A brief review of GCP-NET: *a network for joint denoising and demosaicing of real-world burst images*

Valéry Dewil

valery.dewil@ens-paris-saclay.fr

MLBriefs II presentation day - December 12

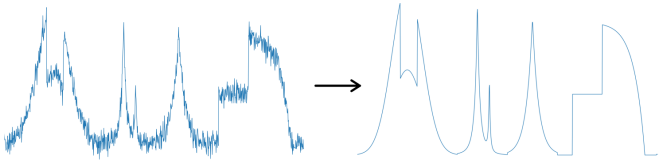
Summary

In this presentation

- Very quick introduction to the problem of Joint Denoising and Demosaicing
- Presentation of GCP-Net
- An experiment: checking the Green Channel Prior hypothesis
- Motivations of this IPOL paper / demo
- An experiment: testing how GCP-Net behaves over different motions
- Description of the online demo

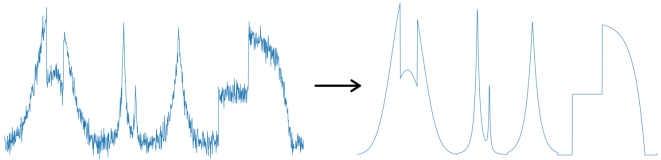
Introduction to the problem of Joint Denoising and Demosaicing

Introduction

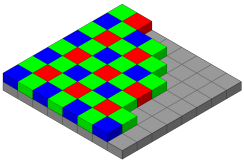


Denoising

Introduction

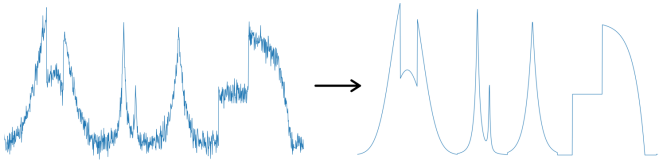


Denoising

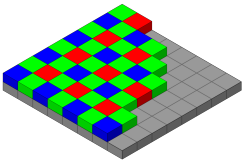


Demosaicing

Introduction



Denoising



Demosaicing

Pipeline

Literature on JDD*

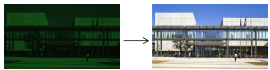
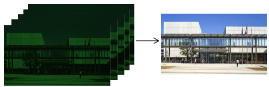


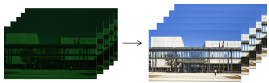
Image JDD:

- Goossens *et al.* 2015
- Gharbi *et al.* 2016
- Dong *et al.* 2018



Burst JDD:

- Wronski *et al.* 2019
- Guo *et al.* 2021
- Lecouat *et al.* 2021
- Eboli *et al.* 2022



Video JDD:

- Buades *et al.* 2018
- Ehret *et al.* 2019
- Yue *et al.* 2020
- Dewil *et al.* 2023

* Joint Denoising and Demosaicing

Literature on JDD*

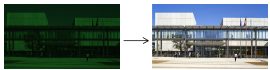
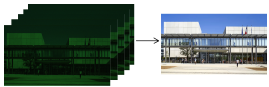


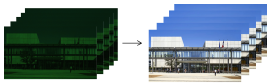
Image JDD:

- Goossens *et al.* 2015
- Gharbi *et al.* 2016
- Dong *et al.* 2018



Burst JDD:

- Wronski *et al.* 2019
- **Guo *et al.* 2021**
- Lecouat *et al.* 2021
- Eboli *et al.* 2022



Video JDD:

- Buades *et al.* 2018
- Ehret *et al.* 2019
- Yue *et al.* 2020
- Dewil *et al.* 2023

