Semantic SuperPoint: Um Descritor Semântico Profundo

Gabriel Soares Gama

Nícolas dos Santos Rosa, Valdir Grassi Junior

Universidade de São Paulo

gabriel_gama@usp.br, nicolas.rosa@usp.br, vgrassi@usp.br

Palavras-chave: Visão Computacional, Odometria Visual, Robótica

Objetivos

Diversos métodos de SLAM se beneficiam do uso de informação semântica. A maioria adiciona informação semântica de alto nível, como detecção de objetos e segmentação semântica, aos métodos fotométricos. Esta pesquisa propõe que informação semântica também pode ser utilizada nos métodos de extração de *features* através do treinamento de um extrator de *features* profundo com segmentação semântica. Assim, seria uma maneira mais robusta em comparação a utilização de apenas informações de alto nível, já que seria aprendida intrinsicamente pelo descritor e não seria dependente da qualidade final.

Métodos e Procedimentos

Nesta pesquisa foi utilizado o modelo SuperPoint (Sp) [1] como base. A arquitetura do modelo Sp é um *encoder* compartilhado seguido de dois *decoders*, um para a extração de KeyPoints e outro para o descritor. Para adicionar informação semântica, foi utilizada a técnica de aprendizado multitarefa, adicionando uma *semantic head* para realizar segmentação semântica. O modelo proposto, Semantic SuperPoint (SSp), pode ser visto na Figura 1.

Para balancear a otimização dos objetivos foram utilizados três métodos. A soma uniforme, balanceamento das funções de perda considerando a incerteza [2] e o método *central dir + tensor* [3] que calcula um gradiente para melhorar todas as tarefas.

Para treinar o modelo, foi selecionado o *dataset* MS-COCO 2017 [4] e o HPatches [5] para a avaliação, de acordo com as métricas de detector e descritor descritas no artigo do Sp [1]. Como foram utilizadas diversas métricas, se fez necessário criar um critério de avaliação. Então, foi considerado o *matching score* (MS) como a principal, já que representa a porcentagem de *inliers.* Logo, avalia tanto o detector quanto o descritor, além de ser a mais relacionada com tarefas de SLAM.

Além disso, os modelos foram utilizados como extratores de *features* no ORB-SLAM2 [6] e foram avaliados de acordo com a métrica *absolute position error* (APE) obtida ao gerar por SLAM as trajetórias no *dataset* do Kitti [7].

Vale notar que, ao adicionar o *decoder* extra, o custo de treinamento aumenta, mas o de inferência é mantido, pois a predição da segmentação semântica não é necessária após o treinamento.

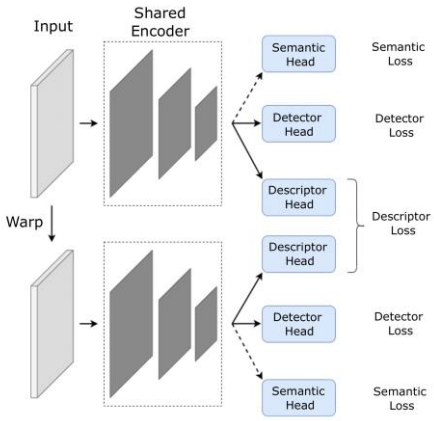

Figura 1: Semantic SuperPoint

Resultados

Na Tabela 1, têm-se os resultados de métodos de extração de *features,* tanto clássicos quanto

profundos. Foram utilizadas as abreviações *uni* para a função de perda uniforme, *unc* para a incerteza e *ct* para o método *central dir + tensor*.

Extrator de features	Estimação Homográfica			Métricas do detector		Métricas do descritor	
	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$	Rep.	MLE	NN mAP.	M.S.
ORB	.150	.395	.538	**.641**	1.157	.735	.266
SIFT	.424	.676	.759	.495	**0.833**	.694	.313
LIFT	.284	.598	.717	.449	1.102	.664	.315
Sp + uni (referência)	.476	.748	**.817**	.599	1.017	**.864**	.519
Sp + unc	.460	.745	.812	.599	1.010	**.864**	.520
Sp + ct	**.493**	.753	.812	.602	1.001	.862	.519
SSp + uni (**nosso**)	.398	.710	.797	.584	1.052	.843	.506
SSp + unc (**nosso**)	.450	.745	.816	.598	1.005	**.864**	**.522**
SSp + ct (**nosso**)	.466	**.762**	.805	.598	0.999	.858	.519

Tabela 1: Resultados no HPatches

Considerando o MS, o modelo SSp teve um resultado melhor, com um pequeno aumento de **+0.2%** em comparação com o Sp + unc. Para a estimação homográfica, teve um melhor resultado que o Sp + unc no caso $\epsilon = 5$, enquanto balanceava as métricas do detector.

Para a avaliação no ORB-SLAM2, as sequências foram executadas 10 vezes e na Tabela 2 as médias dos erros ATE de cada trajetória são exibidas para cada modelo, junto ao valor-p obtido pelo teste Kruskal-Wallis.

Sequência	Sp	SSp	Valor-p
00	6.77	**6.65**	0.71
01	286.71	**209.99**	0.13
02	22.46	**22.31**	0.82
03	**1.32**	1.63	0.00
04	**0.84**	0.90	0.94
05	**5.80**	6.31	0.82
06	11.95	**11.83**	0.41
07	3.39	**2.12**	1.00
08	31.32	**26.70**	0.00
09	35.94	**31.79**	0.00
10	5.51	**4.95**	0.02

Tabela 2: Resultados no Kitti

Analisando os erros em cada trajetória, o modelo SSp obteve um resultado melhor em 8 situações, sendo pelo menos 3 relevantes, enquanto o modelo Sp só possui um resultado melhor que seja relevante.

Portanto, adicionar informação semântica melhora métodos de extração de *features* profundos.

Conclusões

Percebe-se então que adicionar informação semântica em uma arquitetura baseada em um *encoder* compartilhado em combinação com métodos de aprendizado de multitarefa, melhora o desempenho de métodos de extração *features* profundos e, nesse caso, sem adicionar custo computacional na inferência.

Trabalhos futuros podem ajustar a intensidade do processo de *data augmentation* e a complexidade do *dataset* de treinamento para balancear a capacidade de generalização e aprendizado semântico. O modelo também pode ser treinado por outro método de aprendizado que não dependa do processo intensivo de *data augmentation.*

Além disso, caso tenha obtido uma predição semântica de qualidade satisfatória, é possível utilizar a informação de alto nível para aplicar os métodos de SLAM semântico.

Referências Bibliográficas

[1] DeTone, D., Malisiewicz, T., & Rabinovich, A. (2018). Superpoint: Self-supervised interest point detection and description. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 224-236).

[2] Kendall, A., Gal, Y., & Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7482-7491).

[3] Nakamura, A. T. M., Wolf, D. F., & Grassi Jr, V. (2022). Leveraging convergence behavior to balance conflicting tasks in multi-task learning. arXiv preprint arXiv:2204.06698.

[4] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

[5] Balntas, V., Lenc, K., Vedaldi, A., & Mikolajczyk, K. (2017). HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5173-5182).

[6] Raul Mur-Artal and Juan D. Tardos. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-d cameras. IEEE Transactions on Robotics, 33(5):1255–1262, oct 2017.

[7] Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 32(11), 1231-1237.

.