo, o erro calculado é a relação espacial inversa entre a transformação estimada e as poses reais.

A partir desta expressão obtêm-se os erros de rotação $E_{rot}(F)$ e de translação $E_{trans}(F)$ descritos nas equações (4-15) e (4-16).

4.3 Base de dados

Como citado anteriormente, em *DL* redes neurais profundas são capazes de realizar tarefas de predição complexas ao utilizar grandes quantidades de dados. A base de dados disponibilizada pelo projeto *KITTI Vision Benchmark Suite* [28] conta com várias sequências de imagens produzidas para a avaliação de técnicas em Visão Estéreo, Odometria, Fluxo Óptico e Reconhecimento de objetos.

A base de dados para *OV* e *VSLAM* é composta por 23 sequências de imagens capturadas de forma continua a cada 0,1 segundos capturadas com resolução de 0.5 megapixels. Estas sequências corresponde a um trajeto total de 39.2 Km contabilizando um total de 41 mil *frames* capturados em ambiente externo com *ground truth* associado a toda a sequência. Destas, 11 sequências possuem anotações da pose da câmera necessária para o nosso treinamento e as outras 11 sequências são utilizadas para geração dos resultados em seu *benchmark*.

A proposta dessa base de dados foi motivada pela escassez de *benchmark* que permitam concentrar a comparação de técnicas relacionadas às áreas citadas. Portanto, é um esforço no desenvolvimento de veículos autônomos oferecendo imagens de trajetórias de veículos e um *ground truth* construído a partir de sensores inerciais.

A aquisição de imagens foi feita por quatro câmeras, formando um sistema estéreo monocromático e outro colorido. As câmeras são do modelo *PointGrey Flea2*, possuem resolução de 1392×512 *pixels*, aberturas de $90^\circ \times 35^\circ$ e taxa de captura de 10 *frames* por segundo. A Figura 4.5 exibe um exemplo de imagem capturada por estas câmeras. Para a leitura de dados tridimensional com *lasers*, possível abordagem de *OV* foi utilizado o sensor *Velodyne HDL-64E 3D* que também registrava imagens a 10Hz com alcance de 100 metros.

Figura 4.5: *Exemplo de imagem pertencente à base de dados KITTI.*

Os dados do *ground truth* para o problema de Odometria Visual foram registrados com uma unidade de localização *GPS/IMU* com correção de sinais *RTK* com erros de localização a céu aberto inferiores a 5cm. As câmeras e demais sensores foram montadas sobre um veículo sendo que as câmeras que compõe o sistema estéreo são de um mesmo tipo, ou seja, cor ou em escala de cinza foram posicionadas com distância de 54 cm. Neste trabalho são usadas as imagens em escala de cinza.

A base KITTI conta ainda com mapas de disparidade para 389 pares de imagens. Para compor a base de disparidades de treinamento, os autores do projeto [28] explicam que os mapas podem ser gerados usando o algoritmo *PCBP*, descrito por Yamaguchi et al. [81]. O algoritmo *Particle Convex Belief Propagation* (*PCBP*) é comparado a 9 outras técnicas de visão estéreo apresentando o melhor desempenho registrado para esta base de dados.

Para viabilizar a abordagem proposta nesta dissertação, optou-se por utilizar a técnica descrita em [82] para estimar os mapas de disparidade por apresentar melhores resultados no *benchmark*. Esta técnica apresenta menores porcentagens de pixels de disparidade erroneamente calculados para áreas não-oclusas e também para todos os

pixels das imagens. A Figura 4.6 exibe um exemplo do mapa de disparidade obtido para o mesmo frame exibido na Figura 4.5.

Figura 4.6: *Exemplo de mapa de disparidade extraído da base de dados KITTI.*

CAPÍTULO 5

Experimentos e resultados

Este capítulo apresenta os resultados obtidos pela aplicação da metodologia *SiameseVO-Depth* na base *KITTI*. Através do framework *Keras*, os treinamentos e testes foram realizados em um computador desktop equipado com uma *GPU NVIDIA GTX 1060* com 6GB de *VRAM* e processador *Intel Core i5* 3,5GHz.

É comum que em aplicações em utiliza-se o dataset *KITTI* para Odometria Visual, sejam utilizados as sequências 00 a 07 para treinamento com um total de 16330 imagens. Estas sequências de imagens são capturadas de forma continua a cada 0,1 segundos. Para o conjunto de testes, utiliza-se as sequências 08 a 10 reunindo 6860 imagens, o que corresponde a um particionamento dos dados de aproximadamente 70% para treinamento e 30% para testes.

5.1 Os erros de translação e rotação

O *KITTI Vision Benchmark Suite* fornece o *Odometry Development Kit*, que registra as métricas de desempenho de uma técnica para um conjunto de dados. Esta ferramenta permite verificar os erros de rotação, E_{rot} e de translação, E_{trans}, de forma simplificada. Estes erros são calculados através das equações 4-17 e 4-18 definidas anteriormente na Seção 4.2.

Utilizando esta ferramenta, cada sequência de 08 a 10 é avaliada individualmente e os erros são registrados. Em seguida, estas sequências são avaliadas em conjunto para registro dos erros médios para todo o conjunto de testes. A comparação dos resultados será feita com as técnicas *VISO2-S* [29], *SVR VO* [15] e *P-CNN VO* [19] previamente descritas na Seção 3.0.2. Os valores de E_{rot} e E_{trans} são apresentados na Tabela 5.1.

	VISO2-S		SVR VO		P-CNN VO		SiameseVO-Depth	
Sequência	Translação [%]	Rotação [$\circ/m$]	Translação [%]	Rotação [$\circ/m$]	Translação [%]	Rotação [$\circ/m$]	Translação [%]	Rotação [$\circ/m$]
08	10.75	0.0303	14.44	0.0300	**7.60**	**0.0187**	28.81	0.09361
09	9.76	0.0271	8.70	0.0266	**6.75**	**0.0252**	17.57	0.05513
10	**10.30**	**0.0243**	18.81	0.0265	21.23	0.0405	39.01	0.17497
Média	10.27	0.0273	13.81	0.0302	**8.96**	**0.0235**	28.46	0.10791

Tabela 5.1: *Erros médios de translação e rotação medidos para as sequências 08 a 10.*

Pela tabela 5.1 a proposta *SiameseVO-Depth* apresenta erros de rotação maiores do que as outras técnicas comparadas. Em média, os erros de rotação são de aproximadamente 3 a 4,5 vezes maiores do que as técnicas *VISO2-S*, *SVR VO* e *P-CNN VO*. A sequência 08 apresenta valores de erro que são aproximadamente cinco vezes maiores que o melhor valor de rotação, obtido pela técnica *P-CNN VO*. Na sequência 09, a diferença para o menor erro obtido, foi 2 vezes maior. Por fim, a sequência 10 apresenta os maiores erros de rotação. Pelos resultados, o erro de rotação obtido é cerca de 7 vezes maior que os valores de rotação que a técnica *VISO2-S* apresenta, 6,6 vezes maior que a técnica *SVR VO* e 4 vezes maior que os erros da técnica *P-CNN VO* avaliada.

Assim, o melhor desempenho foi obtido na sequência 09 e o de menor desempenho na sequência 10. A sequência 10 também apresenta maiores erros para as técnicas *VISO2-S*, *P-CNN VO* e *SiameseVO-Depth* com apenas a técnica *SVR VO* se destacando. Na sequência 09, os erros de rotação foram os menores obtidos para todas as outras técnicas, com exceção da técnica *SVR VO*, em todas as sequências.

