УДК 004.932.7, 004.4, 004.021

ВОССТАНОВЛЕНИЕ КАРТ ГЛУБИН ИЗОБРАЖЕНИЙ, ПОЛУЧЕННЫХ С ЕДИНСТВЕННОЙ ВИДЕОКАМЕРЫ В РЕАЛЬНОМ ВРЕМЕНИ НА ПЛАТФОРМЕ NVIDIA JETSON TX2

К.Ф. Муравьев (*muravev.kf@phystech.edu*)
Московский физико-технический институт
Федеральный исследовательский центр "Информатика и управление" Российской академии наук, Москва

А.В. Боковой (*bokovoy@isa.ru*)
Федеральный исследовательский центр "Информатика и управление" Российской академии наук, Москва
Российский университет дружбы народов, Москва

Аннотация. В статье рассматривается практическое применение искусственных нейронных сетей для восстановления карт глубин изображений, полученных с единственной камеры малой робототехнической системы, в контексте задачи одновременного картирования и локализации по видеопотоку (vision-based Simultaneous Localization and Mapping – vSLAM) в режиме реального времени. Нейронные сети обучены на актуальных коллекциях данных и протестированы на встраиваемом компьютере NVidia Jetson TX2, который благодаря низкому энергопотреблению, малым размерам и особенностям архитектуры позволяет ускорить параллельные вычисления на борту робототехнической системы. Приводятся результаты экспериментов с разными архитектурами нейронных сетей, а также дается описание программных оптимизаций, позволяющих добиться работы алгоритмов восстановления глубины изображений в реальном времени.[1]

Ключевые слова: восстановление глубины, vSLAM, нейронные сети, Nvidia Jetson.

Введение

В последнее время мобильные роботы и беспилотные летательные аппараты все чаще используются в различных коммерческих и бытовых

1 Работа выполнена при финансовой поддержке РНФ (проект 16-11-00048)

целях. Для навигации малых мобильных роботов широко применяются методы одновременного картирования и локализации по видеопотоку (vision-based SLAM, vSLAM) [Blösch M. et al, 2010][Fraundorfer F. et al., 2012][Yang S. et al, 2016]. Методы vSLAM получили широкое распространение, так как они не требуют наличия на борту робототехнического устройства никаких датчиков, кроме единственной видеокамеры, и делают возможной навигацию в помещениях, где использование спутниковой навигации затруднено. Классические методы vSLAM, основанные на извлечении структуры из движения (Structure from Motion, SfM), имеют существенные недостатки, такие, как потеря структуры при поворотах робота на месте и невозможность восстановления точного масштаба карты (все расстояния на построенной карте - относительные) [Davison A. J. et al., 2007][Klein G. et al., 2007]. Восстановление карты глубин по видеопотоку позволяет устранить данные недостатки, сведя задачу vSLAM к задаче одновременного картирования и локализации с использованием видеокамеры и датчиков глубины, для которой разработаны эффективные методы решения [Enders F. Et al., 2012][Kerl C. et al., 2013].

Классические методы восстановления глубины по видеоданным основаны на использовании оптического потока [Newcombe R. A. et al, 2010]. Как правило, производительность таких методов недостаточна для обработки видеопотока в реальном времени с помощью бортовых вычислителей робототехнических систем. В настоящее время помимо классических алгоритмов восстановления глубины применяются также нейросетевые методы, обрабатывающие отдельно каждый кадр видеопоследовательности [Laina I. et al, 2016][Garg R. et al., 2016][Kuznietsov V. et al., 2017]. Подобные методы позволяют достичь приемлемого качества восстановления глубины и производительности, достаточной для обработки видеопотока в реальном времени, но требуют наличия мощного графического ускорителя, что затрудняет их применение для навигации беспилотных транспортных средств малого размера. В данной работе рассматривается применение нейросетей для восстановления глубины на одноплатном компьютере NVIDIA Jetson TX2, который оснащен графическим ускорителем и при этом достаточно компактен и энергоэффективен для использования на борту малых робототехнических систем. Описываются программные оптимизации, позволяющие сократить время обработки одного изображения на Jetson TX2 до 80 мс в разрешении 320x240 и 150 мс в разрешении 640x480.

1 Описание нейросети

Для восстановления глубины используются нейросети с полносверточной архитектурой, не содержащие полносвязных слоев (fully-convolutional networks [Long J. et al., 2015][Dai J. et al., 2016]). Предсказание карты глубины сверточным слоем вместо полносвязного позволяет значительно уменьшить число параметров нейросети и, как следствие, сократить объем занимаемой памяти и ускорить процесс предсказания карты глубины.

