


УДК 004.42:004.896

 10.25209/2079-3316-2023-14-2-3-26

Повышение качества видеопотока от системы технического зрения беспилотного летательного аппарата

Виталий Петрович **Фраленко**

Институт программных систем им. А. К. Айламазяна РАН, Вельсково, Россия

Аннотация. В исследовании приведены результаты работы над программно-аппаратным комплексом для повышения качества видеоданных, получаемых от беспилотных летательных аппаратов. Рассмотрены задачи деконволюции отдельных кадров (удаление смазов) и стабилизации видеопотока с использованием методов машинного обучения и искусственного интеллекта. Представлены аналитические и практические результаты, позволившие подобрать решения для обработки данных от БПЛА в режиме реального времени.

Ключевые слова и фразы: БПЛА, деконволюция, стабилизация, режим реального времени, экспериментальные данные

Благодарности: Исследование выполнено за счет гранта Российского научного фонда № 21-71-10056

Для цитирования: Фраленко В.П. *Повышение качества видеопотока от системы технического зрения беспилотного летательного аппарата* // Программные системы: теория и приложения. 2023. Т. 14. № 2(57). С. 3–26.
https://psta.psir.ru/read/psta2023_2_3-26.pdf

Введение

В настоящее время особое внимание уделяется анализу и интерпретации видеопотока от беспилотных летательных аппаратов (БПЛА), которые должны обеспечивать решение задач выделения и распознавания целевых объектов, позиционирования и слежения, выполняемых с борта БПЛА, в том числе как автономно, так и с Земли, для чего жизненно необходимо иметь четкие изображения без смазов, а кадры в обрабатываемых видеопоследовательностях должны сменять друг друга без резких смещений камеры. Несмотря на удачные аэродинамические и компоновочные решения, отечественные разработки по оснащению БПЛА пока проигрывают зарубежным в части качества бортовых систем технического зрения. Для обеспечения необходимых показателей требуется создание математического, алгоритмического и программного обеспечения, способного выполнять в режиме реального времени основные функции предобработки и улучшения поступающих от БПЛА кадров.

Для проведения экспериментов с имеющимся и доработанным программным обеспечением применялся процессор общего назначения Intel Core i3 8300 и графический ускоритель вычислений Nvidia GTX 1080 Ti с 11 ГБ видеопамяти. В качестве экспериментальных данных использовались снимки, полученные с мобильной камеры (набор данных GoPro¹), максимально близкие к тем, что получаются от реальных систем технического зрения БПЛА.

1. Деконволюция потока данных от БПЛА

На реальных изображениях от БПЛА присутствуют явные смазы, получившиеся из-за резкого смещения камеры во время получения того или иного кадра. При этом это смещение может быть нелинейным ввиду того, что движение БПЛА зависит от управляющих воздействий оператора, ветровой нагрузки, точности работы систем ориентирования на местности, в том числе Glonass и GPS. Все эти факторы могут накладываться на БПЛА одновременно, что приводит к тому, что положение камеры в пространстве становится нестабильным. Кроме того, объекты в процессе получения одного и того же кадра тоже могут двигаться, например, это автомобили, велосипеды, пешеходы, животные. То есть камера может некачественно фиксировать объект в кадре из-за 1) быстрого движения камеры или самого объекта; 2) неоптимальных настроек камеры (например,

¹Набор данных GoPro для удаления размытия (*Papers with Code*^{url}) состоит из 3214 размытых изображений размером 1280 × 720, которые разделены на 2103 обучающих изображения и 1111 тестовых изображений. Набор данных состоит из пар реалистичного размытого изображения и соответствующего наземного изображения, полученного с помощью высокоскоростной камеры.

поставлена слишком большая выдержка, матрица очень медленная или перегрета). Деконволюция (от англ. «deconvolution») – способ компенсации таких факторов, приводящих к тому, что кадры видеопотока становятся смазанными. Предлагается использовать для деконволюции методы математической статистики и искусственные нейронные сети (ИНС), обучаемые с помощью общедоступных наборов данных.

Примеры обрабатываемых изображений приведены на рисунках 1– 2.



Рисунок 1. Пример 1 исходного изображения

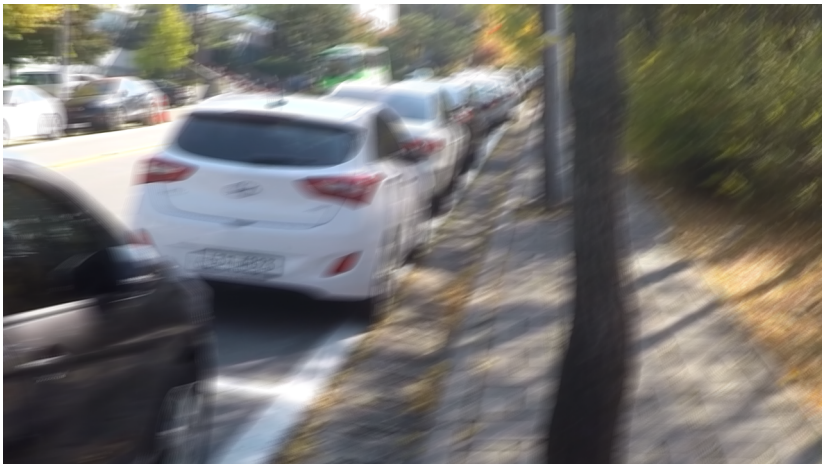
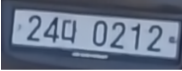

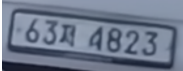
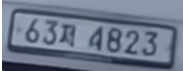
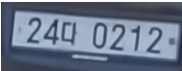