Os erros de translação também são maiores em comparação aos erros das outras técnicas avaliadas. Na sequência 08, os valores de translação apresentam desempenho inferior às outras técnicas ficando mais próximo da técnica *SVR VO*. O erro médio de translação na sequência 09 foi inferior às demais técnicas não se aproximando dos outros valores verificados.

A técnica *SiameseVO-Depth* está mais próxima em termos de valores da técnica *SVR VO*, técnica que utiliza um algoritmo de regressão do tipo *SVM* sem utilizar técnicas de *Deep Learning*. Observando os valores de erro percentual de translação, vemos que os a diferença é maior em aproximadamente 18% em relação ao *VISO2-S*, 15% para a técnica *SVR VO*, e 20% na técnica *P-CNN VO*.

Pode-se destacar que ainda que os valores de erro médio sejam maiores que as demais técnicas o método discutido neste trabalho apresenta comportamento semelhante às demais técnicas. Observa-se que com exceção da técnica *VISO2-S*, todas as demais técnicas apresentam baixo desempenho na sequência 10. Neste ponto, é observado que as técnicas que realizam a estimativa por modelos de aprendizado de máquina apresentam baixo desempenho nesta sequência. A técnica *VISO2-S* baseada na metadologia de *VO* clássica apresentou os melhores resultados. De maneira semelhante, as técnicas *SVR VO*

e *P-CNN VO* apresentam para a sequência 08 desempenho inferior do que a sequência 09. Este comportamento também é observado na abordagem proposta para a arquitetura *SiameseVO-Depth*.

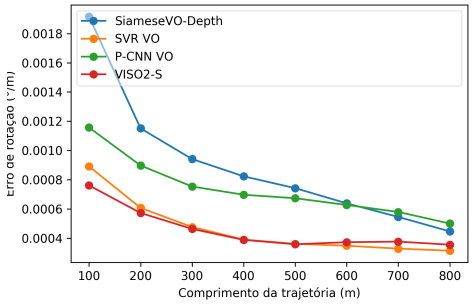
5.1.1 Relacionando E_{rot} e E_{trans} com o comprimento do trajeto e a velocidade

Ao fazer a estimativa de uma *OV* é importante destacar que a integração do erro ao longo de uma trajetória faz com que este acúmulo de erro seja cada vez maior ao fim de uma sequência. Por isso, é interessante avaliar esta técnica sobre a ótica de valores médios para diferentes comprimentos de uma sequência. A velocidade do veículo também é fator importante pois, em técnicas clássicas em que ocorre *matching* de características e mantida mesma taxa de amostragem indica-se que quanto maior a velocidade menor a quantidade de características comum em dois frames consecutivos.

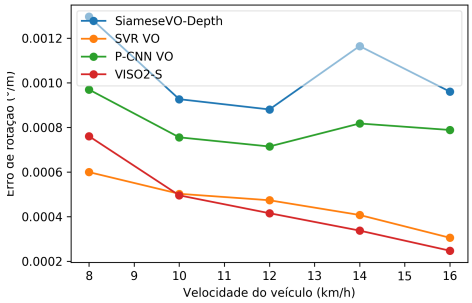
Nestes casos, os erros translacionais e rotacionais são calculados realizando a média pelo comprimento de subsequências de uma sequência. Esse comprimento varia de 100 a 800 metros e permite observar o efeito da integração dos erros ao longo de uma *OV*. Em relação à velocidade, os erros também serão avaliados pela média em relação às velocidades medidas em *km/h*.

A quantidade de amostras onde ocorrem maiores rotações, como em curvas é significativamente menor do que a ocorrência de situações de menores rotações e o veículo está em trajetória mais retilínea. Neste caso, quanto menor a subsequência os erros de rotação nas regiões de maior rotação tendem a serem maiores. Consequentemente, o treinamento da arquitetura será melhor em estimar pequenas rotações do que grandes rotações. A seguir as sequências serão avaliadas de duas maneiras. A primeira será verificar o desempenho de forma individual e também de forma conjunta para verificar o desempenho global do nosso método.

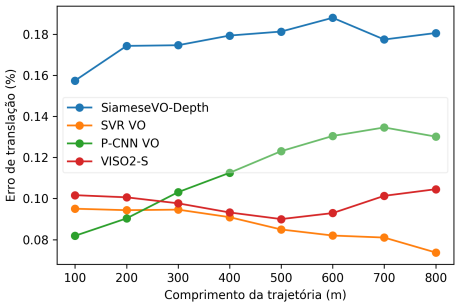
Para a sequência 08, os valores médio de erros rotacionais e translacionais em relação ao comprimento de subsequências e também em relação à velocidade do veículo são exibidas na Figura 5.1.



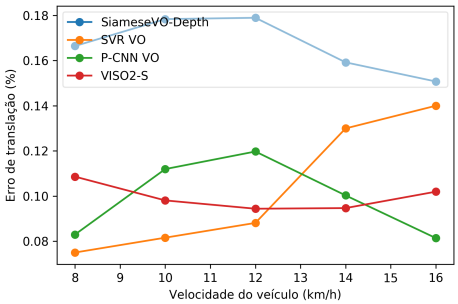
(a) Erro de rotação em relação ao comprimento da trajetória.



(b) Erro de rotação em relação à velocidade do veículo.



(c) Erro de translação em relação ao comprimento da trajetória.

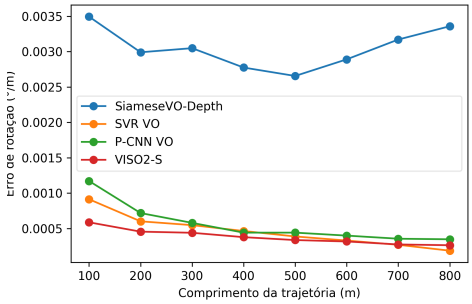


(d) Erro de translação em relação à velocidade do veículo.

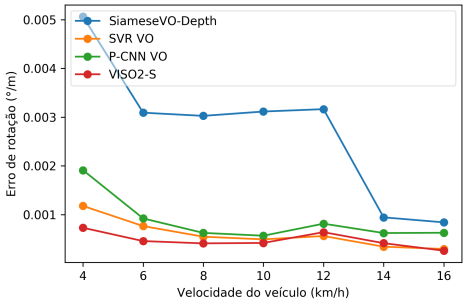
Figura 5.1: *Comparação das técnicas* SiameseVO-Depth, SVR VO, P-CNN VO *e* VISO2-S *para a sequência 08 em relação à velocidade do veículo e comprimento da trajetória.*

A Figura 5.2 exibe os erros de rotação e translação e rotação em relação ao comprimento da trajetória e à velocidade do veículo para a sequência 09. Na figura 5.2 (a) e (b), a rotação têm comportamento novamente semelhante à técnica *P-CNN VO* ainda que na rotação seja inferior às outras técnicas. Ainda assim, os valores médios são inferiores aos dos erros de rotação medidos para as sequências 08 e 10. Isto novamente, indica que a sequência 09 possui menor dificuldade de estimativa do que as outras sequências.

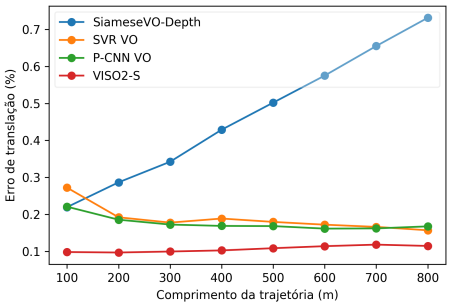
Em seguida, é apresentada a compilação dos resultados para todas sequências avaliadas na Figura 5.4. Os valores de erros foram avaliados unindo todas as sequências permitindo uma visão geral da *OV* das sequências 08, 09 e 10.



