Odometria Multifrequência para Veículos Aéreos Autônomos

Patrick de C. T. R. Ferreira[1]*, **Esther L. Colombini**[1]

[1]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
CEP 10.583-852 – Campinas – SP – Brasil

p175480@dac.unicamp.br, esther@ic.unicamp.br

Abstract. *The navigation of autonomous aerial vehicles requires reliable estimates of state variables such as their position e orientation. Several control methods exist for this purpose, but the inability to accurately estimate the positioning makes autonomous navigation unfeasible, especially in closed environments. This work Proposto a framework for visual-inertial odometry based on neural networks capable of working with sensors at multiple input frequencies, combining the precision of visual localization methods with the high update rates of inertial methods. The framework is modular and achieves an ATE deviation around 0.30m, RPE of 2.5% e 2.0 degrees/s, in addition to a refresh rate of around 200Hz.*
Keywords*: Odometry, Multifrequency, Autonomous Aerial Vehicles.*

Resumo. *A navegação de veículos aéreos autônomos exige estimativas confiáveis de variáveis de estado como posição e orientação dos mesmos. Existem diversos métodos de controle para este fim, mas a inabilidade em estimar com precisão o próprio posicionamento inviabiliza a navegação autônoma, principalmente em ambientes fechados. Este trabalho propôs um framework para odometria visual-inercial baseado em redes neurais capaz de trabalhar com sensores em múltiplas frequências de entrada, aliando a precisão dos métodos de localização visual com as altas taxas de atualizações de métodos inerciais. O framework é modular e apresenta desvio ATE em torno de 0.30m, RPE de 2.5% e 2.0 graus/s, além de taxa de atualização de cerca de 200Hz.*
Palavras-Chave*: Odometria, Multifrequencia, Veículos Aéreos Autônomos.*

*Nível do estudante: M.Sc. Defendido em 21/06/2022.
Orientadora: Profa. Dra. Esther Luna Colombini.
Repositório: github.com/patrickctrf/multifrequency-odometry

1. Introdução

A crescente popularidade de veículos aéreos autônomos, também conhecidos no inglês por *Unnamed Aerial Vehicles* (UAVs), é um indicador do impacto que este tipo de tecnologia está tendo e ainda trará no cenário econômico mundial. Segundo o serviço de imprensa financeira Business Wire, da Berkshire Hathaway, o mercado global de drones comerciais foi avaliado em US$2,72 bilhões em 2020 e deve atingir US$21,69 bilhões até 2030, registrando um CAGR de 23,7%[BusinessWire]. Porém, em situações em que se requer uma precisão mais elevada que a do GPS, em ambientes em que o sinal deste é negado ou o tempo de resposta das trajetórias precisa ser menor, faz-se necessário o uso de técnicas de sensoriamento mais precisas e velozes para viabilizar a navegação autônoma.

Tentativas de implementar técnicas de controle tradicionais para localização de UAVs inevitavelmente falham devido à natureza não linear do problema[Wyeth et al. 2019].

Diante da necessidade de estimativas de localização mais acuradas e sem o uso de sensores pesados e caros para o veículo aéreo, este trabalho propõe uma técnica de localização baseada em software de redes neurais operando sobre leituras de sensores inerciais e visuais, a fim de se obter variáveis de estado confiáveis: Posição e orientação do UAV. Estas redes neurais desempenharão o cálculo de odometria, que consiste na técnica de estimar o deslocamento global de um objeto através da integração de sucessivos deslocamentos parciais. Para tal, são desenvolvidos módulos que desempenham sobre leituras de uma unidade de medição inercial (IMU) contendo acelerômetro e giroscópio, além das entradas visuais de uma câmera acoplada ao veículo.

Como principal contribuição, ao final deste trabalho tem-se a produção de um *framework* multifrequência para estimativa de localização através de odometria. A rede neural utilizada neste *framework* é capaz de atualizar sua saída a cada nova leitura dos sensores fornecida como entrada, mesmo que operem em frequências distintas, sem necessidade de subamostragem e permitindo maior taxa de atualização de suas estimativas. Até onde temos conhecimento, é o primeiro *framework* da categoria construído.