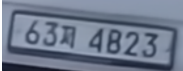
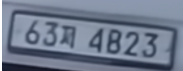
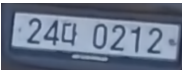

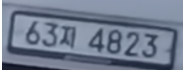
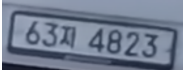
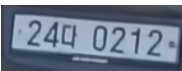

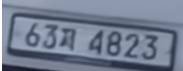
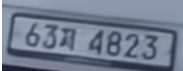
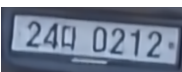

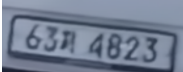
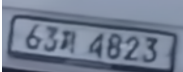
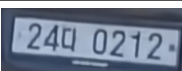

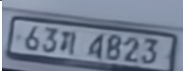
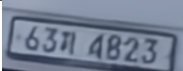
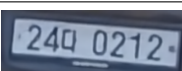

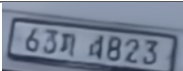
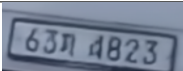
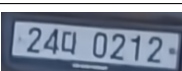

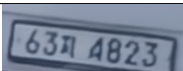
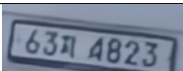


Рисунок 2. Пример 2 исходного изображения

В таблице 1 перечислены выявленные в результате аналитического исследования алгоритмические и программные решения, предназначенные для автоматизированного устранения смазов. Приводятся сведения по сред-

ТАБЛИЦА 1. Результаты деконволюции

Метод	Время, с.	Фрагмент 1		Фрагмент 2	
		PSNR	SSIM	PSNR	SSIM
<i>Test-time Local Converter (TLC)</i> ^{URL} [1]	13.5	 33.012	 0.939	 31.186	 0.929
<i>Multi-Axis MLP</i> ^{URL} [2]	19.5	 30.556	 0.917	 31.748	 0.934
<i>EFNet</i> ^{URL} [3]	1.10	 35.184	 0.948	 35.695	 0.962
<i>Learning degradation</i> ^{URL} [4]	3.60	 32.549	 0.932	 31.739	 0.934
<i>Deep Generalized Unfolding (DGU)</i> ^{URL} [5]	5.80	 22.582	 0.735	 29.923	 0.903
<i>NAFNet width32</i> ^{URL} [6]	5.00	 31.789	 0.918	 30.000	 0.916
<i>Stripformer</i> ^{URL} [7]	0.03	 28.060	 0.876	 30.222	 0.916
<i>Uformer</i> ^{URL} [8]	9.40	 32.296	 0.928	 31.477	 0.929

нему времени обработки отдельных кадров с помощью соответствующего

решения и результаты обработки в виде фрагментов целевых областей. Отметим, что все актуальные решения реализованы в виде программного кода, использующего для вычислений графические процессоры.

В рамках настоящей работы решалась подзадача восстановления областей видеокadra с присутствующим номером автомобиля. БПЛА активно используются в мире для автоматизированного обследования и фиксации автомобильных номеров, поэтому подобная подзадача является актуальной. В данном случае используются изображения с автомобильными номерами Республики Корея, третий символ в номере автомобиля это буквенный префикс, а не цифра, как может показаться. Для численной оценки были получены соответствующие отношения сигнала к шуму (Peak Signal-to-Noise Ratio, PSNR) и значения величин структурных подобий (Structure SIMilarity, SSIM), использовалась программная библиотека *skimage.metrics*^[URL]. Выполнено сравнение полученных изображений с изображениями без смазов (подгруппа снимков «sharp» из набора данных GoPro). Для поврежденных смазами оригинальных изображений автомобильных номеров (см. рисунки 1– 2) PSNR и SSIM соответственно равны следующим величинам: 18.42 и 0.57 (рисунок 1); 18.79 и 0.56 (рисунок 2). При полном совпадении изображений значение SSIM = 1, для абсолютно разных изображений SSIM = 0.

Экспериментальные исследования, проведенные с использованием этих и других изображений из набора данных GoPro, показали, что нейронная сеть Stripformer выполняет обработку в режиме реального времени, при этом получаются достаточно качественные выходные изображения. Stripformer имеет архитектуру, основанную на идее трансформеров. «Визуальный трансформер» состоит из токенизатора, трансформера и проектора, более подробное описание принципов его работы можно найти в исследовании [9]. Stripformer строит внутренние и внешние ленточные токены (strip tokens) в процессе перерасчета информативных признаков изображения в горизонтальных и вертикальных направлениях, что позволяет выявлять зашумленные фрагменты с различной ориентацией (направлением распространения) смаза. Нейронная сеть совмещает внешние и внутренние ленточные слои внимания (attention layers), чтобы определить степень и характер размытия. В дополнение к обнаружению

специфических для области размытых паттернов различных ориентаций и размеров, Stripformer успешно работает даже без данных о динамике сцены и не требует огромного массива данных для обучения. Следует отметить высокое качество работы нейронной сети EFNet.

Пример работы Stripformer показан на рисунке 3. Обученная оригинальная модель требует для своей работы наличия видеокарты с не менее чем 8 ГБ видеопамяти. В рамках настоящего исследования выполнена небольшая модификация программного кода Stripformer, связанная с особенностями запуска модели на заявленном процессоре общего назначения с использованием библиотеки *PyTorch*^{URL}, однако она работает неприемлемо долго, порядка 90 секунд на каждый кадр, что в 3000 раз дольше, чем на используемой видеокarte.



Рисунок 3. Результат удаления смазов на снимке с помощью Stripformer

Таким образом, предложенное решение позволяет в режиме реального времени получать данные от БПЛА, очищенные от смазов.