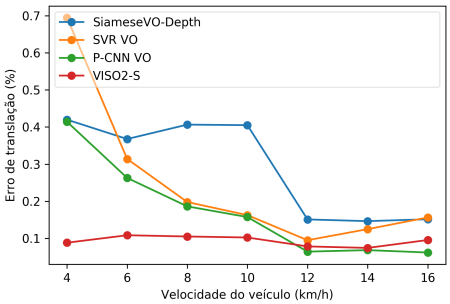
(a) Erro de rotação em relação ao comprimento da trajetória.



(b) Erro de rotação em relação à velocidade do veículo.

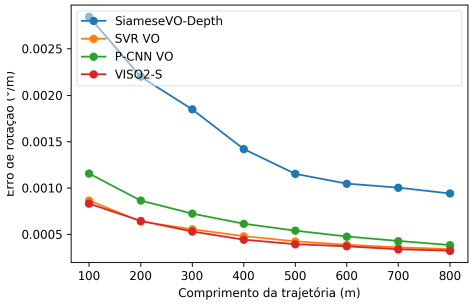


(c) Erro de translação em relação ao comprimento da trajetória.

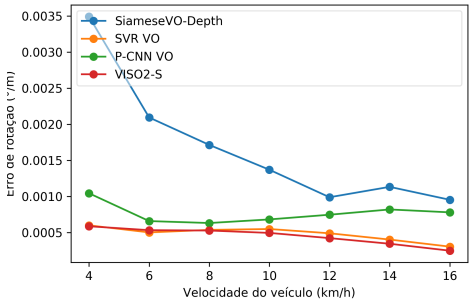


(d) Erro de translação em relação à velocidade do veículo.

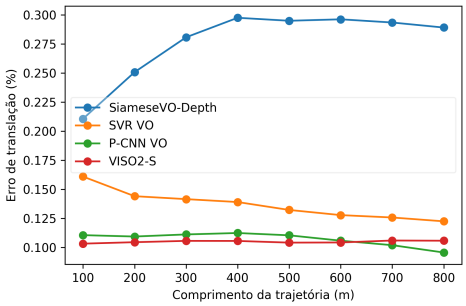
Figura 5.2: *Comparação das técnicas* SiameseVO-Depth, SVR VO, P-CNN VO *e* VISO2-S *para a sequência 09 em relação à velocidade do veículo e ao comprimento da trajetória.*



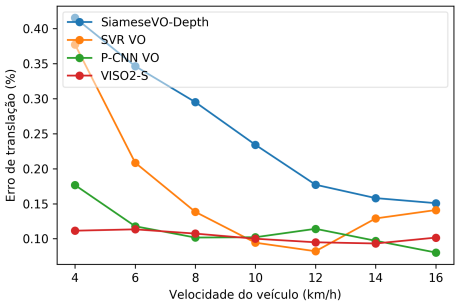
(a) Erro de rotação em relação ao comprimento da trajetória.



(b) Erro de rotação em relação à velocidade do veículo.



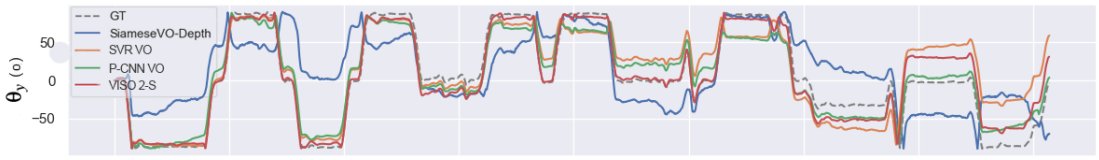
(c) Erro de translação em relação ao comprimento da trajetória.



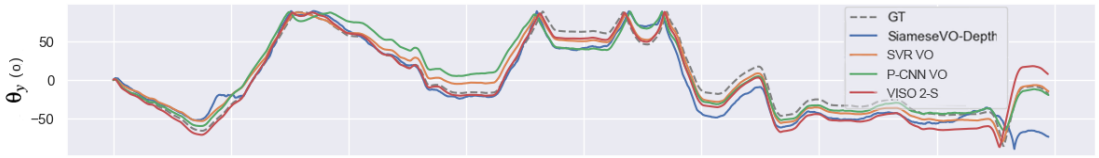
(d) Erro de translação em relação à velocidade do veículo..

Figura 5.3: *Comparação das técnicas* SiameseVO-Depth, SVR VO, P-CNN VO *e* VISO2-S *para a sequência 10 em relação à velocidade do veículo e ao comprimento da trajetória.*

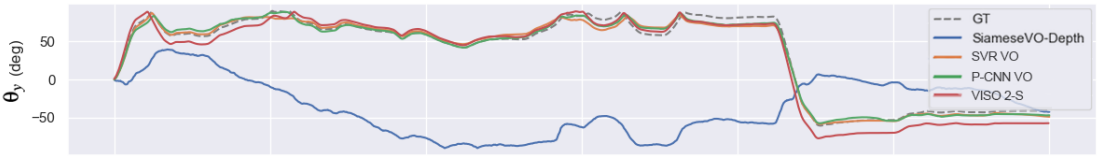
5.2 Erro Absoluto de Trajetória



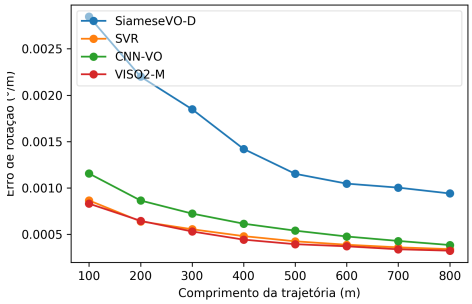
(a) Sequência 08.



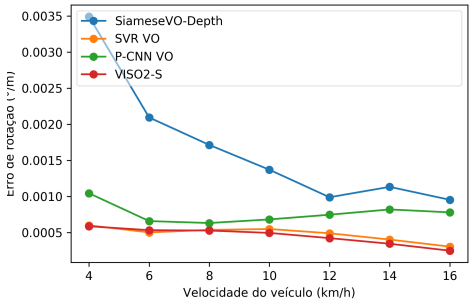
(b) Sequência 09.



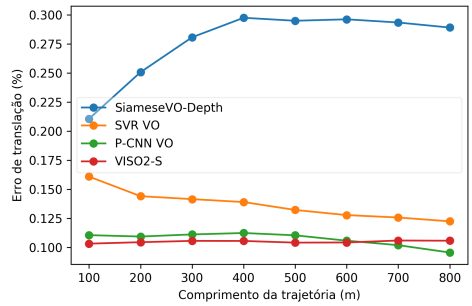
(c) Sequência 10.



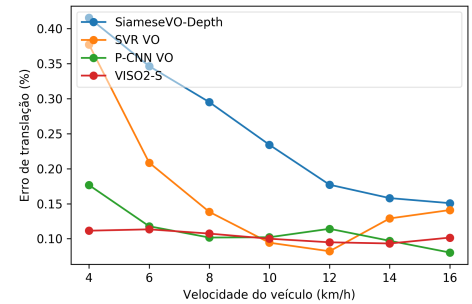
(a) Erro de rotação em relação ao comprimento da trajetória.