2. Trabalhos Relacionados

Em 2017, a primeira rede neural para odometria visual-inercial (VIO) foi apresentada utilizando *deep learning* em uma abordagem *end-to-end*, a chamada VINet[Clark et al. 2017]. Este modelo baseava-se em células LSTM para comprimir janelas de amostragem dos sensores inerciais em um estado de representação latente, de tal forma que a entrada para o segundo estágio dessa rede se ajustasse à frequência de amostragem dos dados visuais, usualmente mais lentos. Esta foi uma solução inicial para o problema de odometria visual-inercial, no entanto, a compressão de amostras inerciais em um vetor latente descarta a possibilidade de atualizar a saída a cada nova amostra inercial inserida, uma vantagem considerável de *frameworks* multifrequência.

Em 2020, Kaufman et al. [Kaufmann et al. 2020] apresentaram a primeira arquitetura que funciona em multifrequência. Visando exercer controle de UAVs para acrobacias, não é um método para localização ou odometria. Por não propor a resolver o problema de localização, mas apenas o de controle do UAV, não resolve o problema do acúmulo de erros na estimativa de variáveis de estado, mas fornece uma proposta para arquiteturas com sensores operando em multifrequência.

Ainda no campo de algoritmos para estimativa de localização, temos os métodos baseados em Mapeamento e Localização Simultâneos (do inglês, SLAM). Destes, podemos destacar o ORB-SLAM[Mur-Artal et al. 2015], um método baseado em *features* visuais e que será utilizado como módulo de navegação visual neste trabalho, dada sua popularidade na literatura e bom desempenho. A vantagem deste tipo de algoritmo sobre a odometria é a promessa de um menor acúmulo de erros na trajetória, enquanto que as desvantagens são o custo de demandar maior capacidade computacional para execução, maior latência na taxa de atualização e a possibilidade de quebra do mapeamento, caso o veículo realize um movimento brusco com sua câmera.

Há também métodos visuais como o BASALT[Usenko et al. 2019], que realizam o cálculo de odometria através de imagens como entrada. A opção de utilizar a odometria

ao invés de SLAM permitiria ao algoritmo processar seus dados de entra mais rápido que os métodos de SLAM, e seu bom desempenho reportado na literatura é o motivo pelo qual utilizamos BASALT como segunda opção de módulo visual nesta pesquisa.

Por último, no campo de odometria puramente inercial (que conta somente com sensores inerciais como fonte de informação) para veículos aéreos, podemos destacar o algoritmo TLIO[Liu et al. 2020], que faz uso de um filtro de Kalman extendido para integrar estimativas de deslocamento advindas de uma rede neural e de um método analítico para realizar a tarefa de odometria. Este método apresenta menor precisão que os algoritmos visuais apresentados anteriormente, porém é mais rápido por trabalhar apenas com informação inercial, e servirá de comparação para o modelo inercial proposto adiante.

3. Materiais e Métodos

A arquitetura do framework de odometria proposto é baseada em dois módulos independentes, um usando odometria visual ou SLAM visual e o outro odometria inercial, que têm suas estimativas combinadas através de um terceiro módulo dedicado à fusão sensorial (figura 1). Este terceiro bloco recebe as entradas das duas sub-redes anteriores e atualiza a estimativa de pose do UAV a cada nova amostra, podendo realizá-la através de blocos LSTM [Clark et al. 2017] e/ou camadas de convolução temporal [Kaufmann et al. 2020].

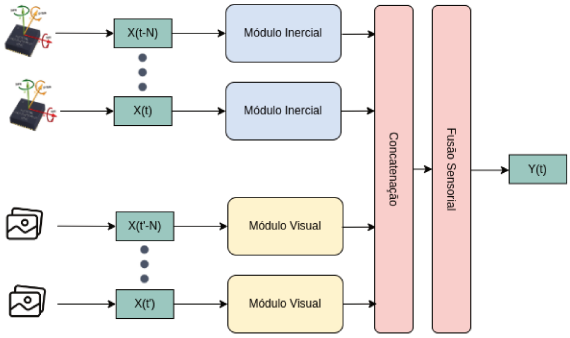


Figura 1. Arquitetura do *framework* proposto.

