Supplemental Material for Handheld Multi-Frame Super-Resolution

BARTLOMIEJ WRONSKI, IGNACIO GARCIA-DORADO, MANFRED ERNST, DAMIEN KELLY, MICHAEL KRAININ, CHIA-KAI LIANG, MARC LEVOY, and PEYMAN MILANFAR, Google Research

S SUPPLEMENT

S.1 Adaptive Super-Resolution and Denoising

Fig. 1. **Denoising:** Example effect of local kernel denoising, **Left**: Low light image without local kernel denoising, $k_{denoise} = 1.0$. **Middle**: image with strong local kernel denoising $k_{denoise} = 5.0$. **Right**: local denoising mask. Black pixels denote areas where we do not apply any spatial denoising and adjust kernel values for super-resolution, while white pixels denote areas where we do not observe enough image details to justify super-resolution and adjust the kernel values for denoising. By analyzing the local structure, our algorithm can cover a continuous balance between resolution enhancement and spatio-temporal denoising.

In Section 5.1.2 we describe adapting the spatial support of the sampling kernel based on the local gradient structure tensor. We use the magnitude of the structure tensor's dominant eigenvalue λ_1 to drive the spatial support of the kernel and the trade-off between the super-resolution and denoising, where $\frac{\lambda_1}{\lambda_2}$ is used to drive the desired anisotropy of the kernels (Figure 7 in the main paper text). We use the following heuristics to estimate the kernel shapes (k_1 and k_2 in Equation (4) in the main paper text):

$$
\begin{aligned}
A &= 1 + \sqrt{\frac{\lambda_1}{\lambda_2}}, \\
D &= clamp(1 - \frac{\sqrt{\lambda_1}}{D_{tr}} + D_{th}, 0, 1), \\
\hat{k_1} &= k_{detail} \cdot (k_{stretch} \cdot A), \\
\hat{k_2} &= \frac{k_{detail}}{(k_{shrink} \cdot A)}, \\
k_1 &= ((1 - D) \cdot \hat{k_1} + D \cdot k_{detail} \cdot k_{denoise})^2, \\
k_2 &= ((1 - D) \cdot \hat{k_2} + D \cdot k_{detail} \cdot k_{denoise})^2.
\end{aligned}
$$

Authors' address: Bartlomiej Wronski, bwronski@google.com; Ignacio Garcia-Dorado, ignaciod@google.com; Manfred Ernst, ernstm@google.com; Damien Kelly, damienkelly@google.com; Michael Krainin, mkrainin@google.com; Chia-Kai Liang, ckliang@google.com; Marc Levoy, levoy@google.com; Peyman Milanfar, milanfar@google.com Google Research, 1600 Amphitheatre Parkway, Mountain View, CA, 94043.

We use the symbol A for the computed gradient anisotropy and D for the estimated denoising strength. We use the following tuning parameters: D_{th} as the denoising threshold, D_{tr} as how fast we go from full denoising to no denoising depending on the gradient strength, $k_{stretch}$ as the amount of kernel stretching along the edges, k_{shrink} as the amount of kernel shrinking perpendicular to the edges, k_{detail} as the base kernel standard deviation, and $k_{denoise}$ as the kernel standard deviation suitable for denoising. The denoising strength will make the whole kernel shape bigger and more radial, effectively also overriding the anisotropic stretching in regions that are candidates for denoising.

The reasoning behind these heuristics is that small dominant eigenvalues (comparable to the amount of noise expected in the given raw image) signify relatively flat, noisy regions while large eigenvalues appear around features whose resolution we want to enhance (Figure 1). Figure 1 **left** and **middle** show the visual impact of $k_{denoise}$ parameter, while the contrast of the mask presented on the **right** depends on D_{th} and D_{tr}.