(b) Erro de rotação em relação à velocidade do veículo.



(c) Erro de translação em relação ao comprimento da trajetória.



(d) Erro de translação em relação à velocidade do veículo..

Figura 5.4: *Comparação das técnicas* SiameseVO-Depth, SVR VO, P-CNN VO *e* VISO2-S *para as sequências 08 a 10 considerando o comprimento da trajetória e a velocidade do veículo.*

5.3 Avaliando as trajetórias

O pacote *evo* permite também traçar as trajetórias das técnicas comparadas. A Figura 5.6 exibe as trajetórias estimadas plotadas para cada técnica avaliada. O método proposto exibe semelhança ao traçado real, porém o acúmulo de erros translacionais faz com que a trajetória seja distorcida.

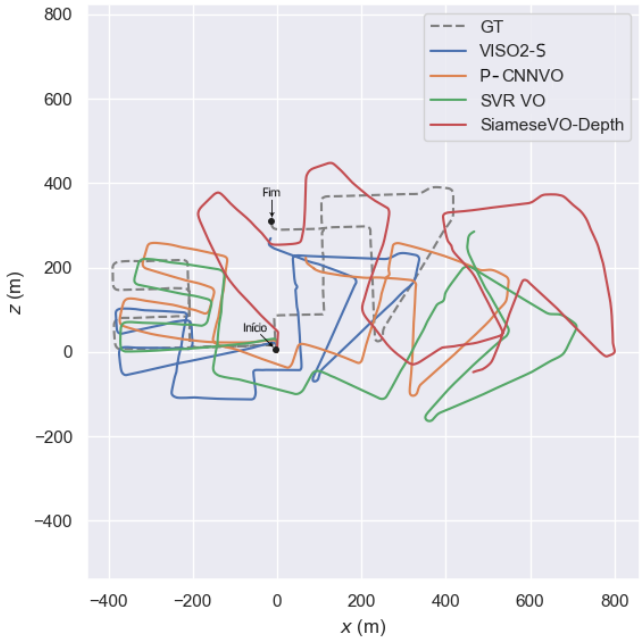


Figura 5.6: *Comparação da trajetória estimada pela arquitetura* SiameseVO-Depth *com as demais técnicas avaliadas para a sequência 08.*

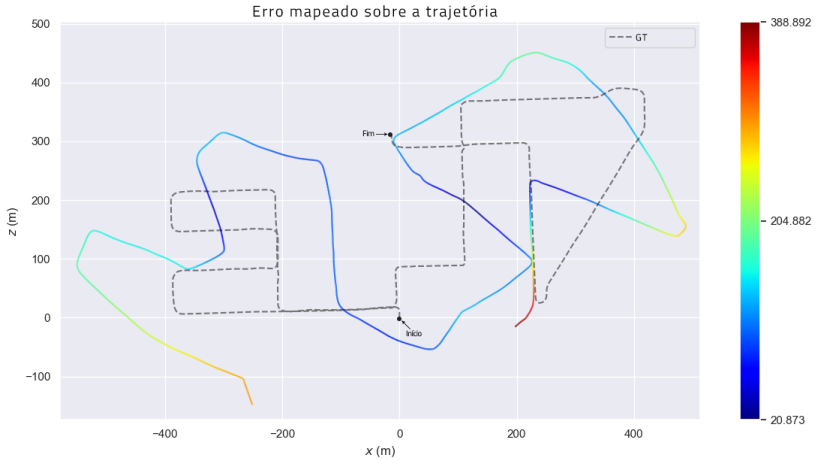


Figura 5.7: *Erro ATE mapeado sobre cada ponto da sequência 08.*

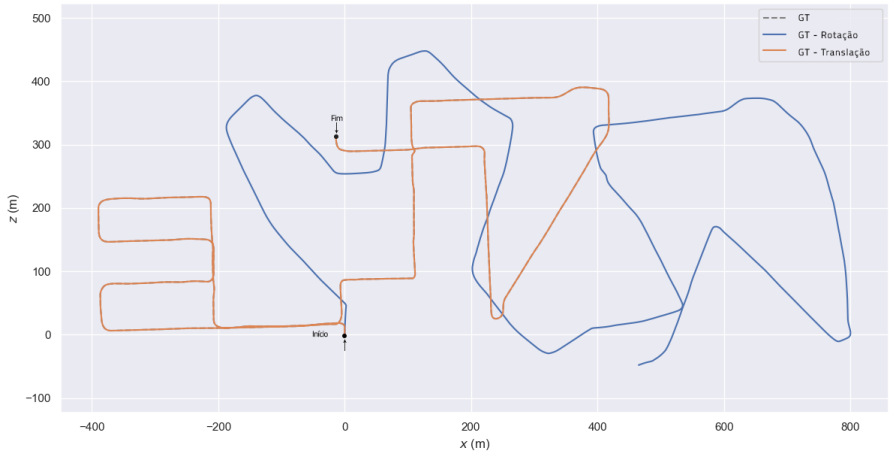


Figura 5.8: *Influência dos erros de rotação e translação na distorção da trajetória predita para a sequência 08.*

Observa-se que ao utilizar a rotação estimada pela arquitetura *SiameseVO-Depth* e a translação do *ground truth* τ_k, a trajetória coincide com a trajetória real esperada. Ao utilizarmos a rotação $\mathbf{R}_k$ e a translação predita $\tau_{pred,k}$ observa-se que a trajetória do veículo localiza-se bem próxima da trajetória predita pela arquitetura *SiameseVO-Depth*. Isto indica que o método proposto é capaz de realizar boa estimativa da rotação sofrida pela câmera. Consequentemente, a translação possui maior influência na distorção sofrida exibida nas Figuras 5.6 e 5.7.

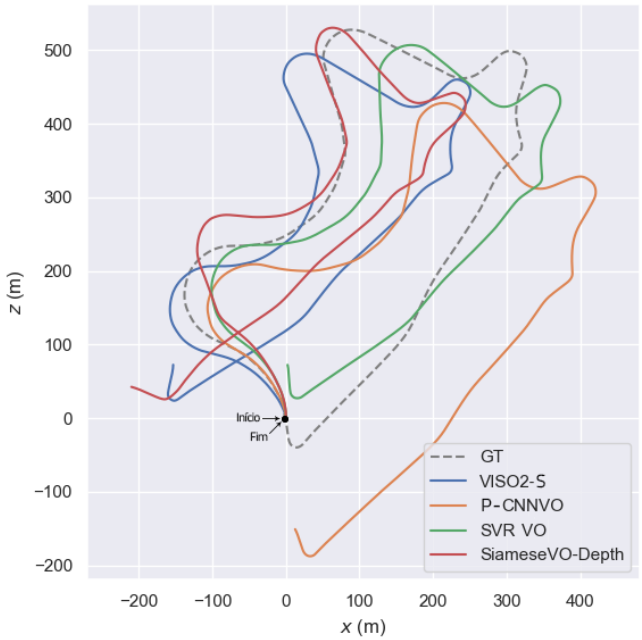


Figura 5.9: *Comparação da trajetória estimada pela arquitetura* SiameseVO-Depth *com as demais técnicas avaliadas para a sequência 09.*

A Figura 5.10 mostra que os erros de translação foram maiores no início da trajetória e em trechos em que o veículo manteve trajetória mais curvilineia. Observa-se que o resultado para arquitetura proposta apresenta melhor desempenho em regiões de maior rotação e translação em comparação à trajetória 08, que é mais retilínea.