$X(t)$ é a leitura da IMU no momento t (mais recente) e $X(t')$, a amostra mais recente do módulo visual. Usamos t' para explicar que essas amostras são recebidas de forma assíncrona e em diferentes frequências. $y(t)$ é a estimativa de pose atual da rede. A razão pela qual também inserimos cálculos de deslocamentos anteriores é para promover a consistência da análise, pois a rede pode considerar o próprio erro nas estimativas anteriores e descontá-lo da chegada de novas leituras mais atuais dos sensores [Clark et al. 2017, Saputra et al. 2020, Kaufmann et al. 2020]. Após concatenar a resposta de cada bloco em seus respectivos ramos, cabe ao módulo de fusão sensorial verificar a relação entre as amostras atuais e passadas e estimar a posição atual do UAV em termos de tradução (x, y, z) e rotação de quatérnios (qw, qx, qy, qz). Neste bloco, também habilitamos a entrada multifrequência, pois o bloco visual nos dá o deslocamento atual. Em contraste, o bloco inercial é responsável por nos informar o deslocamento **desde o último recurso visual recebido**. É como se a rede adicionasse o deslocamento de ambos os módulos, mas ponderasse eficientemente a relação entre eles e corrigisse possíveis imperfeições, conhecimento adquirido durante o treinamento e que caracteriza

a principal vantagem das redes neurais sobre os métodos clássicos, que só podem corrigir os fenômenos contemplados em seu projeto.

O módulo visual é responsável por emitir uma estimativa de posicionamento concisa periodicamente com menor erro acumulado em menor frequência. Ao mesmo tempo, o bloco inercial é responsável por atualizar a saída da rede em um tempo de resposta menor. Diferentes métodos de localização visual serão testados e avaliados, uma vez que o projeto propõe que este módulo possa ser alternado para melhor aproveitar as características de cada processo existente (robustez, rapidez, precisão e afins). A arquitetura de rede é projetada especificamente para que o método de estimativa de pose visual usado seja fácil de substituir (figura 1).

3.1. Módulo Inercial

A arquitetura desenvolvida para o módulo inercial utiliza somente leituras inerciais da IMU que está acoplada ao UAV. Por não utilizar imagens como entrada, seu processamento é mais rápido, permitindo alcançar taxas de atualização da saída de 200Hz (taxa de atualização da IMU). A desvantagem de usar um método de odometria inercial sobre métodos de odometria visual é o acúmulo inerente de erros ao longo do caminho.

Duas abordagens diferentes serão avaliadas para o módulo (ou bloco) inercial: usando **convoluções temporais** e usando **redes recorrentes**. Ambos podem lidar com tamanho de sequência de entrada variável e têm vantagens diferentes ao lidar com séries temporais.

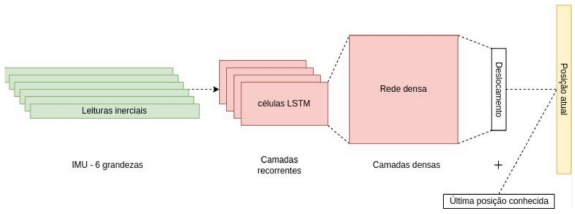


Figura 2. Estimador de deslocamento inercial ao usar redes recorrentes.

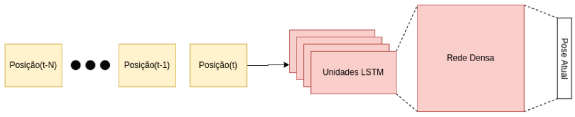


Figura 3. Estimador de posição inercial ao usar redes recorrentes.

Ambas as abordagens utilizam um princípio de construção do módulo em 2 estágios: Uma primeira rede neural (figuras 2 e 4) que computa o deslocamento relativo a uma janela de 200 amostras da IMU (equivalente a 1 segundo de amostragem), e um segundo estágio (figuras 3 e 5) com outra rede neural treinada para receber estes deslocamentos parciais e integrá-los em um único deslocamento global. O módulo recorrente (figuras 2 e 3) baseado em LSTMs tem a vantagem de requerer menos processamento que o módulo convolucional (figuras 4 e 5), enquanto que este pode ter vantagem em tempo de treinamento por evitar o desaparecimento de gradiente, típico de redes recorrentes. Ambos serão testados na seção de resultados, de forma a verificar o quanto estas características se manifestam na prática.

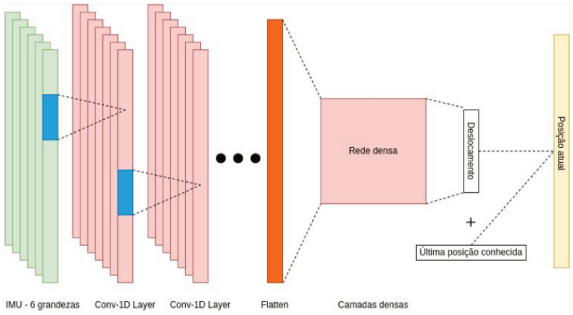


Figura 4. Estimador de deslocamento inercial ao usar redes convolucionais.