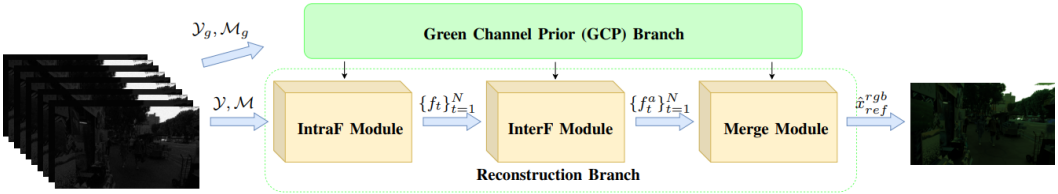
* Joint Denoising and Demosaicing

Description of GCP-Net

What is GCP-Net

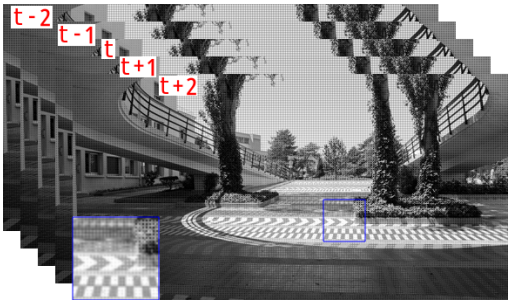
- GCP-Net is a deep learning method, solving the JDD task (for Bayer-CFA)
- It's a burst method: 5 frames (large improvment compared with single-image method)
- It handles signal-dependent noise (*Poisson-Gaussian* model). Goal of the authors: deal with real data
- It requires the noise maps
- It has improved the state of the art (on synthetic noisy images + on real-world burst images)

GCP-Net architecture



- The network is composed of 2 branches
- One branch (convs+LeakyReLU blocks) takes only the green channel of each frame
- The other takes the full frames and features from the green-channel branch
- Features extracted from the green channel are used to guide: the deep feature extraction (IntraF), the motion alignment (InterF with DConvs), the upsampling (Merge module which aggregates)

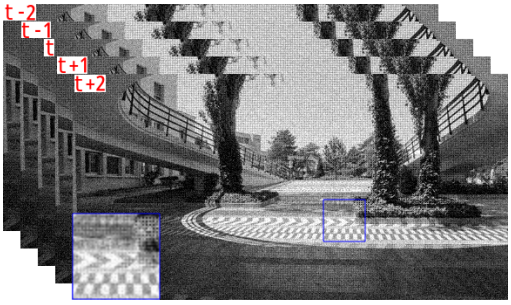
GCP-Net: input & output



Input: a clean burst of 5 frames + noise params

Output: a denoised version of the center frame

GCP-Net: input & output



Input: a noisy burst of 5 frames + noise params

Output: a denoised version of the center frame

Comparison with state of the art

- Great work from the authors of comparison with other methods!
- Quantitative & qualitative results
- Synthetic problems as well as real-world case
- Comparison with:
    - SOTA in single image JDD (Flex-ISP, ADMM)
    - SOTA in video denoising followed by DemosaicNet (VBM3D, EDVR, RViDeNet)
    - KPN followed by DemosaicNet
    - GCP has improved all these methods

Comparison with state of the art

Methods	Average for low level of noise	Average for high level of noise
FlexISP	25.57 / 0.7319	23.16 / 0.5789
ADMM	27.79 / 0.7714	26.89 / 0.7520
VBM3D+DMN	28.30 / 0.7896	26.86 / 0.7444
KPN+DMN	30.43 / 0.8860	28.48 / 0.8254
EDVR+DMN	30.85 / 0.9014	29.07 / 0.8556
RviDeNet+DMN	32.30 / 0.9267	30.85 / 0.8998
EDVR*	30.99 / 0.9220	29.16 / 0.8578
RviDeNet*	33.53 / 0.9395	31.67 / 0.9117
GCP-Net	**33.91 / 0.9429**	**32.30 / 0.9205**