Используемые в работе нейросети принимают на вход цветное трехканальное изображение и выдают предсказанную карту глубины. Они состоят из двух частей: свертки и развертки. Сверточная часть (энкодер) содержит серию слоев свертки (convolution) и субдискретизации (pooling) и последовательно уменьшает размерность изображения, преобразуя его в набор высокоуровневых признаков. В данной работе в качестве сверточной части используется архитектура ResNet-50 [He K. et al., 2016], предобученная на коллекции изображений ImageNet. Разверточная часть (декодер) представляет собой серию операций транспонированной свертки (Deconvolution, [Dumoluin V. et al., 2016][Zeiler M. D. et al., 2010]) и активаций. Декодер преобразует набор высокоуровневых признаков, предсказанных энкодером, в карту глубины. Карта глубины представляет собой двумерный массив той же ширины и высоты, что и входное изображение, в каждой ячейке которого находится глубина соответствующего пикселя входного изображения. Глубина задается положительным числом с плавающей точкой.

Для экспериментов на NVIDIA Jetson использовались две нейросетевые архитектуры. Первая нейросеть обрабатывает изображения размером 640x480 пикселей. Ее архитектура представлена на рисунке 1. Энкодером является предобученная сеть ResNet-50 без полносвязных слоев. Энкодер преобразует входное изображение в карту признаков размерности 20x15x2048. Разверточная часть начинается со свертки с ядром размера 1x1 и 1024 фильтрами. Затем следуют 5 блоков развертки, в которых последовательно применяются операции нормализации, транспонированной свертки (Deconvolution) с шагом 2 и ядрами размера 5x5, активации (ReLU). Количество фильтров в Deconvolution-слоях блоков развертки уменьшается последовательно с 512 до 32. После блоков развертки следует сверточный слой с одним фильтром, выводящий предсказанную карту глубины. Вторая нейросеть обрабатывает изображения размером 320x240 пикселей. Энкодером является предобученная сеть ResNet-50 без полносвязных слоев и последнего блока сверток. Энкодер преобразует входное изображение в карту признаков размерности 20x15x1024. Разверточная часть содержит свертку с ядром

размера 1x1 и 1024 фильтрами и 4 блока развертки, идентичные разверточным блокам первой нейросети. После блоков развертки следует сверточный слой с одним фильтром, выводящий предсказанную карту глубины.

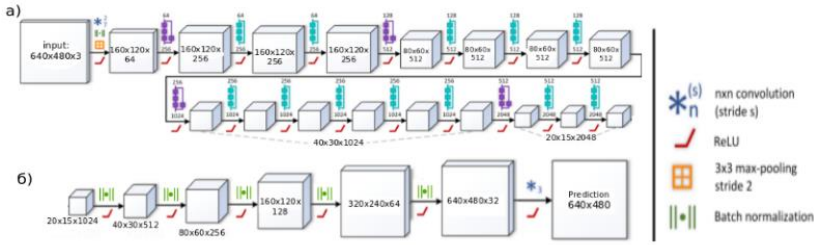

Рисунок 1. Архитектура нейросети для разрешения 640x480. а) архитектура энкодера; б) архитектура декодера

Обучение нейросетей производилось на коллекции NYU Depth Dataset, содержащей около 1100 видеосцен и более 400 тысяч кадров, снятых в помещениях различного типа, с размеченными с помощью сенсора LIDAR картами глубины. Глубина изображений ограничивалась 10 метрами. Функцией потерь являлась среднеквадратичная ошибка между истинной и предсказанной глубиной. Валидация проводилась на двух выборках: первая содержала кадры из сцен, отсутствующих в обучающей выборке, но снятых в помещениях схожего с обучающей выборкой типа, а вторая — кадры из сцен, снятых в помещениях типа NYU office (офисы Нью-Йоркского университета). Тип помещений NYU Office в обучающей выборке отсутствовал.

Обучение проводилось в четыре этапа:

1. Обучаются только веса декодера, веса энкодера не меняются
2. Обучаются веса декодера и последнего ResNet-блока
3. Обучаются веса декодера и двух последних ResNet-блоков
4. Обучаются все веса нейросети

Такой подход позволяет ускорить процесс адаптации нейросети с предобученными слоями к новой задаче и повысить качество работы [Yosinsky J. et al., 2014].

По итогам обучения, среднеквадратичное отклонение (RMSE) предсказаний на изображениях размера 640x480 составило 0.78м на первой валидационной выборке и 0.88м на второй, на изображениях размера 320x240 — 0.87м на первой валидационной выборке и 0.92м на второй. Графики обучения представлены на рисунке 2.