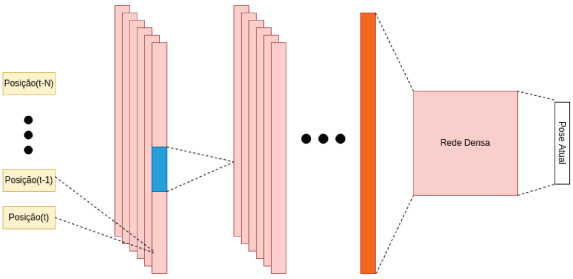


Figura 5. Estimador de posição inercial ao usar redes convolucionais.

3.2. Módulo Visual

Dada a elevada quantidade de algoritmos de localização visual presentes na literatura, optamos por utilizar o ORB-SLAM e o BASALT nesta função do *framework* proposto, já que as principais contribuições desta pesquisa são a produção de uma estrutura capaz de desempenhar odometria com sensores em múltiplas frequências, além de aceitar entradas de tamanho variável, caso haja atraso no processamento dos dados de entrada, e modularidade no quesito de permitir a substituição dos módulos visuais sem retreinamento. Testes para verificação do desempenho de cada uma das configurações e com a substituição dos módulos com e sem retreinamento são exibidos na seção de resultados.

3.3. Datasets

Diferentes bases de dados estão disponíveis para treinamento de navegação inercial, com entradas de acelerômetro e giroscópio, e saídas (também chamadas de *target* ou *ground truth*) em termos de posição e orientação, dentre as quais destacamos TUM[Schubert et al. 2018] e EuRoC[Burri et al. 2016]. O dataset EuRoC possui alinhamento entre entrada e saída, facilitando seu uso para treinamento e validação, enquanto que o dataset TUM é utilizado, neste caso, para testes exclusivamente, de forma a evitar possíveis viéses de uma mesma base de dados no momento de testar a performance do *framework* proposto. Mantemos também as sequências MH_02, MH_04 e V2_02 do EuRoC para teste, para aumentar o conjunto de avaliação.

3.4. Treinamento e Avaliação

Para treinar o primeiro estágio dos módulos inerciais, é necessário realizar o split dos datasets em janelas de comprimento N, o qual é realizado de maneira não trivial, já que necessitamos referenciar a variação de posicionamento em relação à posição atual do

veículo, utilizando as equações 1 e 2, onde $\mathbf{R}$ é a matriz de rotação associada à orientação do veículo e $\mathbf{p}$ é a posição translacional do mesmo.

$$\Delta\mathbf{R} = \mathbf{R}_{t-N}^{T}\mathbf{R}_{t} \tag{1}$$

$$\Delta\mathbf{p} = \mathbf{R}_{t-N}^{T}(\mathbf{p_t} - \mathbf{p_{t-N}}) \tag{2}$$

A loss (equação 3) de treinamento utilizada penaliza o erro translacional (posição, p) a partir da raíz do erro quadrático médio (RMSE), enquanto que o erro dos quatérnios (orientação, q) é penalizado com a similaridade por cosseno, pois apenas é relevante a direção estimada para a orientação do veículo, e não a magnitude da mesma. Esta abordagem mostra bons resultados em tarefas de odometria, como reportado em [Huynh 2009].

$$Loss = ||\hat{p}_i - p_i||_2 + 1 - |\hat{q} \cdot q| \tag{3}$$

$$= RMSE(\hat{p}_i,\, p_i) + 1 - SimilaridadePorCosseno(\hat{q},\, q). \tag{4}$$

As métricas de erro utilizadas para avaliação do *framework* são o erro absoluto de trajetória (ATE) e o erro relativo de pose (RPE). O ATE (equação 5) é muito semelhante ao RMSE, com a diferença de que as trajetórias preditas e refenciadas são alinhadas antes do cálculo, pois o que nos interessa é a forma de trajetoria, já que o veículo adota seu próprio referencial ao invés do referencial escolhido pelo observador externo. Já o RPE (equação 6) pode ser interpretado como uma medida de deformação da trajetória, sendo calculado em intervalos da mesma.