Average PSNR / SSIM on the Vid4 dataset

Comparison with state of the art

(a) Noisy image (b) VBM3D+DMN (c) KPN+DMN (d) EDVR*

(e) RViDeNet+DMN (f) RViDeNet* (g) Ours (h) GT

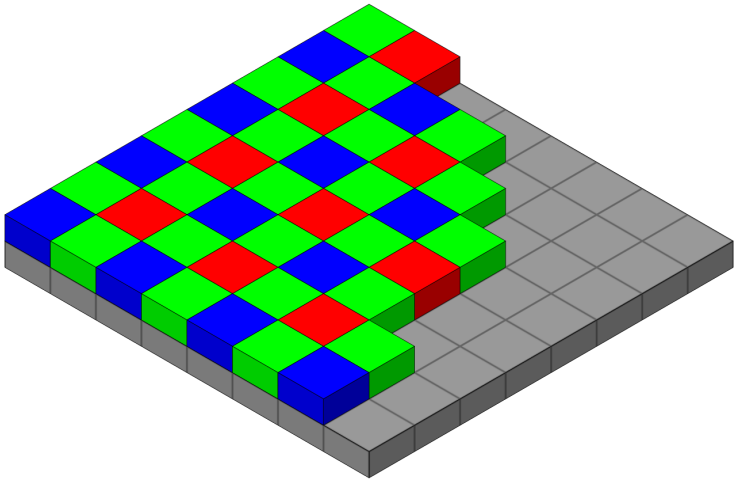
Green Channel Prior hypothesis

Why use the green channel?

- I was interested in some basic statistics about the green channels (compared to other channels)
- I worked with 270 raw videos from this dataset [1]
- In a *Bayer*-CFA, the green channel has twice the sampling rate (compared with R/B)

[1] *Video joint denoising and demosaicing with recurrent CNNs* '23

Why use the green channel?



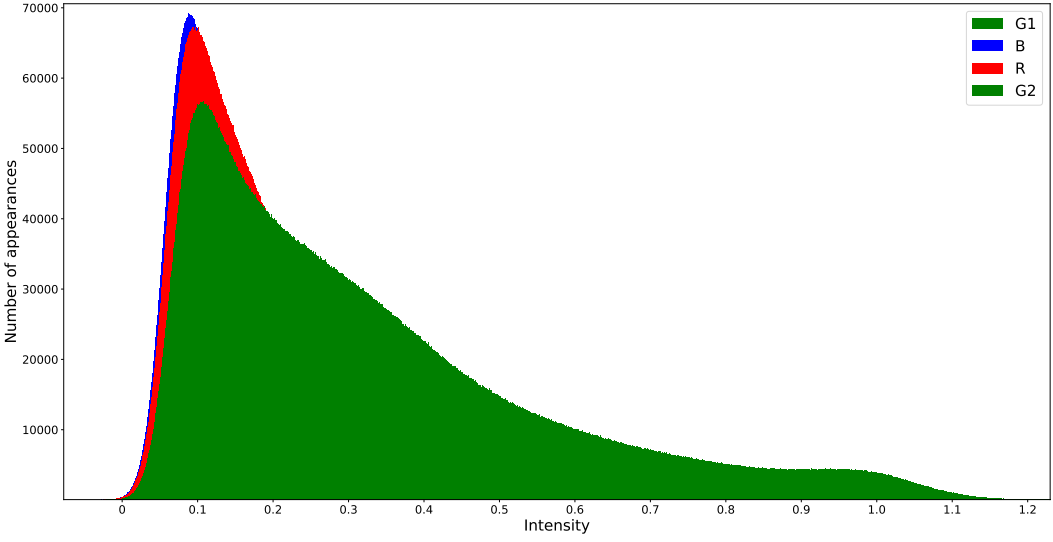
***Bayer*-CFA: the sensor has 2 times more green filters than red/blue**

Why use the green channel?

- I was interested in some basic statistics about the green channels (compared to other channels)
- I worked with 270 raw videos from this dataset [1]
- In a *Bayer*-CFA, the green channel has twice the sampling rate (compared with R/B)

[1] *Video joint denoising and demosaicing with recurrent CNNs* '23

Why use the green channel?