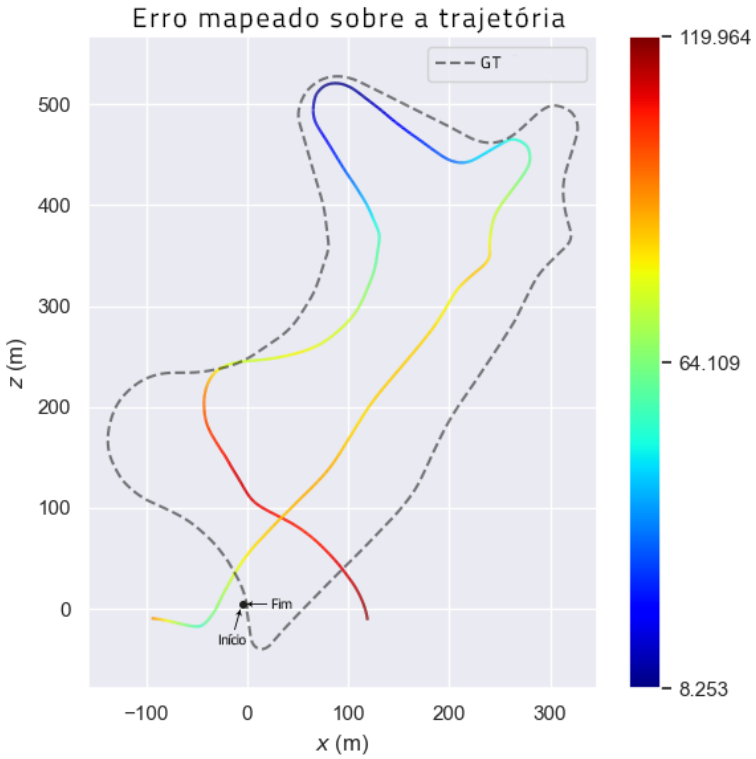


Figura 5.10: *Erro ATE mapeado sobre cada ponto da sequência 09.*

Ao observar as distorções sofridas pela trajetória na Figura 5.11 considerando individualmente as rotações e translações preditas, observamos novamente que a translação estimada possui maior influência na distorção do que a rotação. A trajetória construída utilizando a translação τ_k também está localizada bem próxima ao *ground truth*.

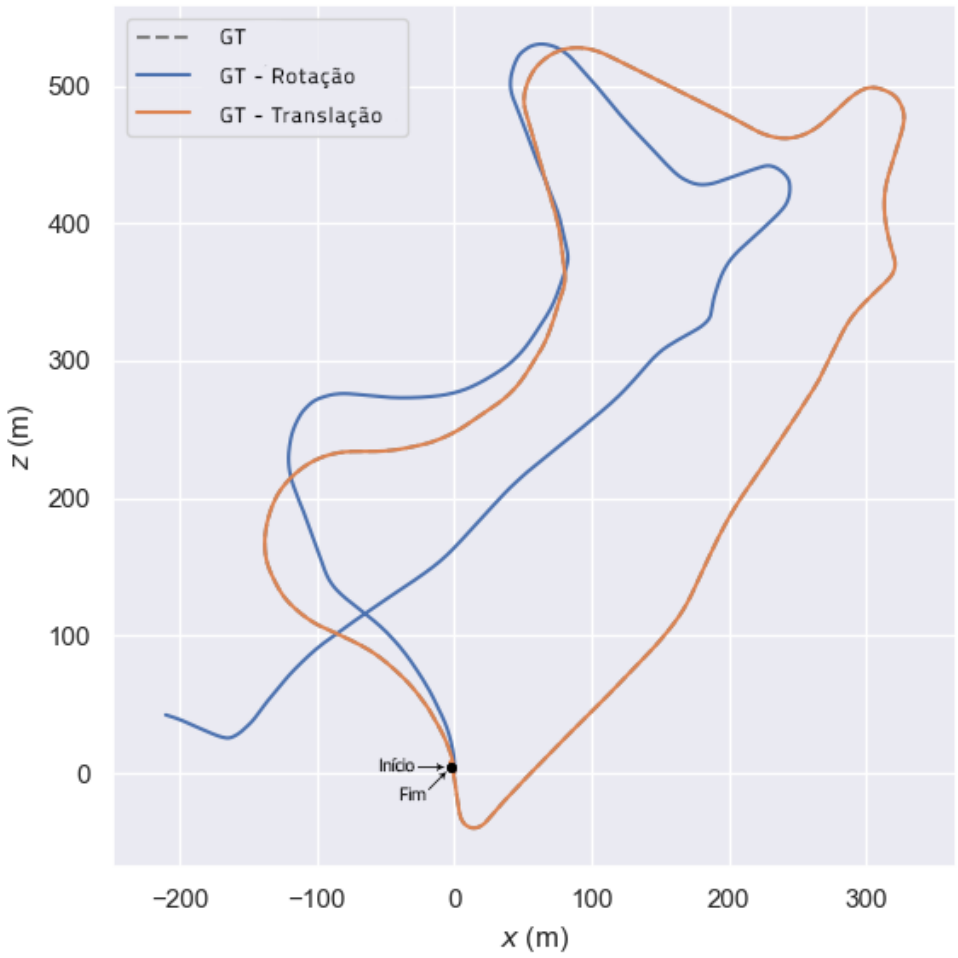


Figura 5.11: *Influência dos erros de rotação e translação na distorção da trajetória predita para a sequência 09.*

Na última das sequências, tendo em vista os altos valores de erro esperava-se que a trajetória fosse bem distorcida. A Figura 5.12 mostra o método *SiameseVO-Depth* em comparação às outras técnicas.

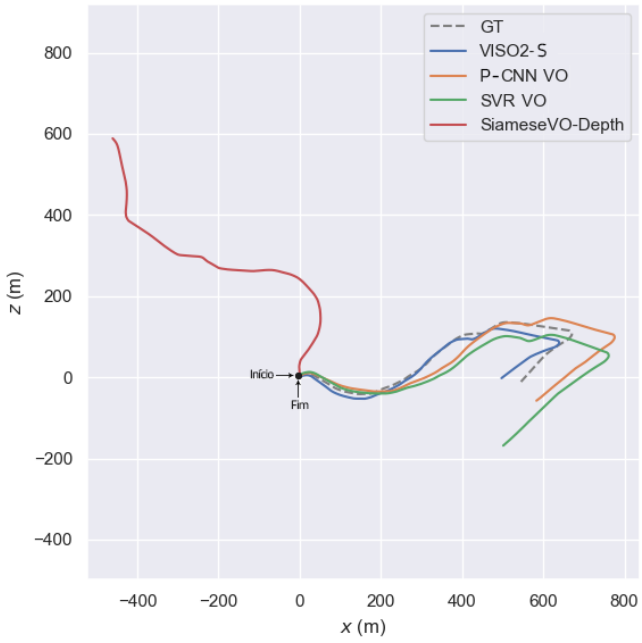


Figura 5.12: *Comparação da trajetória estimada pela arquitetura* SiameseVO-Depth *com as demais técnicas avaliadas para a sequência 10.*

Por fim, a Figura 5.13 exibe as trajetórias projetadas pela translação τ_k e a rotação $\mathbf{R}_k$ do *ground truth* em combinação com as predições $\mathbf{R}_{pred,k}$ e $\tau_{pred,k}$.