2. Стабилизация видеопотока

При стабилизации видеопотока кадр обрезается со всех сторон, а оставшиеся части кадра используются для компенсации движения

и формирования итогового изображения. Эта компенсация обычно выполняется с помощью ИНС и статистических методов на основе предыдущих кадров, для чего поддерживается буфер определенного размера.

В качестве тестовых в настоящей работе использовались выборки по 100 кадров, восстановленные с помощью Stripformer и подаваемые со скоростью 24 кадра в секунду.

Далее перечислены выявленные в результате исследования алгоритмические и программные решения, предназначенные для автоматизированной стабилизации видеопотока от БПЛА. Приводится среднее время обработки всего видеофайла. Каждое решение снабжено комментарием, отражающим приобретенный опыт его использования, некоторые реализованы с поддержкой графических ускорителей вычислений.

- **VidGear^{URL} [10]**. 3.91 секунды. Рамки кадров обрезаются с помощью отступа от границ и зума. Требуется инициализация стабилизатора, для чего используется объем входных данных порядка 24 кадров. Время подготовки стабилизатора к работе составляет порядка 0.70 с., то есть оно выполняется в фоновом режиме по мере поступления кадров на 30% быстрее того, как кадры поступают на обработку.
- **FuSta: Hybrid Neural Fusion^{URL} [11]**. 1200 секунд. Используется графический ускоритель. На первом шаге запускается скрипт с генерацией данных оптического потока (*python main.py* с параметрами), который попутно выполняет стабилизацию видео, но без отрезания краев, зума и пр. Полная обработка 100 кадров заняла 5 минут. Итоговый стабилизированный файл получился с разрешением 960x576 пикселей. На втором шаге запускается основная программа, использующая собранные данные оптического потока, *python run_FuSta.py* с опцией *-temporal_width 18* вместо исходной *-temporal_width 41*. Иначе графическому ускорителю не хватает имеющихся 11 ГБ видеопамати. Оригинальная библиотека содержала уже неподдерживаемые коды нейронной сети *flownet2*, поэтому пришлось переписать их. Результат – набор изображений в исходном разрешении. Обработка выполняется за 15 минут. Видео, получаемое из этих итоговых кадров плавное, однако, в отличие от результатов, полученных с помощью скрипта *python main.py*, на границах объектов «автомобиль» и «дерево» имеются артефакты (см. рисунок 4).



Рисунок 4. Артефакты, которые получены при работе с FuSta

- *MeshFlow: Minimum Latency Online Video Stabilization*^{URL} [12]. 1260 секунд (см. рисунок 5). Проведены эксперименты с двумя профилями



Рисунок 5. Оригинальное изображение до стабилизации и артефакт с наклоном автомобиля при использовании профиля Constant High метода MeshFlow

настроек. Профиль Original: отличное качество без рамок, кадр

обрезается автоматически. Профиль Constant High: обнаружены артефакты в виде неестественно наклоненных объектов, например, автомобилей.

- *Video Stabilization with L1 optimal camera paths*^{URL} [13]. 0.50 секунды. Результирующее видео формируется как будто волнами. Камера то наезжает на объекты в кадре, то возвращается обратно, что выглядит неестественным.
- *Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths*^{URL} [13] (используется дополнительная предобработка изображений). 120 секунд. Итоговое изображение существенно обрезается при применении достаточно высоких уровней стабилизации. Например, при пороге в 40 пикселей изображение 1280x720 уменьшается до 1180x620. Использование меньшего порога дает недостаточный уровень стабилизации видео. Обработка разбита на несколько этапов: 2 минуты на предобработку, затем быстрая стабилизация с использованием полученных данных.
- *Video stabilization using homography transform*^{URL} [14]. 4.18 секунды. Итоговое изображение существенно обрезается, особенно при высоких коэффициентах стабилизации. Наилучшие результаты получены на realtime-стабилизации, при значении параметра $amount=20$.
- *DUT: Learning Video Stabilization*^{URL} [15]. 32 секунды. Используется графический ускоритель. Метод опирается на несколько предобученных нейронных сетей. В основе – самообучение на нестабильных видеоданных. Изображение автомобиля на итоговом видео немного отклоняется влево-вправо (не так существенно, как на рисунке 5), однако в целом получившееся видео можно назвать качественным.
- *Deep Iterative Frame Interpolation for Full-frame Video Stabilization*^{URL} [16]. 100 секунд. Используется графический ускоритель. Обработка выполняется итеративно, число итераций регулируется параметром n_iter , по умолчанию $n_iter = 3$. Итоговый видеопоток имеет неустойчивый характер, камера «прыгает» от кадра к кадру, общее качество создаваемого видеопотока низкое.
- *PWStableNet*^{URL} [17]. 8 секунд. Используется графический ускоритель. Результирующее видео с подергиваниями.





Наилучшие результаты в задаче стабилизации получены с использованием библиотеки VidGear, в которой накапливается массив особых точек на заданном числе кадров с дальнейшим его обновлением в режиме реального времени. Накопленные данные используются для оценки

движения камеры БПЛА с дальнейшей компенсацией ее нестабильности. Реализация соответствует подходу, используемому в библиотеке *OpenCV*^{URL}. При этом в целом обработка происходит в конвейерном режиме.








Заключение

Проведенные исследования позволили получить решение двух важнейших задач улучшения видеопотока от БПЛА: удаление смазов и стабилизация видео. Нейронная сеть Stripformer покадрово удаляет смазы, затем обработанные данные сразу попадают в буфер модуля стабилизации на базе библиотеки VidGear, с помощью которого осуществляется стабилизация видеопотока. Итоговое решение позволяет осуществлять полный цикл обработки видеопотока с задержкой не более чем на 0.03 секунды за счет применения конвейерного метода обработки данных. Обработка выполняется в режиме реального времени.