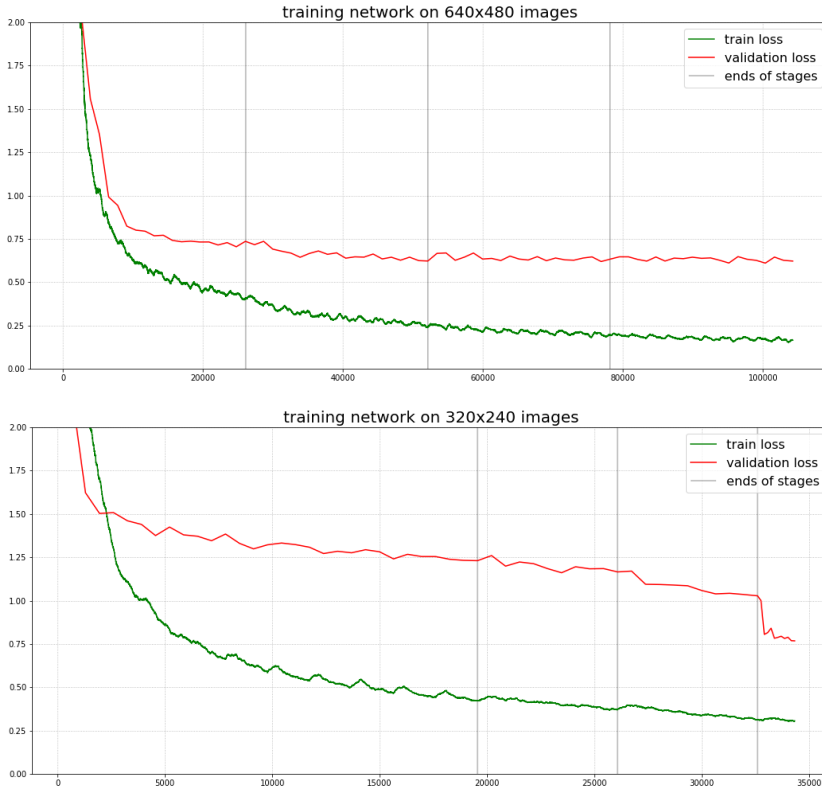

Рисунок 2. Графики обучения нейросетей для восстановления глубины. Сверху график обучения на изображениях разрешения 640x480, снизу — 320x240. Вертикальными линиями показаны границы этапов обучения. По оси абсцисс — число шагов обучения, по оси ординат — средний квадрат ошибки (MSE). Для валидации использовалась первая валидационная выборка

Ошибка на валидационной выборке намного превосходит ошибку на обучающей выборке почти на всем протяжении обучения. Такое различие объясняется отсутствием в валидационной выборке кадров из помещений, присутствующих в обучающей выборке и большими различиями в интерьере и в условиях съемки разных помещений.

2 Особенности реализации

Архитектура нейросети была построена с использованием библиотеки глубинного обучения TensorFlow и ее расширения Keras. Код, реализующий архитектуру, был написан на языке Python. Для борьбы с переобучением после каждого блока развертки был включен слой дропаута (Dropout) с коэффициентом 0.5.

Обучение нейросетей проводилось на гибридном высокопроизводительном вычислительном кластере ФИЦ ИУ РАН [ГВВК, 2018]. Так как обучающая коллекция данных полностью не помещалась в оперативную память, то для ускорения процесса обучения она была разбита на 10 частей, которые последовательно загружались в оперативную память и разбивались на батчи для подачи на вход нейросети.

Тестирование алгоритмов восстановления глубины проводилось на встраиваемом компьютере NVIDIA Jetson TX2 с операционной системой Ubuntu 16.04 и пакетом инструментов JetPack 3.0 [Buonaiuto N. et al., 2017][Hadidi R. et al., 2018]. Данный компьютер оснащен 256-ядерной видеокартой с архитектурой PASCAL и 6-ядерным центральным процессором CPU Complex ARMv8 и имеет 8 ГБ оперативной памяти, разделяемой между GPU и CPU. При этом он имеет размеры 50x87 мм, а его энергопотребление составляет 10-13 Вт на максимальной тактовой частоте, что позволяет встраивать его в малогабаритные робототехнические системы, в том числе в малые БПЛА. Высокая скорость работы нейронных сетей на NVIDIA Jetson достигается за счет поддержки библиотек CuDNN и TensorRT, а также аппаратной поддержки вычислений с половинной точностью (fp16).