Ainda que os valores de erro rotacionais sejam altos, a projeção da trajetória utilizando a translação real do *ground truth* em conjunto com a rotação sobrepôs-se novamente à trajetória esperada. Portanto, mesmo que na trajetória 10 o desempenho seja inferior ao das trajetórias 08 e 09 considera-se que esta ainda seja uma boa estimativa em relação à rotação.

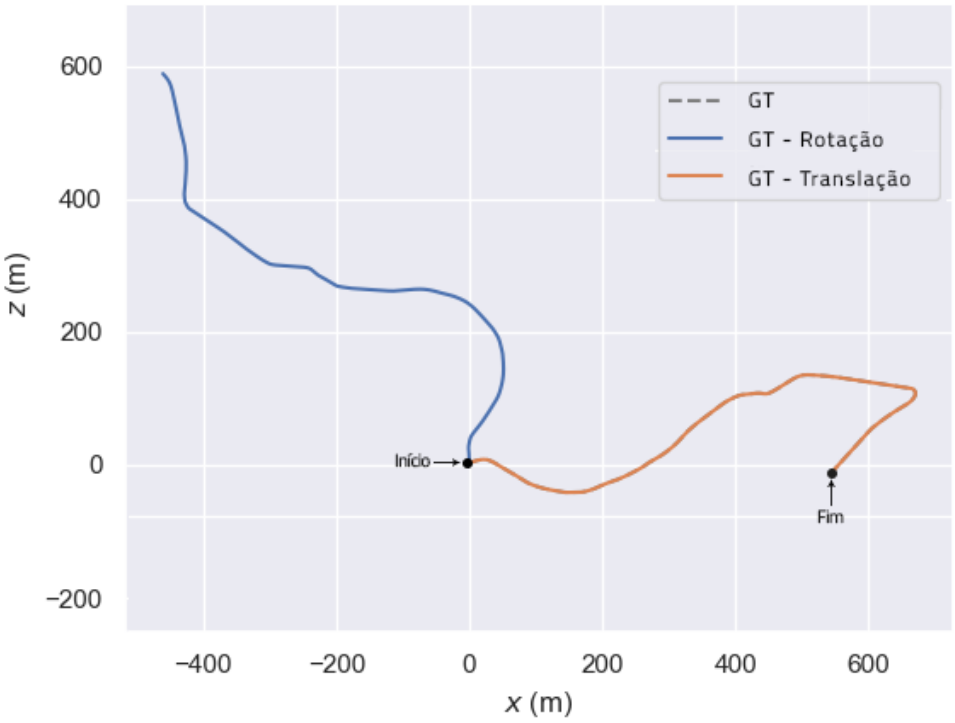


Figura 5.13: *Influência dos erros de rotação e translação na distorção da trajetória predita para a sequência 10.*

CAPÍTULO 6

Conclusões e Trabalhos Futuros

6.1 Conclusões

Ainda que a acurácia não supere as técnicas no estado da arte a comparação foi possível. Os resultados apresentados mostram que o modelo *SiameseVO-Depth* consegue fazer estimativas consistentes da componente de rotação da pose e que a estimativa da translação possui limitações quando o movimento realizado é pequeno.

Discutimos como o problema de Odometria Visual pode ser explorado através da técnica de aprendizado profundo *SiameseVO-Depth*. Outras técnicas de *OV* foram utilizadas como base de comparação em experimentos realizados para verificar o desempenho desta proposta. Foram explorados aspectos da otimização e foi observado que a função de custo utilizada foi capaz de conduzir o treinamento ainda que modificando os dados extraídos do *ground truth* para um padrão que envolve apenas movimentos realizados no único plano considerado, o plano $\overline{XZ}$.

A arquitetura proposta contribui com o estudo desta área do conhecimento ao mostrar-se capaz de aprender padrões complexos por uma abordagem fim-a-fim, suficientes para que uma odometria seja construída através de mapas de disparidade. Como sugerido na literatura, isto indica que informações de profundidade contribuem na estimava das transformações sofridas pela pose da câmera.

Levando-se em consideração esses aspectos, a aplicação de uma arquitetura siamesa demonstrou capacidade de extrair padrões equivalentes de imagens distintas e partindo destas informações, realizar estimativas. Esta é uma das principais contribuições deste trabalho considerando que foram encontradas poucas técnicas exploram estas propriedades. A combinação de uma base de dados constituída por imagens de mapas de disparidade com este tipo de arquitetura é no presente momento inédita na literatura sendo outra importante contribuição desta pesquisa.

6.2 Trabalhos Futuros

Considerando os resultados expostos, pode-se verificar que a técnica proposta possui limitações para regiões de maiores rotações e menores translações pois estas situações ocorrem em uma proporção menor. Para contornar esta limitação, no futuro deseja-se aumentar a quantidade de amostras que representem estas situações para o conjunto de dados *KITTI* através de técnicas de *Data Augmentation*. *Também sugere-se a aplicação de funções de normalização tais como* Batch Normalization *e também técnicas de regularização para proporcionar, conforme descrito na literatura, melhorias no processo de otimização.*

Outra limitação imposta é relacionada ao caráter do movimento realizado pelo veículo de captura da base de dados *KITTI*. Para aplicações com veículos que realizem rotações e translações nos eixos não considerados como aviões e drones, é necessário utilizar outra base de dados.

A exemplo de técnicas clássicas de Odometria Visual, o erro acumulado ao longo da trajetória pela integração das poses relativas foi verificado. Sobre este aspecto, entende-se que uma etapa de otimização como a aplicação de Filtro de *Kalman* Estendido na predição das transformações comum em técnicas de *VSLAM* é possível. Isto permitiria realizar correções dos erros cometidos e tornar o modelo mais robusto.

Ainda que a utilização de mapas de disparidade é destacada, outros tipos de entrada poderiam ser aplicados. A arquitetura siamesa proposta neste sentido é flexível permitindo que a mesma metodologia empregada neste trabalho seja estendida para estimativa de Odometria Visual monocular e também considera-se o uso do fluxo ótico como entrada. Por fim, considerando o aspecto temporal particular de aplicações no contexto de *OV*, cogita-se investigar se estas informações podem ser empregadas para expandir a capacidade de aprendizado da arquitetura através de arquiteturas de redes convolucionais recorrentes.

Bibliografia

[1] **The KITTI Vision Benchmark Suite a project of karlsruhe institute of technology and toyota technological institute at chicago**. http://www.cvlibs.net/datasets/kitti/. Accessed: 2017-06-14.

[2] ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G. S.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; GOODFELLOW, I.; HARP, A.; IRVING, G.; ISARD, M.; JIA, Y.; JOZEFOWICZ, R.; KAISER, L.; KUDLUR, M.; LEVENBERG, J.; MANÉ, D.; MONGA, R.; MOORE, S.; MURRAY, D.; OLAH, C.; SCHUSTER, M.; SHLENS, J.; STEINER, B.; SUTSKEVER, I.; TALWAR, K.; TUCKER, P.; VANHOUCKE, V.; VASUDEVAN, V.; VIÉGAS, F.; VINYALS, O.; WARDEN, P.; WATTENBERG, M.; WICKE, M.; YU, Y.; ZHENG, X. **TensorFlow: Large-scale machine learning on heterogeneous systems**, 2015. Software available from tensorflow.org.

[3] AGRAWAL, P.; CARREIRA, J.; MALIK, J. **Learning to see by moving**. In: *Proceedings of the IEEE International Conference on Computer Vision*, p. 37–45, 2015.

[4] ALISMAIL, H.; BROWNING, B. **Direct disparity space: Robust and real-time visual odometry**. 2014.