$$ATE = \min_{\boldsymbol{T}\in SE}\sqrt{\frac{1}{|I_{gt}|}\sum_{i\in I_{gt}}\|\boldsymbol{T}\hat{p}_i - p_i\|^2}. \tag{5}$$

$$RPE = \sqrt{\frac{1}{I_{gt,\Delta}}\sum_{i\in I_{gt,\Delta}}\|trans(\mathbf{E}_i)\|^2}, \tag{6}$$

com E_i dado por:

$$\mathbf{E}_i = (\hat{\mathbf{T}}_i^{-1}\hat{\mathbf{T}}_{i+\Delta}^{-1})(\mathbf{T}_i^{-1}\mathbf{T}_{i+\Delta}^{-1}). \tag{7}$$

3.5. Arquitetura das Redes Neurais

Existem dois tipos de redes neurais sendo utilizadas nesta pesquisa: Convolucional e recorrente. A tabela 1 mostra a arquitetura da rede convolucional temporal, enquanto a tabela 2 mostra a arquitetura da rede recorrente. A saída de todos eles é composta por 7 dimensões, que correspondem aos dados translacionais e rotacionais em quatérnions (px, py, pz, qw, qx, qy, qz). Para o primeiro estágio do módulo inercial, essa saída corresponde ao deslocamento parcial da rede calculado, enquanto para o segundo estágio desses módulos e para a unidade de fusão sensorial do módulo visual-inercial, a saída corresponde à estimativa de posicionamento global atual. A única diferença entre a arquitetura utilizada pelos segundos estágios dos módulos inerciais e o módulo de fusão sensorial é que o *framework* visual-inercial concatena as estimativas de cada um destes em diferentes canais antes de enviá-los para a rede. No caso do segundo estágio do módulo inercial, apenas os canais de saída deste módulo são entregues à arquitetura.

Tabela 1. Arquitetura do módulo inercial convolucional.

Tipo	Kernel	Dilatação	Stride	Filtros	Activação
Conv1D	3	2	3	256	LeakyReLU
Conv1D	3	1	1	256	LeakyReLU
Conv1D	3	1	1	256	LeakyReLU
Conv1D	3	1	1	256	LeakyReLU
Conv1D	3	1	1	256	LeakyReLU
Conv1D	3	1	1	256	LeakyReLU
Conv1D	3	1	1	256	LeakyReLU
Conv1D	3	1	1	256	LeakyReLU
Global Average Pooling	—	—	—	—	—
Densa	—	—	—	512	LeakyReLU
Densa	—	—	—	7	Linear

Tabela 2. Arquitetura do módulo inercial recorrente.

Tipo	Vetor Latente	Bidirecional	Neurônios	Ativação
LSTM	256	True	256	Sig.+TanH
LSTM	256	True	256	Sig.+TanH
LSTM	256	True	256	Sig.+TanH
Densa	—	—	512	LeakyReLU
Densa	—	—	7	Linear

4. Resultados

Após o treinamento das redes inerciais e do módulo de fusão sensorial, realizamos as avaliações sobre as sequências de teste. A tabela 3 mostra a comparação do módulo inercial desenvolvido neste projeto contra outros métodos disponíveis na literatura. Nota-se que o TLIO, dentre os inerciais puros, ainda possui melhor desempenho na média, possivelmente pela adição de um filtro de Kalman, o que pode tornar suas estimativas menos ruidosas. Entretanto, sob um custo computacional reduzido, conseguimos propor um módulo unicamente inercial mais veloz (tabela 12) que se aproxima do desempenho, e o mais importante para este módulo é sua velocidade, já que a maior parte da precisão do *framework* como um todo virá dos módulos visuais.

As tabelas 4 e 5 apresentam o desempenho dos diferentes módulos inerciais (recorrente e convolucional) em associação com os módulos visuais selecionados, BASALT e ORB-SLAM. Verificou-se que a inserção de dados preprocessados na entrada do módulo, como *wavelets* e envelope de sinal (*features*), não afetavam significativamente seu desempenho. Nota-se que o módulo convolucional possui um desempenho minimamente superior ao recorrente, motivo pelo qual servirá de base nos próximos experimentos. Já quanto ao *framework* como um todo, BASALT aparenta ter melhor desempenho, sendo que ambos os módulos visuais têm sua performance levemente prejudicada pela adição do sinal inercial, tipicamente ruidoso, o que piora as métricas, mas não atrapalha a trajetória global, como será mostrado adiante. As tabelas de 6 a 11 demonstram o desempenho

Tabela 3. Comparação entre TLIO, ORB-SLAM e nosso módulo inercial proposto.