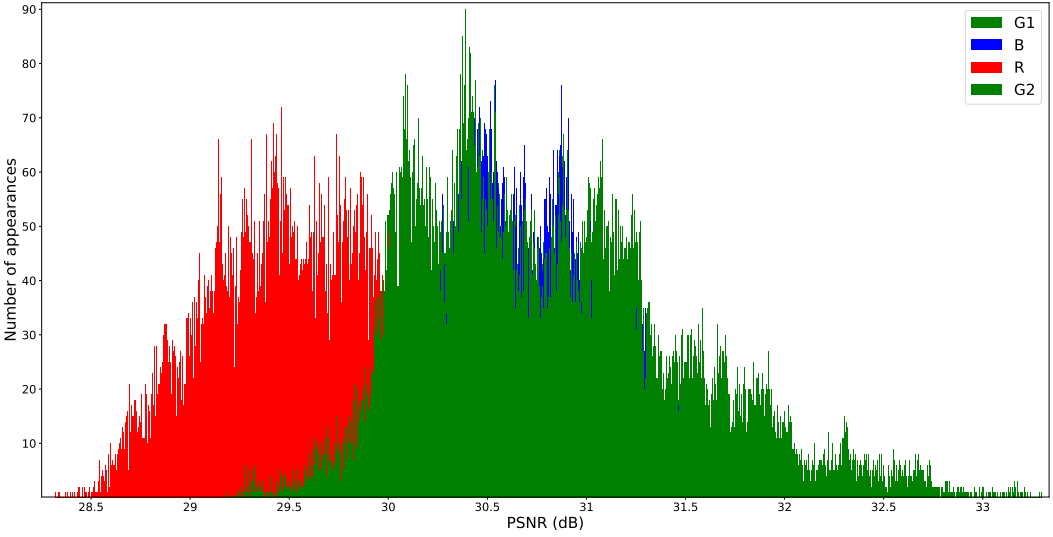
Histogram of intensity over a whole dataset of 270 raw videos

Why use the green channel?

- I was interested in some basic statistics about the green channels (compared to other channels)
- I worked with 270 raw videos from this dataset [1]
- In a *Bayer*-CFA, the green channel has twice the sampling rate (compared with R/B)
- In average, over all the frames, green channels:
  - is brighter than the other channels
  - has a better signal-to-noise ratio
  - has also a higher SSIM

[1] *Video joint denoising and demosaicing with recurrent CNNs* '23

Why use the green channel?



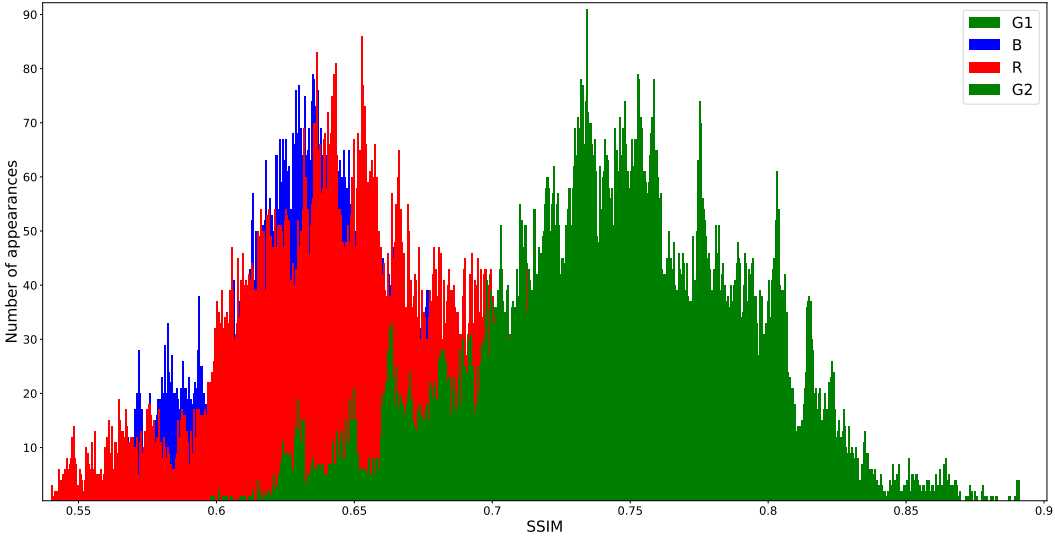
Histogram of PSNR over a whole dataset of 270 raw videos

Why use the green channel?