[5] AQEL, M. O.; MARHABAN, M. H.; SARIPAN, M. I.; ISMAIL, N. B. **Review of visual odometry: types, approaches, challenges, and applications**. *SpringerPlus*, 5(1):1897, 2016.

[6] BALAZADEGAN SARVROOD, Y.; HOSSEINYALAMDARY, S.; GAO, Y. **Visual-lidar odometry aided by reduced imu**. *ISPRS International Journal of Geo-Information*, 5(1):3, 2016.

[7] BARRON, J. L.; FLEET, D. J.; BEAUCHEMIN, S. S.; BURKITT, T. **Performance of optical flow techniques**. In: *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, p. 236–242. IEEE, 1992.

[8] BAY, H.; TUYTELAARS, T.; VAN GOOL, L. **SURF: Speeded up robust features**. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3951 LNCS:404–417, 2006.

[9] BOTTOU, L. **Stochastic gradient learning in neural networks**. *Proceedings of Neuro-Nımes*, 91(8), 1991.

[10] BROMLEY, J.; GUYON, I.; LECUN, Y.; SÄCKINGER, E.; SHAH, R. **Signature verification using a"siamese"time delay neural network**. In: *Advances in Neural Information Processing Systems*, p. 737–744, 1994.

[11] CHEN, D. M.; BAATZ, G.; KÖSER, K.; TSAI, S. S.; VEDANTHAM, R.; PYLVÄNÄINEN, T.; ROIMELA, K.; CHEN, X.; BACH, J.; POLLEFEYS, M.; OTHERS. **City-scale landmark identification on mobile devices**. In: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, p. 737–744. IEEE, 2011.

[12] CHENG, Y.; MAIMONE, M.; MATTHIES, L. **Visual Odometry on the Mars Exploration Rovers**.

[13] CHIEN, H.-J.; GENG, H.; KLETTE, R. **Improved visual odometry based on transitivity error in disparity space: A third-eye approach**. In: *Proceedings of the 29th International Conference on Image and Vision Computing New Zealand*, p. 72–77. ACM, 2014.

[14] CHOLLET, F.; OTHERS. **Keras**. https://github.com/fchollet/keras, 2015.

[15] CIARFUGLIA, T. A.; COSTANTE, G.; VALIGI, P.; RICCI, E. **Evaluation of non-geometric methods for visual odometry**. *Robotics and Autonomous Systems*, 62(12):1717–1730, 2014.

[16] CIRESAN, D. C.; MEIER, U.; MASCI, J.; GAMBARDELLA, L. M.; SCHMIDHUBER, J. **Flexible, high performance convolutional neural networks for image classification**. In: *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[17] CLEVERT, D.-A.; UNTERTHINER, T.; HOCHREITER, S. **Fast and accurate deep network learning by exponential linear units (elus)**. *arXiv preprint arXiv:1511.07289*, 2015.

[18] COSTANTE, G.; CIARFUGLIA, T. A. **Ls-vo: Learning dense optical subspace for robust visual odometry estimation**. *IEEE Robotics and Automation Letters*, 3(3):1735–1742, 2018.

[19] COSTANTE, G.; MANCINI, M.; VALIGI, P.; CIARFUGLIA, T. A. **Exploring representation learning with cnns for frame-to-frame ego-motion estimation**. *IEEE Robotics and Automation Letters*, 1(1):18–25, 2016.

[20] DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. **ImageNet: A Large-Scale Hierarchical Image Database**. In: *CVPR09*, 2009.

[21] DUCHI, J.; HAZAN, E.; SINGER, Y. **Adaptive subgradient methods for online learning and stochastic optimization**. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[22] FABIAN, J.; CLAYTON, G. **Error Analysis for Visual Odometry on Indoor, Wheeled Mobile Robots With 3-D Sensors**. *IEEE/ASME Transactions on Mechatronics*, 19(6):1896–1906, 2014.

[23] FIALA, M.; UFKES, A. **Visual Odometry Using 3-Dimensional Video Input**. 2011.

[24] FISCHER, P.; DOSOVITSKIY, A.; ILG, E.; HÄUSSER, P.; HAZIRBAŞ, C.; GOLKOV, V.; VAN DER SMAGT, P.; CREMERS, D.; BROX, T. **Flownet: Learning optical flow with convolutional networks**. *arXiv preprint arXiv:1504.06852*, 2015.

[25] FU, C.; CARRIO, A.; CAMPOY, P. **Efficient visual odometry and mapping for Unmanned Aerial Vehicle using ARM-based stereo vision pre-processing system**. *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, p. 957–962, 2015.

[26] FUENTES-PACHECO, J.; RUIZ-ASCENCIO, J.; RENDÓN-MANCHA, J. M. **Visual simultaneous localization and mapping: a survey**. *Artificial Intelligence Review*, 43(1):55–81, 2015.

[27] GARDNER, M. W.; DORLING, S. **Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences**. *Atmospheric environment*, 32(14-15):2627–2636, 1998.

[28] GEIGER, A.; LENZ, P.; URTASUN, R. **Are we ready for autonomous driving? the kitti vision benchmark suite**. In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, p. 3354–3361. IEEE, 2012.

[29] GEIGER, A.; ZIEGLER, J.; STILLER, C. **Stereoscan: Dense 3d reconstruction in real-time**. In: *Intelligent Vehicles Symposium (IV), 2011 IEEE*, p. 963–968. Ieee, 2011.

[30] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; BENGIO, Y. **Deep learning**, volume 1. MIT press Cambridge, 2016.

[31] GOVENDER, N. **Evaluation of Feature Detection Algorithms for Structure from Motion**. *Csir*, 2009.

[32] GRUPP, M. **evo**. https://github.com/MichaelGrupp/evo, 2017.

[33] HADSELL, R.; CHOPRA, S.; LECUN, Y. **Dimensionality reduction by learning an invariant mapping**. In: *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, p. 1735–1742. IEEE, 2006.

[34] HAGNELIUS, A. **Visual odometry**. *Master Thesis*, 2005.

[35] HARRIS, C.; STEPHENS, M. **A Combined Corner and Edge Detector**. *Procedings of the Alvey Vision Conference 1988*, p. 147–151, 1988.

[36] HARTLEY, R.; ZISSERMAN, A. **Multiple view geometry in computer vision**. Cambridge university press, 2003.

[37] HE, K.; ZHANG, X.; REN, S.; SUN, J. **Deep residual learning for image recognition**. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 770–778, 2016.

[38] JIANBO SHI.; C. TOMASI. **Good features to track**. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*, p. 593–600, 1994.

[39] KENDALL, A.; GRIMES, M.; CIPOLLA, R. **PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization**.

[40] KINGMA, D. P.; BA, J. **Adam: A method for stochastic optimization**. *arXiv preprint arXiv:1412.6980*, 2014.

[41] KOCH, G. **Siamese neural networks for one-shot image recognition**. PhD thesis, University of Toronto, 2015.

[42] KONDA, K. R.; MEMISEVIC, R. **Learning visual odometry with a convolutional network.** In: *VISAPP (1)*, p. 486–490, 2015.

[43] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. **Imagenet classification with deep convolutional neural networks**. In: *Advances in neural information processing systems*, p. 1097–1105, 2012.

[44] KÜMMERLE, R.; STEDER, B.; DORNHEGE, C.; RUHNKE, M.; GRISETTI, G.; STACHNISS, C.; KLEINER, A. **On measuring the accuracy of slam algorithms**. *Autonomous Robots*, 27(4):387–407, 2009.