Список литературы

- [1] Xiaojie Chu, Liangyu Chen, Chengpeng Chen, Xin Lu *Improving image restoration by revisiting global information aggregation*, ECCV 2022: Computer Vision – ECCV 2022 (October 23–27, 2022, Tel Aviv, Israel), Lecture Notes in Computer Science.– vol. **13678**, Cham: Springer.– 2022.– Pp. 330–351 .  [arXiv:2112.04491](#) ↑6
- [2] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, Yinxiao Li *MAXIM: multi-axis MLP for image processing // 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.– 2022.– 34 pp.  [arXiv:2201.02973](#) ↑6
- [3] Lei Sun, Christos Sakaridis, Jingyun Liang, Qi Jiang, Kailun Yang, Peng Sun, Yaozu Ye, Kaiwei Wang, Luc Van Gool *Event-based fusion for motion deblurring with cross-modal attention*, ECCV 2022: Computer Vision – ECCV 2022 (October 23–27, 2022, Tel Aviv, Israel), Lecture Notes in Computer Science.– vol. **13678**, Cham: Springer.– Pp. 412–428; 2023.– 17 pp.  [arXiv:2112.00167](#) ↑6
- [4] Dasong Li, Yi Zhang, Ka Chun Cheung, Xiaogang Wang, Hongwei Qin, Hongsheng Li *Learning degradation representations for image deblurring // Computer Vision – ECCV 2022*, ECCV 2022, Lecture Notes in Computer Science, eds. Avidan S., Brostow G. Cissé M. Farinella G.M. Hassner T.; 2022.– 18 pp.  [arXiv:2208.05244](#) ↑6

- [5] Chong Mou, Qian Wang, Jian Zhang *Deep generalized unfolding networks for image restoration* // *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.– 2022.– Pp. 17399–17410.– 12 pp. arXiv:2204.13348 doi ↑6
- [6] Chen L., Chu X., Zhang X., Sun J. *Simple baselines for image restoration* // *Computer Vision – ECCV 2022*, ECCV 2022, Lecture Notes in Computer Science, eds. Avidan S., Brostow G., Cissé M., Farinella G.M., Hassner T..– 2022; 21 pp. doi arXiv:2204.04676 ↑6
- [7] Fu-Jen Tsai, Yan-Tsung Peng, Yen-Yu Lin, Chung-Chi Tsai, Chia-Wen Lin *Stripformer: strip transformer for fast image deblurring*. // *Computer Vision – ECCV 2022*, ECCV 2022, Lecture Notes in Computer Science.– vol. **13679**, eds. Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T..– 2022; 17 pp. doi arXiv:2204.04627 ↑6
- [8] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, Houqiang Li *Uformer: a general U-shaped transformer for image restoration*, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).– 2022.– 17 pp. doi arXiv:2106.03106 ↑6
- [9] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, Peter Vajda *Visual transformers: token-based image representation and processing for computer vision*.– 2020.– 12 pp. arXiv:2006.03677 doi ↑7
- [10] Thakur A., Papakipos Z., Clauss C., Hollinger C., Andolina I. M., Boivin V., enarcheahn, freol35241, Lowe B., Schoentgen M., Bouckenooghe R. *abhiTronix/vidgear: VidGear v0.2.6*.– 2022. doi URL ↑9
- [11] Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, Jia-Bin Huang *Hybrid neural fusion for full-frame video stabilization*, Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).– 2021.– Pp. 2299–2308. doi arXiv:2102.06205 ↑9
- [12] Shuaicheng Liu, Ping Tan, Lu Yuan, Jian Sun, Bing Zeng *MeshFlow: minimum latency online video stabilization*, ECCV 2016: Computer Vision – ECCV 2016 (October 11-14, 2016, Amsterdam, The Netherlands), Lecture Notes in Computer Science.– vol. **9910**.– 2016.– Pp. 800–815. doi URL ↑10
- [13] Grundmann M., Kwatra V., Essa I. *Auto-directed video stabilization with robust L1 optimal camera paths* // *CVPR '11: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition* (20–25 June 2011, Colorado Springs, CO, USA).– IEEE Computer Society.– 2011.– ISBN 978-1-4577-0394-2.– Pp. 225–232. doi ↑11

- [14] Grundmann M., Kwatra V., Essa I. *Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths* // *CVPR '11: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition* (20–25 June 2011, Colorado Springs, CO, USA).– IEEE Computer Society.– 2011.– Pp. 225–232.  [↑11](#)
- [15] Yufei Xu, Jing Zhang, Stephen J. Maybank, Dacheng Tao *DUT: learning video stabilization by simply watching unstable videos* // *IEEE Transactions on Image Processing*.– 2022.– Vol. **31**.– Pp. 4306–4320 .  [arXiv:2011.14574](#)  [↑11](#)
- [16] Jinsoo Choi, In So Kweon *Deep iterative frame interpolation for full-frame video stabilization* // *ACM Transactions on Graphics*.– 2019.– Vol. **39**.– No. 1.– id. 4.– 9 pp.  [arXiv:1909.02641](#)  [↑11](#)
- [17] Wang M., Yang G. -Y., Lin J. -K., Zhang S. -H., Shamir A., Lu S. -P., Hu S. -M. *Deep online video stabilization with multi-grid warping transformation learning* // *IEEE Transactions on Image Processing*.– 2019.– Vol. **28**.– No. 5.– Pp. 2283–2292 .  [arXiv:1909.02641](#)  [↑11](#)

Поступила в редакцию 11.03.2023;
одобрена после рецензирования 23.03.2023;
принята к публикации 28.03.2023;
опубликована онлайн 07.07.2023.