- I was interested in some basic statistics about the green channels (compared to other channels)
- I worked with 270 raw videos from this dataset [1]
- In a *Bayer*-CFA, the green channel has twice the sampling rate (compared with R/B)
- In average, over all the frames, green channels:
  - is brighter than the other channels
  - has a better signal-to-noise ratio
  - has also a higher SSIM

[1] *Video joint denoising and demosaicing with recurrent CNNs* '23

Why use the green channel?



Histogram of SSIM over a whole dataset of 270 raw videos

Why use the green channel?

- I was interested in some basic statistics about the green channels (compared to other channels)
- I worked with 270 raw videos from this dataset [1]
- In a *Bayer*-CFA, the green channel has twice the sampling rate (compared with R/B)
- In average, over all the frames, green channels:
    - is brighter than the other channels
    - has a better signal-to-noise ratio
    - has also a higher SSIM

[1] *Video joint denoising and demosaicing with recurrent CNNs* '23

Motivations of this IPOL paper / demo

Motivations of the IPOL paper

- *Main motivation from far*: have a JDD method in IPOL
- Have a simple demo, able to deal with the burst case
- The authors presented their method, evaluated it on their dataset and compared with the others
- Evaluate the generalization of GCP-Net. For this IPOL demo/paper: 2 options
  - Option 1: how does it perform with variable noise level (the authors only evaluate for 2)
  - Option 2: how does it perform on sequences with different motion statistics
- Users can test / play with the online demo

How does GCP-Net behaves with motion of different statistics?

Generalization of GCP-Net on data with different motion statistics

- GCP-Net was trained on synthetic data from a video dataset Vimeo90k
- Vimeo90k is a smooth dataset (average motion magnitude between two consecutive frames $\leq$ 8px)
- GCP-Net was trained with an explicit and internal motion compensation (DConvs)
- I wanted to test this motion alignment in difficult conditions (large motion + strong noise).

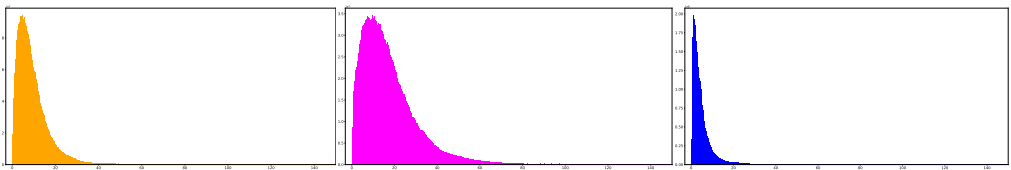
Generalization of GCP-Net on data with different motion statistics

- GCP-Net was trained on synthetic data from a video dataset Vimeo90k
- Vimeo90k is a smooth dataset (average motion magnitude between two consecutive frames $\leq$ 8px)
- GCP-Net was trained with an explicit and internal motion compensation (DConvs)
- I wanted to test this motion alignment in difficult conditions (large motion + strong noise).
- Experiment: evaluate GCP-Net on 3 versions of noisy dataset:
    - normal motion (dataset from [1] set as *reference* conditions)
    - expanded motion, by temporal downsampling (1/2)
    - stabilized motion (video stabilization from IPOL)

Some statistics about the estimated motion distances in the 3 datasets

Dataset version	average	med	min (x1e-5)	max	1^{st}.	2^{nd}	98^{th}	99^{th}
normal sequences	9.52	7.55	1.21	325.3	.477	.733	32.8	41.7
temp. downsampled sequences	17.9	14.6	17.3	264.8	.884	1.37	58.9	71.9
stabilized sequences	5.17	3.38	2.61	289.6	.295	.401	25.1	34.4

Table: Some statistics about the estimated motion distances (subpixelic) in adjacent frames. The values are given in number of pixels.