	Sequência	Comprimento[m]	ORB-SLAM	TLIO	Módulo Inercial
ATE	MH_01	79.84	0.14	3.92	2.37
	MH_04	91.55	0.25	4.61	5.22
	V2_02	83.01	0.04	1.01	2.52
RPE %	MH_01	79.84	5.60	18.39	21.81
	MH_04	91.55	12.50	19.25	21.54
	V2_02	83.01	2.00	18.64	18.79
RPE o/s	MH_01	79.84	1.54	10.66	12.60
	MH_04	91.55	3.75	12.42	13.43
	V2_02	83.01	0.80	7.42	9.45

Tabela 4. Modelo proposto em associação com ORB-SLAM as visual module.

Métrica	Extrator de características				Métodos comparativos	
	CNN	LSTM	CNN+Feat.	LSTM+Feat.	ORB-SLAM	BASALT
Max ATE [m]	0.32	0.32	0.31	0.32	0.32	0.09
RMS ATE [m]	0.22	0.23	0.22	0.22	0.21	0.07
Max RPE [%]	0.21	0.22	0.21	0.21	0.20	0.06
RMS RPE [%]	0.14	0.15	0.15	0.15	0.13	0.05
Latência [ms]	9.09	10.53	13.59	14.45	50.00	29.39

do módulo inercial em conjunto com os módulos visuais propostos, além de demonstrar como o desempenho se mantém mesmo quando substituímos os módulos visuais por outro sem necessidade de retreinar a rede, demonstrando a modularidade do *framework*. Os valores de erro do *framework* se mantiveram próximos daqueles obtidos pelos módulos visuais individualmente, apesar do ruído inerente às estimativas inerciais que garantem a taxa de atualização mais elevado, demonstrando novamente o sucesso do *framework* na tarefa de fusão sensorial.

Por fim, a tabela 12 apresenta o ganho em frequência de atualização do *framework* em relação aos módulos visuais originais operando individualmente. As comparações de hardware são entre um laptop 7200u Intel core i5 com GPU 940MX e uma placa NVIDIA Jetson NANO (hardware com consumo de 10W, mais adequado para um UAV). O teste do p-valor é usado para comparar a latência do modelo proposto usando o ORB-SLAM

Tabela 5. Proposto modelo em associação com BASALT como módulo visual.

Métrica	Extrator de características				Métodos comparativos	
	CNN	LSTM	CNN+Feat.	LSTM+Feat.	ORB-SLAM	BASALT
Max ATE [m]	0.10	0.10	0.09	0.10	0.31	0.09
RMS ATE [m]	0.07	0.07	0.07	0.07	0.21	0.06
Max RPE [%]	0.07	0.07	0.07	0.07	0.19	0.06
RMS RPE [%]	0.04	0.05	0.05	0.05	0.13	0.05
Latência [ms]	5.94	6.86	8.91	9.38	50.00	29.39

Tabela 6. Métricas ATE [m], RPE [%] e RPE [graus/s] (de cima para baixo) para o modelo proposto treinado e testado com ORB-SLAM.

	Sequência	Comprimento[m]	ORB-SLAM	BASALT	Proposto + ORB-SLAM
ATE	MH_01	79.84	0.14	0.08	0.15
	MH_04	91.55	0.25	0.10	0.38
	V2_02	83.01	0.04	0.02	0.09
RPE %	MH_01	79.84	5.60	1.60	5.81
	MH_04	91.55	12.50	2.00	12.54
	V2_02	83.01	2.00	0.40	0.84
RPE o/s	MH_01	79.84	1.54	1.76	1.62
	MH_04	91.55	3.75	2.10	3.29
	V2_02	83.01	0.80	0.42	0.54

Tabela 7. Métricas ATE [m], RPE [%] e RPE [graus/s] (de cima para baixo) para o modelo proposto treinado ORB-SLAM e testado com BASALT.

	Sequência	Comprimento[m]	ORB-SLAM	BASALT	Proposto + BASALT
ATE	MH_01	79.84	0.14	0.08	0.11
	MH_04	91.55	0.25	0.10	0.27
	V2_02	83.01	0.04	0.02	0.17
RPE %	MH_01	79.84	5.60	1.60	2.48
	MH_04	91.55	12.50	2.00	3.17
	V2_02	83.01	2.00	0.40	0.57
RPE o/s	MH_01	79.84	1.54	1.76	2.08
	MH_04	91.55	3.75	2.10	2.51
	V2_02	83.01	0.80	0.42	0.37

(configuração mais lenta) e comprovar estatisticamente que houve diferença (melhoria) da mesma. O teste é realizado coletando o tempo de resposta para cada amostra de uma sequência inteira do dataset (EuRoC MH_04). A figura 6 mostra o resultado da reconstrução de uma das trajetórias de teste usando o *framework* visual-inercial proposto, e também uma reconstrução usando apenas informações inerciais com o módulo inercial, para fins de comparação do ganho de desempenho.