Для эффективной работы нейросети на встраиваемом компьютере она была переведена в формат TensorRT engine с поддержкой вычислений с половинной точностью. Код, осуществляющий обработку полученных с камеры изображений и визуализацию предсказанной карты глубины в реальном времени, был реализован на языке C++ с использованием технологии CUDA. Изображение с камеры захватывалось с помощью библиотеки Gstreamer и записывались в видеопамять с помощью CUDA. Затем изображение переводилось в формат RGBA и нормализовалось для подачи на вход нейросети. Конвертация в RGBA и нормализация были распараллелены с помощью CUDA. Далее нормализованное изображение подавалось на вход нейросети для восстановления карты глубины. Восстановленная нейросетью карта глубины и исходное изображение отрисовывались на экране с помощью библиотеки OpenGL.

Исходный код восстановления глубины по изображениям в реальном времени доступен в Github-репозитории[2]. Нейросети в формате TensorRT engine доступны в облачном хранилище Яндекс.Диск[3].

3 Результаты экспериментов

Эксперимент проводился на платформе NVIDIA Jetson TX2. Изображения, полученные с установленной на платформе видеокамеры, подавались на вход нейросети. Вычисленная нейросетью карта глубины вместе с исходным изображением визуализировались с помощью библиотеки OpenGL. Примеры визуализации представлены на рисунке 3.

Рисунок 3. Визуализация восстановления глубины в реальном времени на NVIDIA Jetson по изображениям в разрешении 640x480. а) Трехканальное изображение с камеры, установленной на платформе; б) Предсказанная карта глубины в формате grayscale; в) Предсказанная карта глубины, визуализированная в цветовой гамме

Средняя частота восстановления глубины при разрешении 320x240 составила 13 Гц. Средняя частота восстановления глубины при разрешении 640x480 составила 6 Гц.

Для оценки производительности библиотеки TensorRT на NVIDIA Jetson было проведено сравнение работы нейросети для восстановления глубины в различных фреймворках на различных устройствах по скорости и объему занимаемой памяти. Результаты сравнения представлены в таблице 1. Для сравнения с NVIDIA Jetson TX2 использовался ноутбук, оснащенный 8-ядерным процессором Intel Core i7-8550 и видеокартой NVIDIA GeForce MX150 (384 CUDA-ядра, архитектура Pascal), имеющей энергопотребление 30 Вт на максимальной тактовой частоте.

В ходе сравнения выяснилось, что перевод нейросети из формата Keras model в формат TensorRT engine позволил ускорить ее работу более чем в 4 раза, а также сократить в 3 раза объем потребляемой оперативной памяти и в 5-8 раз — объем потребляемой памяти GPU. Также выяснилось, что NVIDIA Jetson по производительности в задаче восстановления глубины

2 https://github.com/CnnDepth/jetson-inference/tree/master/fcrn-camera
3 https://yadi.sk/d/fgSHHUpgw_aw4w

почти не уступает ноутбуку с видеокартой MX150, имеющей вдвое большую тепловую мощность, чем видеокарта NVIDIA Tegra X2, установленная на Jetson.

Таблица 1. Сравнение производительности нейросети в различных фреймворках на различных устройствах

Фреймворк	Устройство	Разрешение	Время обработки одного кадра, мс	Объем занимаемой оперативной памяти, ГБ	Объем занимаемой памяти GPU, ГБ
TensorRT	Jetson TX2	640x480	152	0.62	0.31
Keras	Jetson TX2	640x480	630	1.9	2.65
TensorRT	Ноутбук	640x480	143	0.77	0.51
TensorRT	Jetson TX2	320x240	80	0.62	0.28
Keras	Jetson TX2	320x240	340	1.8	2.6
TensorRT	Ноутбук	320x240	73	0.77	0.37

Выводы

В результате работы были созданы и обучены нейронные сети для восстановления карты глубины по изображениям с единственной видеокамеры в реальном времени. Качество восстановления глубины было протестировано на коллекции NYU Depth Dataset, содержащей изображения из помещений различного типа с глубинами до 10 метров. Среднеквадратичная ошибка на тестовой выборке данной коллекции составила 0.87м при разрешении 640x480 и 0.92м при разрешении 320x240. Для достижения высокой скорости работы нейросетей на встраиваемом компьютере они были переведены в формат TensorRT engine с поддержкой вычислений половинной точности. При тестировании на платформе NVIDIA Jetson TX2 частота построения карты глубины составила 13 Гц при разрешении 320x240 и 6 Гц при разрешении 640x480. Такая скорость работы в совокупности с приемлемым качеством восстановления глубины делает возможным применение данных нейросетей в методах одновременного картирования и локализации, в том числе для навигации малых беспилотных летательных аппаратов. В дальнейшем планируется использование восстановленных с помощью подобных нейросетей карт глубины в методах SLAM, основанных на данных с единственной видеокамеры, с помощью фреймворка ROS [ROS, 2009] и алгоритма RTABMap [Labble M. Et al. 2011].