Results

Noise levels	normal sequences	temporally downsampled	stabilized sequences sequences
minimum	44.19 / 0.990	43.67 / 0.990	45.37 / 0.992
low level	40.06 / 0.973	39.40 / 0.970	41.45 / 0.979
middle level	38.83 / 0.966	38.13 / 0.962	40.31 / 0.974
high level	38.00 / 0.961	37.28 / 0.955	39.54 / 0.969
maximum	36.78 / 0.951	36.06 / 0.945	38.39 / 0.962

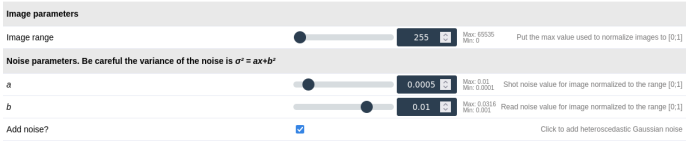
- Of course, the performance degrades with larger motion and *vice-versa*
- Gap $\approx$ 0.6dB between the normal and temporally downsampled burst
- Surprisingly, same gap irrespective of the noise level
- Performance gap on stabilized data depends on the noise level
- $\longrightarrow$ GCP-Net is very bad for strong noise and get close to a single-image method (unless motion is very smooth)

Online demo

Online demo

The parameters

- A novice can just click on run!
- Very few parameters!
  - A normalization factor
  - Heteroscedastic Gaussian noise parameters
  - A button to add noise



Online demo

Some generalities

- **Everything in the paper can be tested by yourself**
- You can play with the amount of noise and see the impact on the performance
- You can also play with the motion: I put an example of small clip with the 3 options (normal, stabilized and larger motion)
- It takes about $\approx$ 20 sec

Online demo

Output

- The output of the network is a denoised linear RGB (center frame)
- The results are displayed in the gallery, in linear RGB + a simplistic sRGB
- You can download the data (clean & noisy raw + output in linear RGB)
- If a ground-truth is provided: - PSNR & SSIM are computed

  - this ground-truth is displayed (linear + sRGB)

Online demo

Visual results generated with the IPOL demo

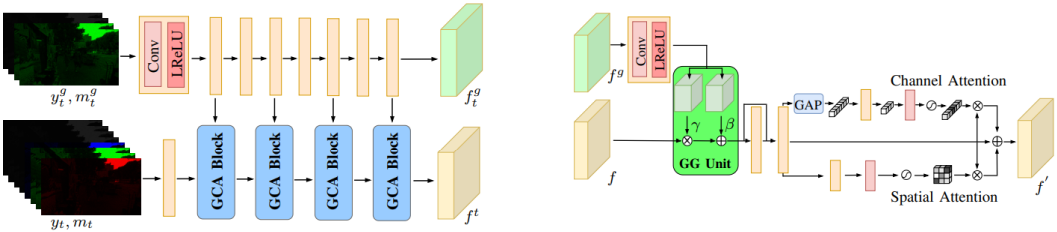
(a) GT *(b)* noisy *(c)* GCP-Net

Conclusion

- IPOL has been supplemented by an implementation of GCP-Net
- The main interest here is indeed the demo (*in my opinion*)
- The goal is not to copy the original article...
- In the IPOL article, extensive tests with GCP-Net (in different conditions)
- Make your own experiments (no need to have your own data)

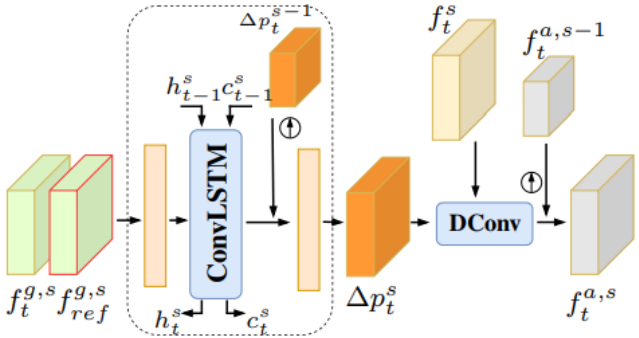
Original article: *Joint denoising and demosaicking with green channel prior for real-world burst images*, Guo *et al.* '21
Github code of GCP-Net: https://github.com/GuoShi28/GCP-Net
Demo ID in IPOL and associated paper: 77777000369

IntraF module



(left) Architecture of the Intra-frame (IntraF) module. *(right)* Architecture of the GCA block.

InterF module



Architecture of the Inter-frame (InterF) module for frame t with scale s.