5. Conclusão e Trabalhos Futuros

Esta pesquisa desenvolveu um *framework* multifrequência para cálculo de odometria através de redes neurais profundas. Demonstramos ser possível a criação de um módulo inercial individual de odometria, o qual pode ser fundido a outro método visual mantendo a alta taxa de atualização da odometria inercial com o nível de precisão dos métodos visuais. Além disso, o *framework* visual-inercial apresentou boa modularidade, permitindo seus componentes serem substituídos sem grandes perdas de desempenho, sendo o primeiro de sua categoria na literatura. Em trabalhos futuros, pretendemos aplicar o algoritmo desenvolvido em um UAV físico e testá-lo sob ambiente com sistema de captura de pose em tempo real para avaliação da precisão em um cenário real.

Tabela 8. Métricas ATE [m], RPE [%] e RPE [graus/s] (de cima para baixo) para o Proposto model trained with BASALT e testado com ORB-SLAM.

	Sequência	Comprimento[m]	ORB-SLAM	BASALT	Proposto + BASALT
ATE	MH_01	79.84	0.14	0.08	0.15
	MH_04	91.55	0.25	0.10	0.66
	V2_02	83.01	0.04	0.02	0.33
RPE %	MH_01	79.84	5.60	1.60	5.94
	MH_04	91.55	12.50	2.00	12.95
	V2_02	83.01	2.00	0.40	1.09
RPE o/s	MH_01	79.84	1.54	1.76	1.42
	MH_04	91.55	3.75	2.10	3.33
	V2_02	83.01	0.80	0.42	0.62

Tabela 9. Métricas ATE [m], RPE [%] e RPE [graus/s] (de cima para baixo) for Proposto model trained e testado com BASALT.

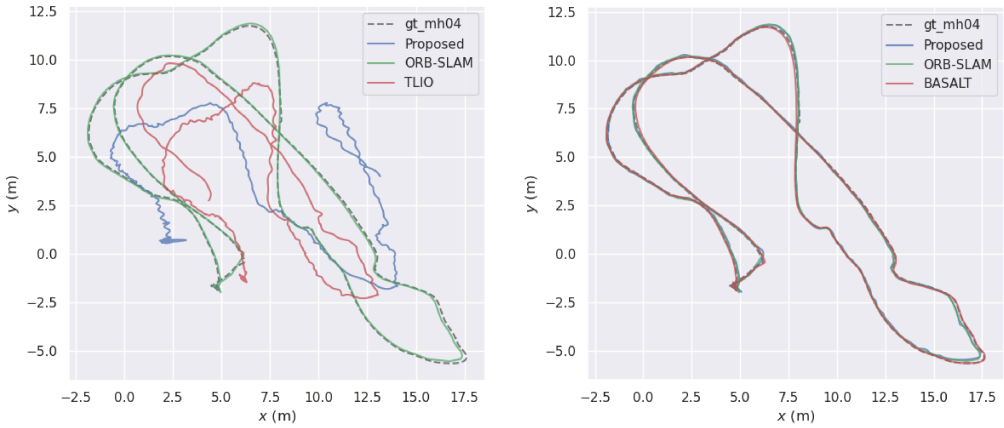
	Sequência	Comprimento[m]	ORB-SLAM	BASALT	Proposto + BASALT
ATE	MH_01	79.84	0.14	0.08	0.10
	MH_04	91.55	0.25	0.10	0.14
	V2_02	83.01	0.04	0.02	0.04
RPE %	MH_01	79.84	5.60	1.60	2.44
	MH_04	91.55	12.50	2.00	3.01
	V2_02	83.01	2.00	0.40	0.41
RPE o/s	MH_01	79.84	1.54	1.76	2.05
	MH_04	91.55	3.75	2.10	2.31
	V2_02	83.01	0.80	0.42	0.35

Tabela 10. Métricas ATE [m], RPE [%] e RPE [graus/s] (de cima para baixo) para o modelo proposto com ORB-SLAM no TUM dataset.