Список литературы

[Blösch M. et al. 2010] Blösch, M., Weiss, S., Scaramuzza, D., & Siegwart, R. Vision based MAV navigation in unknown and unstructured environments //Robotics and automation (ICRA), 2010 IEEE international conference on. – IEEE, 2010. – С. 21-28

[Fraundorfer F. et al., 2012] Fraundorfer, F., Heng, L., Honegger, D., Lee, G. H., Meier, L., Tanskanen, P., & Pollefeys, M. Vision-based autonomous mapping and exploration using a quadrotor MAV //2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. – IEEE, 2012. – С. 4557-4564.

[Yang S. et al, 2016] Yang S., Scherer S. A., Zell A. Robust Onboard Visual SLAM for Autonomous MAVs //Intelligent Autonomous Systems 13. – Springer International Publishing, 2016. – С. 361-373

[Laina I. et al., 2016] Laina I, Rupprecht C., Belagiannis V., Tombari F., Navab N. Deeper depth prediction with fully convolutional residual networks //3D Vision (3DV), 2016 Fourth International Conference on. – IEEE, 2016. – С. 239-248.

[Garg R. et al., 2016] Garg R., Kumar V., Carneiro G., Reid I. Unsupervised cnn for single view depth estimation: Geometry to the rescue //European Conference on Computer Vision. – Springer, Cham, 2016. – С. 740-756.

[Kuznietsov Y. et al., 2017] Kuznietsov Y., Stückler J., Leibe B. Semi-supervised deep learning for monocular depth map prediction //Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. – 2017. – С. 6647-6655.

[Davison A. J. et al., 2007] Davison A. J. et al. MonoSLAM: Real-time single camera SLAM //IEEE Transactions on Pattern Analysis & Machine Intelligence. – 2007. – №. 6. – С. 1052-1067.

[Klein G. et al., 2007] Klein G., Murray D. Parallel tracking and mapping for small AR workspaces //Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. – IEEE Computer Society, 2007. – С. 1-10.

[Enders F. et al, 2012] Endres F. et al. An evaluation of the RGB-D SLAM system //Icra. – 2012. – С. 1691-1696.

[Kerl C. et al, 2013] Kerl C., Sturm J., Cremers D. Dense visual SLAM for RGB-D cameras //2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. – IEEE, 2013. – С. 2100-2106.

[Long J. et al, 2015] Long J., Shelhamer E., Darrell T. Fully convolutional networks for semantic segmentation //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – С. 3431-3440.

[Dai J. et al, 2016] Dai J. et al. R-fcn: Object detection via region-based fully convolutional networks //Advances in neural information processing systems. – 2016. – С. 379-387.

[Newcombe R. A. et al., 2010] Newcombe R. A., Davison A. J. Live dense reconstruction with a single moving camera //2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. – IEEE, 2010. – С. 1498-1505.

[He K. et al., 2016] He K. et al. Deep residual learning for image recognition //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2016. – С. 770-778.

[Dumoluin V. et al., 2016] Dumoulin V., Visin F. A guide to convolution arithmetic for deep learning //arXiv preprint arXiv:1603.07285. – 2016.

[Zeiler M. D. et al, 2010] Zeiler M. D. et al. Deconvolutional networks. – 2010.

[Yosinsky J. et al., 2014] Yosinski J. et al. How transferable are features in deep neural networks? //Advances in neural information processing systems. – 2014. – С. 3320-3328.

[Buonaiuto J. et al., 2017] Buonaiuto N. et al. Satellite identification Imaging for small satellites using NVIDIA. – 2017.

[Hadidi R. et al., 2018] Hadidi R. et al. Real-time image recognition using collaborative IoT devices //Proceedings of the 1st on Reproducible Quality-Efficient Systems Tournament on Co-designing Pareto-efficient Deep Learning. – ACM, 2018. – С. 4.

[ГВВК, 2018] Федеральный исследовательский центр Информатика и управление РАН [Электронный ресурс]: сайт. – Москва: ФИЦ ИУ РАН. – URL: http://hhpcc.frccsc.ru (дата обращения: 12.09.2018)

[ROS, 2009] Quigley M. et al. ROS: an open-source Robot Operating System //ICRA workshop on open source software. – 2009. – Т. 3. – №. 3.2. – С. 5.

[Labble M. Et al. 2011] Labbé M., Michaud F. Memory management for real-time appearance-based loop closure detection //2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. – IEEE, 2011. – С. 1271-1276.