Рекомендовал к публикации


д.ф.-м.н. А. М. Елизаров

Информация об авторе:



Виталий Петрович Фраленко


Кандидат технических наук, ведущий научный сотрудник ИЦМС ИПС им. А.К. Айламазяна РАН. Область научных интересов: интеллектуальный анализ данных и распознавание образов, искусственный интеллект и принятие решений, параллельно-конвейерные вычисления, сетевая безопасность, диагностика сложных технических систем, графические интерфейсы.

 0000-0003-0123-3773

e-mail: alarmod@pereslavl.ru

Автор заявляет об отсутствии конфликта интересов.

UDC 004.42:004.896

 10.25209/2079-3316-2023-14-2-3-26

Improving quality of video stream from the unmanned aerial vehicle technical vision system

Vitaly Petrovich **Fralenko**

Ailamazyan Program Systems Institute of RAS, Ves'kovo, Russia

Abstract. The study contains the results of work on the software and hardware complex to improve the quality of video data obtained from unmanned aerial vehicles. Including the tasks of independent video-flow images deconvolution (motion blur removal) and stabilization of the video stream using machine learning and artificial intelligence methods. Analytical and practical results are presented that allowed to choose solutions for processing data from UAVs in real time.

Key words and phrases: UAV, deconvolution, stabilization, real time, experimental data

2020 *Mathematics Subject Classification:* 68T07; 68T40, 68U10

Acknowledgments: This work was financially supported by the Russian Science Foundation, project 21-71-10056

For citation: Vitaly P. Fralenko. *Improving quality of video stream from the unmanned aerial vehicle technical vision system.* Program Systems: Theory and Applications, 2023, **14**:2(57), pp. 3–26. https://psta.psiras.ru/read/psta2023_2_3-26.pdf

Introduction

Currently, special attention is paid to the analysis and interpretation of video stream from unmanned aerial vehicles (UAVs), which should ensure the solution of the problems of the allocation and recognition of target objects, positioning and tracking, performed from the UAV's board, including autonomously mode and with using hardware and software on Earth, which is vital to have clear images without "motion blur" effect, and frames in the processed video stream should replace each other without sharp shifts of the camera. Despite successful aerodynamic and layout solutions, domestic developments for equipping UAVs are still losing foreign in terms of the on-board technical vision systems quality. To ensure the necessary indicators, the creation of mathematical, algorithmic and software is required, capable of performing real-time mode of the main functions of exporting and improving the frames from UAVs.

To conduct experiments with existing and modified software, was used a general purpose processor Intel Core i3 8300 and a graphic computing accelerator NVIDIA GTX 1080 Ti with 11 GB of video memory. As experimental data, images of 1280x720 pixels received from a mobile chamber (Dataset Gopro¹), as close as possible from real technical vision systems of the UAV.

1. Deconvolution of UAV data stream

Real images from UAVs contain apparent motion blur caused by the sharp camera displacement due to operator control inputs, wind loads, and the operation of orientation systems on the ground, including GLONASS and GPS. All these factors can occur simultaneously, which destabilizes the camera's position in space. In addition, objects such as cars, bicycles, pedestrians, and animals can also move during the capture of the same frame. Therefore, the camera may inadequately capture an object in the frame due to 1) rapid camera or object movement, or 2) suboptimal camera settings (for example, an excessively long exposure time, or a very slow or

¹The GoPro dataset for deblurring (*Papers with Code*[®]) consists of 3,214 blurred images with the size of 1,280×720 that are divided into 2,103 training images and 1,111 test images. The dataset consists of pairs of a realistic blurry image and the corresponding ground truth shapr image that are obtained by a high-speed camera.

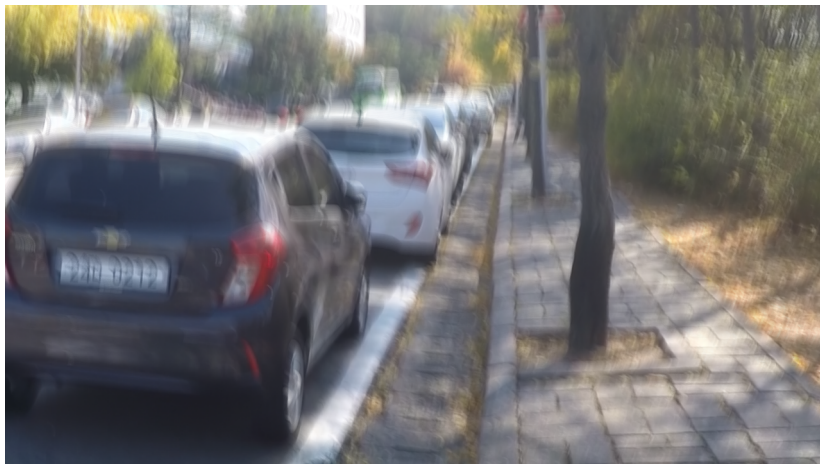


FIGURE 1. Original image example 1

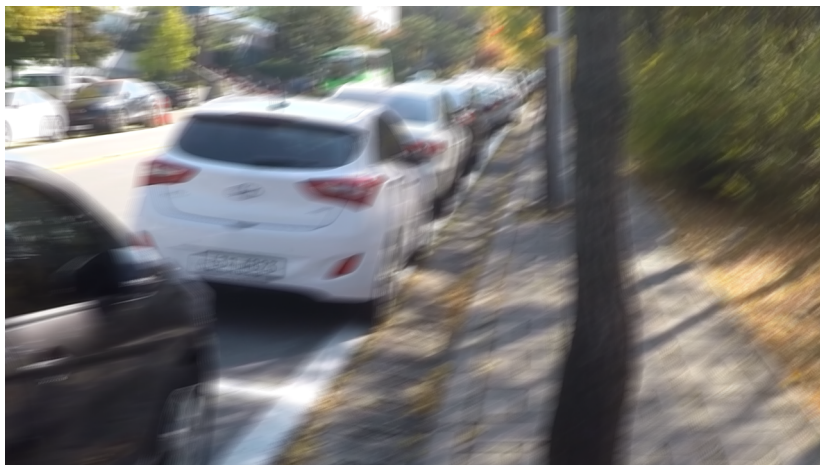


FIGURE 2. Original image example 2

overheated matrix). Deconvolution is a method of compensating for these factors that cause video stream frames to become blurred. We propose to use for deconvolution the methods of mathematical statistics and artificial neural networks (ANNs) trained with publicly available datasets.