	Sequência	Comprimento[m]	ORB-SLAM	BASALT	Proposto+ ORB-SLAM
ATE	room1	146.00	0.15	0.09	0.16
	room2	142.00	0.13	0.07	0.21
	room3	135.00	0.17	0.13	0.20
RPE %	room1	146.00	1.72	0.78	1.96
	room2	142.00	1.76	0.94	2.10
	room3	135.00	2.54	1.13	2.74
RPE o/s	room1	146.00	0.66	0.52	0.89
	room2	142.00	0.52	0.21	0.65
	room3	135.00	0.67	0.41	0.93

Tabela 11. Métricas ATE [m], RPE [%] e RPE [graus/s] (de cima para baixo) para o modelo proposto testado com BASALT no TUM dataset.

	Sequência	Comprimento[m]	ORB-SLAM	BASALT	Proposto+ BASALT
ATE	room1	146.00	0.15	0.09	0.11
	room2	142.00	0.13	0.07	0.12
	room3	135.00	0.17	0.13	0.19
RPE %	room1	146.00	1.72	0.78	0.89
	room2	142.00	1.76	0.94	1.31
	room3	135.00	2.54	1.13	1.53
RPE o/s	room1	146.00	0.66	0.52	0.68
	room2	142.00	0.52	0.21	0.33
	room3	135.00	0.67	0.41	0.59



(a) Módulo inercial proposto.

(b) *Framework* visual-inercial proposto.

Figura 6. Trajetórias do *framework* visual-inercial (com ORB-SLAM) e do módulo inercial em comparação com outros métodos sobre a sequência de testes MH_04 do EuRoC.

Tabela 12. Comparação da taxa de atualização do modelo sob diferentes hardwares.

Algoritmo	Latência [ms]	Frequência [Hz]	p-valor	Plataforma
ORB-SLAM	50.00	20.00	0.007	Laptop
BASALT	28.37	35.25	0.011	Laptop
TLIO	14.56	68.68	0.016	Laptop
Proposto+ORB-SLAM	5.14	194.55	—	Laptop
Proposto+BASALT	4.89	204.50	—	Laptop
ORB-SLAM	50.00	20.00	0.010	Jetson
BASALT	40.56	24.65	0.013	Jetson
TLIO	22.13	45.19	0.016	Jetson
Proposto+ORB-SLAM	7.66	130.55	—	Jetson
Proposto+BASALT	7.32	136.61	—	Jetson

Referências

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M., and Siegwart, R. (2016). The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35.

BusinessWire. Global Commercial Drones Market Opportunity Analysis and Industry Forecasts, 2021-2022 & 2030 - ResearchAndMarkets.com | Business Wire.

Clark, R., Wang, S., Wen, H., Markham, A., and Trigoni, N. (2017). VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem. *arXiv:1701.08376 [cs]*. arXiv: 1701.08376.

Huynh, D. Q. (2009). Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision 2009 35:2*, 35:155–164.

Kaufmann, E., Loquercio, A., Ranftl, R., Müller, M., Koltun, V., and Scaramuzza, D. (2020). Deep Drone Acrobatics. *arXiv:2006.05768 [cs]*. arXiv: 2006.05768.

Liu, W., Caruso, D., Ilg, E., Dong, J., Mourikis, A. I., Daniilidis, K., Kumar, V., and Engel, J. (2020). TLIO: Tight Learned Inertial Odometry. *IEEE Robotics and Automation Letters*, 5(4):5653–5660.

Mur-Artal, R., Montiel, J. M., and Tardos, J. D. (2015). ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163.

Saputra, M. R. U., de Gusmao, P. P. B., Lu, C. X., Almalioglu, Y., Rosa, S., Chen, C., Wahlström, J., Wang, W., Markham, A., and Trigoni, N. (2020). DeepTIO: A Deep Thermal-Inertial Odometry with Visual Hallucination. *arXiv:1909.07231 [cs]*. arXiv: 1909.07231.

Schubert, D., Goll, T., Demmel, N., Usenko, V., Stückler, J., and Cremers, D. (2018). The TUM VI Benchmark for Evaluating Visual-Inertial Odometry. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1680–1687. arXiv: 1804.06120.

Usenko, V., Demmel, N., Schubert, D., Stückler, J., and Cremers, D. (2019). Visual-Inertial Mapping with Non-Linear Factor Recovery. *IEEE Robotics and Automation Letters*, 5(2):422–429.

Wyeth, G., Buskey, G., and Roberts, J. (2019). Flight control using an artificial neural network. In *in Proc. Australian Conf. Robotics and Automation*, pages 65–70.