[45] LECUN, Y.; BOSER, B. E.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W. E.; JACKEL, L. D. **Handwritten digit recognition with a back-propagation network**. In: *Advances in neural information processing systems*, p. 396–404, 1990.

[46] LECUN, Y.; CORTES, C. **MNIST handwritten digit database**. 2010.

[47] LECUN, Y.; KAVUKCUOGLU, K.; FARABET, C.; OTHERS. **Convolutional networks and applications in vision.** In: *ISCAS*, volume 2010, p. 253–256, 2010.

[48] LECUN, Y.; OTHERS. **Generalization and network design strategies**. *Connectionism in perspective*, p. 143–155, 1989.

[49] LEMAIRE, T.; BERGER, C.; JUNG, I.-K.; LACROIX, S.; CNRS, L.; ROCHE, C.; CEDEX, F.-T. **Vision-Based SLAM : Stereo and Monocular Approaches**. 74(3):343–364, 2007.

[50] LI, R.; WANG, S.; LONG, Z.; GU, D. **Undeepvo: Monocular visual odometry through unsupervised deep learning**. *arXiv preprint arXiv:1709.06841*, 2017.

[51] LOWE, D. G. **Distinctive image features from scale-invariant keypoints**. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[52] LUCAS, B. D.; KANADE, T.; OTHERS. **An iterative image registration technique with an application to stereo vision**. 1981.

[53] MAIMONE, M.; CHENG, Y.; MATTHIES, L. **Two years of visual odometry on the Mars Exploration Rovers**. *Journal of Field Robotics*, 24(3):169–186, 2007.

[54] MEHDIPOUR-GHAZI, M.; EKENEL, H. K. **A comprehensive analysis of deep learning based representation for face recognition**. *CoRR*, abs/1606.02894, 2016.

[55] MILELLA, A.; SIEGWART, R. **Stereo-based ego-motion estimation using pixel tracking and iterative closest point**. *Proceedings of the Fourth IEEE International Conference on Computer Vision Systems, ICVS'06*, 2006(Icvs):21, 2006.

[56] MOHANTY, V.; AGRAWAL, S.; DATTA, S.; GHOSH, A.; SHARMA, V. D.; CHAKRAVARTY, D. **Deepvo: A deep learning approach for monocular visual odometry**. *arXiv preprint arXiv:1611.06069*, 2016.

[57] MUELLER, J.; THYAGARAJAN, A. **Siamese recurrent architectures for learning sentence similarity.** In: *AAAI*, p. 2786–2792, 2016.

[58] MULLER, P.; SAVAKIS, A. **Flowdometry: An optical flow and deep learning based approach to visual odometry**. In: *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, p. 624–631. IEEE, 2017.

[59] MULLER, P. M. **Optical flow and deep learning based approach to visual odometry**. 2016.

[60] MULLER, P. M. **Optical Flow and Deep Learning Based Approach to Visual Odometry Optical Flow and Deep Learning Based Approach to Visual Odometry**. 2016.

[61] NAGATANI, K.; IKEDA, A.; ISHIGAMI, G.; YOSHIDA, K.; NAGAI, I. **Development of a visual odometry system for a wheeled robot on loose soil using a telecentric camera**. *Advanced Robotics*, 24(8-9):1149–1167, 2010.

[62] NESTEROV, Y. **Smooth minimization of non-smooth functions**. *Mathematical programming*, 103(1):127–152, 2005.

[63] ROBERTS, R. J. W. **Optical flow templates for mobile robot environment understanding**. PhD thesis, Georgia Institute of Technology, 2014.

[64] ROSENBLATT, F. **Principles of neurodynamics. perceptrons and the theory of brain mechanisms**. Technical report, CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1961.

[65] ROSTEN, E.; DRUMMOND, T. **Machine learning for high-speed corner detection**. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3951 LNCS:430–443, 2006.

[66] SCARAMUZZA, B. D.; FRAUNDORFER, F. **Visual Odometry**. (December), 2011.

[67] SCARAMUZZA, D.; FRAUNDORFER, F. **Tutorial: Visual odometry**. *IEEE Robotics and Automation Magazine*, 18(4):80–92, 2011.

[68] SCHERER, D.; MÜLLER, A.; BEHNKE, S. **Evaluation of pooling operations in convolutional architectures for object recognition**. In: *Artificial Neural Networks–ICANN 2010*, p. 92–101. Springer, 2010.

[69] SCHMIDHUBER, J. **Deep learning in neural networks: An overview**. *Neural networks*, 61:85–117, 2015.

[70] SIMONYAN, K.; ZISSERMAN, A. **Very deep convolutional networks for large-scale image recognition**. *arXiv preprint arXiv:1409.1556*, 2014.

[71] SMITH, R.; SELF, M.; CHEESEMAN, P. **Estimating uncertain spatial relationships in robotics**. *arXiv preprint arXiv:1304.3111*, 2013.

[72] STURM, J.; ENGELHARD, N.; ENDRES, F.; BURGARD, W.; CREMERS, D. **A benchmark for the evaluation of rgb-d slam systems**. In: *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, p. 573–580. IEEE, 2012.

[73] SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. **Going deeper with convolutions**. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 1–9, 2015.

[74] TARRIO, J. J.; PEDRE, S. **Realtime Edge-Based Visual Odometry for a Monocular Camera**. *2015 IEEE International Conference on Computer Vision (ICCV)*, p. 702–710, 2015.

[75] TIELEMAN, T.; HINTON, G. **Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude**. COURSERA: Neural Networks for Machine Learning, 2012.

[76] VEIT, A.; KOVACS, B.; BELL, S.; MCAULEY, J.; BALA, K.; BELONGIE, S. **Learning visual clothing style with heterogeneous dyadic co-occurrences**. In: *Proceedings of the IEEE International Conference on Computer Vision*, p. 4642–4650, 2015.

[77] VON NEUMANN, J.; OTHERS. **The general and logical theory of automata**. *Cerebral mechanisms in behavior*, 1(41):1–2, 1951.

[78] WANG, S.; CLARK, R.; WEN, H.; TRIGONI, N. **Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks**. In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, p. 2043–2050. IEEE, 2017.

[79] WIRTH, S.; CARRASCO, P. L. N.; CODINA, G. O. **Visual odometry for autonomous underwater vehicles**. In: *OCEANS-Bergen, 2013 MTS/IEEE*, p. 1–6. IEEE, 2013.

[80] XU, Y.; XIAO, T.; ZHANG, J.; YANG, K.; ZHANG, Z. **Scale-invariant convolutional neural networks**. *CoRR*, abs/1411.6369, 2014.

[81] YAMAGUCHI, K.; HAZAN, T.; MCALLESTER, D.; URTASUN, R. **Continuous markov random fields for robust stereo estimation**. *Computer Vision–ECCV 2012*, p. 45–58, 2012.

[82] YAMAGUCHI, K.; MCALLESTER, D.; URTASUN, R. **Efficient joint segmentation, occlusion labeling, stereo and flow estimation**. In: *European Conference on Computer Vision*, p. 756–771. Springer, 2014.

[83] YOUSIF, K.; BAB-HADIASHAR, A.; HOSEINNEZHAD, R. **An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics**. *Intelligent Industrial Systems*, 1(4):289–311, 2015.

[84] ZAMBANINI, S.; KAMPEL, M. **A local image descriptor robust to illumination changes**. In: *Scandinavian Conference on Image Analysis*, p. 11–21. Springer, 2013.