Examples of processed images are shown in Figure 1-2.

TABLE 1. Deconvolution results

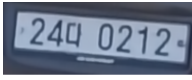
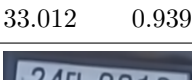
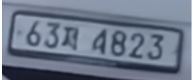
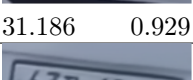
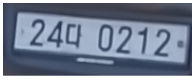
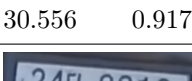
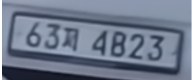
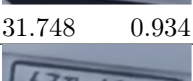

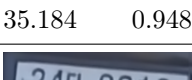
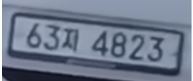
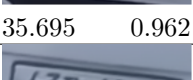
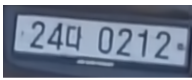
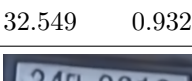
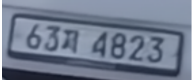
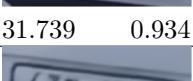
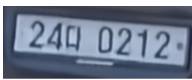
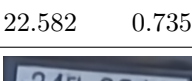
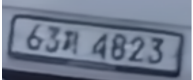
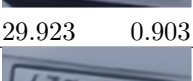
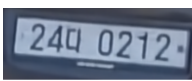
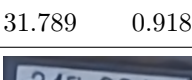
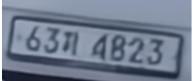
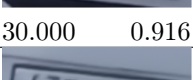
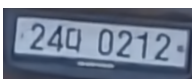
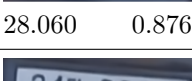
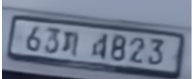
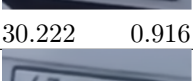
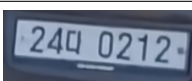

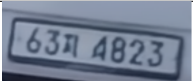

Method	Processing time, s.	Fragment 1		Fragment 2	
		PSNR	SSIM	PSNR	SSIM
<i>Test-time Local Converter (TLC)</i> ^{URL} [1]	13.5	 33.012	 0.939	 31.186	 0.929
<i>Multi-Axis MLP</i> ^{URL} [2]	19.5	 30.556	 0.917	 31.748	 0.934
<i>EFNet</i> ^{URL} [3]	1.10	 35.184	 0.948	 35.695	 0.962
<i>Learning degradation</i> ^{URL} [4]	3.60	 32.549	 0.932	 31.739	 0.934
<i>Deep Generalized Unfolding (DGU)</i> ^{URL} [5]	5.80	 22.582	 0.735	 29.923	 0.903
<i>NAFNet width32</i> ^{URL} [6]	5.00	 31.789	 0.918	 30.000	 0.916
<i>Stripformer</i> ^{URL} [7]	0.03	 28.060	 0.876	 30.222	 0.916
<i>Uformer</i> ^{URL} [8]	9.40	 32.296	 0.928	 31.477	 0.929

Table 1 lists algorithmic and software solutions designed for the automated motion blur removal. The information is provided as individual

frames average processing time (with using the appropriate solution) and original processing results in the form of target areas fragments. It should be noted that all current solutions are implemented as software code using graphics processors for calculations.

This study addressed to the subtask of restoring areas of a video frame with a car number visible. UAVs are actively used worldwide for automated inspection and registration of car numbers, making this subtask relevant. In this case, images with car numbers from the Republic of Korea are used, the third symbol in the car room is a letter prefix, not a figure, as it might seem. For a numerical assessment, the corresponding relationships of the Peak Signal-to-Noise Ratio (PSNR) and the values of the Structure Similarity (SSIM) were obtained, the *skimage.metrics*^{URL} program library was used. A comparison of the obtained images with images without motion blur (subgroup images “sharp” from the GoPro dataset) was performed. The metrics for comparing PSNR and SSIM image pairs for the damaged original blurred images of car numbers (see Figure 1–2), respectively, are as follows: 18.42 and 0.57 (Figure 1); 18.79 and 0.56 (Figure 2). With a complete coincidence of images, the value of $SSIM = 1$, for completely different images $SSIM = 0$.

Experimental studies conducted using these and other images from the GoPro dataset have shown that the Stripformer neural network performs real-time processing while producing high-quality output images. Stripformer has an architecture based on the transformers idea. The “visual transformer” consists of a tokenizer, transformer, and projector, and a more detailed description of its operating principles can be found in the study [9]. Stripformer constructs internal and external strip tokens during the recalculation of informative image features in horizontal and vertical directions, allowing it to identify noisy fragments with different orientations (direction of blur propagation). The neural network combines external and internal strip attention layers to determine the degree and character of a blur. In addition to detecting specific blurred patterns of different orientations and sizes, Stripformer works successfully even



FIGURE 3. Result of removing motion blur in the image using Stripformer

without data on the dynamics of the scene and does not require a huge amount of training data. It is worth noting the high quality of the EFNet neural network.

An example of Stripformer's model result is shown in Figure 3. A trained original model requires a video card with at least 8 GB of video memory for its work. Within the framework of this study, a small modification of the StripFormer software code has been performed, related to the features of the model launch on the stated general purpose processor using the *PyTorch*^{URU} library, but it works unacceptably for a long time, about 90 seconds for each frame, which is 3000 times longer than on the used video card.

Thus, the proposed solution allows for obtaining data from UAVs without motion blur in real-time.

2. Video stream stabilization

During video stream stabilization, the frame is cropped on all sides, and the remaining parts of the frame are used to compensate for motion and

generate the final image. This compensation is usually performed using ANN and statistical methods based on previous frames, which requires maintaining a buffer of a certain size.

In this study, test sets of 100 frames were used, restored using Stripformer and fed at a rate of 24 frames per second.

Next are listed the algorithmic and software solutions intended for automated UAV video stream stabilization. The average processing time for the entire video file is provided. Each solution is accompanied by a comment reflecting the acquired experience of its use, some are implemented with support of a Graphics Processing Units.

- *VidGear*^{URL} [10]. 3.91 seconds. The frame borders are cropped using a border offset and scaling. Initialization of the stabilizer is required, with input data of about 24 frames used. The time required to prepare the stabilizer for operation is approximately 0.70 s., that is, it is performed in the background as the frames turn 30% faster than the frames enter to the processing.
- *FuSta: Hybrid Neural Fusion*^{URL} [11]. 1200 seconds. A graphics accelerator is used. At the first step, a script is launched with the generation of optical stream data (*python main.py* with parameters), which simultaneously performs video stabilization, but without cutting the edges, zoom, etc. Complete processing of 100 frames took 5 minutes. The final stabilized file turned out with a resolution of 960x576 pixels. At the second step, the main program is launched using the collected data of the optical flow, *python run_FuSta.py* with the *-temporal_width 18* instead of the original *-temporal_width 41*. Otherwise, the graphic accelerator is not enough of the available 11 GB video memory. The original library contained the already unsupported codes of the Flownet2 neural network, so its software codes were rewritten. Result — a set of images in the initial resolution. Processing is performed in 15 minutes. The video obtained from these final frames is smooth, however, in contrast to the results obtained using the *python main.py* script, there are artifacts on the boundaries of objects “car” and “tree” (see Figure 4).



FIGURE 4. Artifacts that received when working with Fusta



FIGURE 5. Original image to stabilization and an artifact with a car inclination using the Constant High profile of the Meshflow method

- *MeshFlow: Minimum Latency Online Video Stabilization* ^{URL} [12], 1260 seconds (see Figure 5). Experiments were conducted with two settings

profiles. The Original profile provides excellent quality without borders, and the frame is automatically cropped. The Constant High profile has artifacts in the form of unnaturally tilted objects, such as cars.

- *Video Stabilization with L1 optimal camera paths*^{URL} [13]. 0.50 seconds. The resulting video is “wavy”. The camera sometimes moves towards objects in the frame and then back, what looks unnatural.
- *Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths*^{URL} [13] (additional image preprocessing is used). 120 seconds. The resulting image is significantly cropped when high levels of stabilization are used. For example, at a threshold of 40 pixels, an image of 1280x720 is cropped at the edges to 1180x620. Using a lower threshold gives an insufficient level of video stabilization. Processing is divided into two stages: 2 minutes for preprocessing, followed by fast stabilization using the obtained data.
- *Video stabilization using homography transform*^{URL} [14]. 4.18 seconds. The resulting image is significantly cropped, especially at high stabilization coefficients. The best results are obtained with realtime-stabilization, with the *amount* parameter set to 20.
- *DUT: Learning Video Stabilization*^{URL} [15]. 32 seconds. A graphics accelerator is used. The method relies on several pre-trained neural networks. The approach is based on self-learning on unstable video data. The car image in the resulting video is slightly offset left and right (not as essential as in the. 5), but the overall resulting video can be considered of good quality.
- *Deep Iterative Frame Interpolation for Full-frame Video Stabilization*^{URL} [16]. 100 seconds. Processing is performed iteratively, with the number of iterations controlled by the *n_iter* parameter, which is set to 3 by default. The resulting video stream is unstable, and the overall quality of the resulting video stream is low.
- *PWStableNet*^{URL} [17]. 8 seconds. A graphics accelerator is used. The resulting video has jerky movements.









The best stabilization results were achieved using the VidGear library, which accumulates an array of key points over a specified number of frames

























with subsequent real-time updating. The accumulated data is used to estimate the motion of the UAV camera with compensation for its jerks. The implementation corresponds to the approach used in the *OpenCV*^{URL} library. The overall processing occurs in a pipeline mode, as deconvolution and stabilization are performed in parallel.








Conclusion

The conducted research allowed to obtain a solution to two of the most important tasks of improving the video stream from UAVs: removing blurs and stabilizing the video. The Stripformer neural network removes motion blur frame by frame, and then the processed data immediately enters the buffer of the stabilization module based on the VidGear library, which stabilizes the video stream. The final solution allows for the full cycle of video stream processing with a delay of no more than 0.03 seconds due to the use of a pipeline method for data processing. Processing is carried out in real-time mode.

References

- [1] Xiaojie Chu, Liangyu Chen, Chengpeng Chen, Xin Lu. “Improving image restoration by revisiting global information aggregation”, *ECCV 2022: Computer Vision – ECCV 2022 (October 23–27, 2022, Tel Aviv, Israel), Lecture Notes in Computer Science*, vol. **13678**, Springer, Cham, 2022, pp. 330–351 .  [arXiv:2112.04491](https://arxiv.org/abs/2112.04491)  [↑18](#)
- [2] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, Yinxiao Li. “MAXIM: multi-axis MLP for image processing”, *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, 34 pp.  [arXiv:2201.02973](https://arxiv.org/abs/2201.02973)  [↑18](#)
- [3] Lei Sun, Christos Sakaridis, Jingyun Liang, Qi Jiang, Kailun Yang, Peng Sun, Yaozu Ye, Kaiwei Wang, Luc Van Gool. “Event-based fusion for motion deblurring with cross-modal attention”, *ECCV 2022: Computer Vision – ECCV 2022 (October 23–27, 2022, Tel Aviv, Israel), Lecture Notes in Computer Science*, vol. **13678**, Springer, Cham, pp. 412–428; 2023, 17 pp.  [arXiv:2112.00167](https://arxiv.org/abs/2112.00167)  [↑18](#)
- [4] Dasong Li, Yi Zhang, Ka Chun Cheung, Xiaogang Wang, Hongwei Qin, Hongsheng Li. “Learning degradation representations for image deblurring”, *Computer Vision – ECCV 2022*, *ECCV 2022*, Lecture Notes in Computer Science, eds. Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T.; 2022, 18 pp.  [arXiv:2208.05244](https://arxiv.org/abs/2208.05244)  [↑18](#)

- [5] Chong Mou, Qian Wang, Jian Zhang. “Deep generalized unfolding networks for image restoration”, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17399–17410, 12 pp. [arXiv:2204.13348](#)    18
- [6] L. Chen, X. Chu, X. Zhang, Sun J.. “Simple baselines for image restoration”, *Computer Vision – ECCV 2022*, ECCV 2022, Lecture Notes in Computer Science, eds. Avidan S., Brostow G., Cissé M., Farinella G.M., Hassner T., 2022; 21 pp.  [arXiv:2204.04676](#)   18
- [7] Fu-Jen Tsai, Yan-Tsung Peng, Yen-Yu Lin, Chung-Chi Tsai, Chia-Wen Lin. “Stripformer: strip transformer for fast image deblurring.”, *Computer Vision – ECCV 2022*, ECCV 2022, Lecture Notes in Computer Science, vol. **13679**, eds. Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T., 2022; 17 pp.  [arXiv:2204.04627](#)   18
- [8] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, Houqiang Li. “Uformer: a general U-shaped transformer for image restoration”, *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, 17 pp.  [arXiv:2106.03106](#)   18
- [9] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, Peter Vajda. *Visual transformers: token-based image representation and processing for computer vision*, 2020, 12 pp. [arXiv:2006.03677](#)   19
- [10] A. Thakur, Z. Papanikopos, C. Clauss, C. Hollinger, I. M. Andolina, V. Boivin, enarche-ahn, freol35241, B. Lowe, M. Schoentgen, R. Bouckennooghe. “abhiTronix/vidgear: VidGear v0.2.6”, 2022.    21
- [11] Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, Jia-Bin Huang. “Hybrid neural fusion for full-frame video stabilization”, *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 2299–2308.  [arXiv:2102.06205](#)   21
- [12] Shuaicheng Liu, Ping Tan, Lu Yuan, Jian Sun, Bing Zeng. “MeshFlow: minimum latency online video stabilization”, *ECCV 2016: Computer Vision – ECCV 2016 (October 11-14, 2016, Amsterdam, The Netherlands)*, Lecture Notes in Computer Science, vol. **9910**, 2016, pp. 800–815.    22
- [13] M. Grundmann, V. Kwatra, I. Essa. “Auto-directed video stabilization with robust L1 optimal camera paths”, *CVPR ’11: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (20–25 June 2011, Colorado Springs, CO, USA)*, IEEE Computer Society, 2011, ISBN 978-1-4577-0394-2, pp. 225–232.   23

- [14] M. Grundmann, V. Kwatra, I. Essa. “Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths”, *CVPR ’11: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition* (20–25 June 2011, Colorado Springs, CO, USA), IEEE Computer Society, 2011, pp. 225–232.  [DOI](#)
- ↑²³
- [15] Yufei Xu, Jing Zhang, Stephen J. Maybank, Dacheng Tao. “DUT: learning video stabilization by simply watching unstable videos”, *IEEE Transactions on Image Processing*, **31** (2022), pp. 4306–4320.  [DOI](#)  [arXiv:2011.14574](#) ↑²³
- [16] Jinsoo Choi, In So Kweon. “Deep iterative frame interpolation for full-frame video stabilization”, *ACM Transactions on Graphics*, **39**:1 (2019), id. 4, 9 pp.  [DOI](#)  [arXiv:1909.02641](#) ↑²³
- [17] M. Wang, G.-Y. Yang, J.-K. Lin, S.-H. Zhang, A. Shamir, S.-P. Lu, S.-M. Hu. “Deep online video stabilization with multi-grid warping transformation learning”, *IEEE Transactions on Image Processing*, **28**:5 (2019), pp. 2283–2292.  [DOI](#)  [arXiv:1909.02641](#) ↑²³


Received	11.03.2023;
approved after reviewing	23.03.2023;
accepted for publication	28.03.2023;
published online	07.07.2023.

Recommended by

*prof. A. M. Elizarov***Information about the author:**

Vitaly Petrovich Fralenko

PhD, Leading Researcher at RCMS Ailamazyan Program Systems Institute. The field of scientific interests: intellectual data analysis and images recognition, artificial intelligence and decision making, parallel-conveyor computing, network security, diagnosis of complex technical systems, graphic interfaces.

 0000-0003-0123-3773

e-mail: alarmod@pereslavl.ru

The author declare no conflicts of interests.