S.2 Tuning Procedure and Parameters

In this section we describe the tuning parameters that we used for the results presented for our algorithm. Parameters that affect the trade-off between the resolution-increase and spatio-temporal denoising (Section S.1) depend on the signal-to-noise ratio of the input frames. In such case the parameters are piece-wise linear functions of SNR in the range [6..30].

$$
\begin{aligned}
T_s &= [16, 32, 64]px, \\
k_{detail} &= [0.25, ..., 0.33]px, \\
k_{denoise} &= [3.0, ..., 5.0], \\
D_{th} &= [0.001, ..., 0.010], \\
D_{tr} &= [0.006, ..., 0.020], \\
k_{stretch} &= 4, \\
k_{shrink} &= 2, \\
t &= 0.12, \\
s_1 &= 12, \\
s_2 &= 2, \\
M_{th} &= 0.8px.
\end{aligned}
$$

The T_s, k_{detail}, and M_{th} are in units of pixels, D_{th} and D_{tr} are in units of gradient magnitude of the image normalized to the range $[0, ..., 1]$. The remaining parameters are either unitless multipliers ($k_{denoise}, k_{stretch}, k_{shrink}$) or operate on color differences normalized by the standard deviation (t, s_1, s_1).

Since our algorithm is designed to produce visually pleasing images taken with a mobile camera, we tuned those parameters based on perceputal image quality assessment ensuring visual consistency for SNR values from 6 to over 30 where the SNR was measured from

a single frame. Next, we discuss the impact of some of those parameters on the final image. The chosen kernel parameters balance the

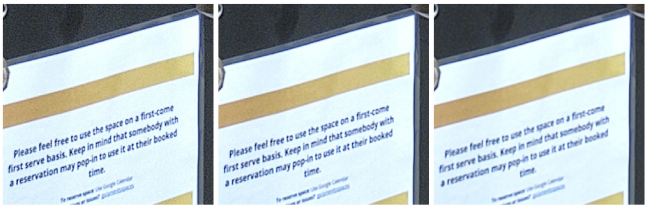

Fig. 2. **Impact of k_{detail} on the visual results. Left:** k_{detail} of 0.1px produces very sharp results with significant amounts of noise and some artifacts. **Middle:** k_{detail} of 0.25px produces results balanced between resolution enhancement and denoising. **Right:** k_{detail} of 0.4px produces over-smoothed results.

resolution enhancement with suppression of noise and artifacts in the image. Figure 2 shows the visual impact of adjusting the base kernel size k_{detail}. Figure 7 presented earlier in the main paper shows how the $k_{stretch}$ and k_{shrink} impact the result, smoothing the edges and getting rid of alignment artifacts that can result from the aperture problem. The T_s is increased from $16px$ to up to $64px$ in very low light situations to increase the robustness of alignment to significant amounts of noise.

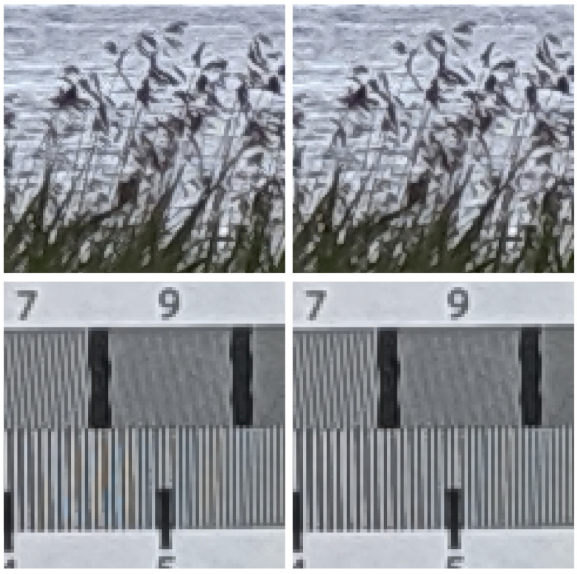

Fig. 3. **Impact of s_2 on the visual results. Top-left:** Too small s_2 of 1 produces small high-frequency artifacts. **Bottom-left:** Too large s_2 of 4 causes over-rejection in highly aliased regions and loss of super-resolution. **Bottom-right and top-right:** s_2 of 2 correctly treats areas with local movement as well as heavily aliased regions.

Tuning of the s and M_{th} is performed to balance the false-positive and the false-negative rate of our robustness logic. A rejection rate that is too large leads to not merging some heavily aliased areas (like test chart images), while too small rejection rate leads to the manifestation of fusion artifacts. The effect of having this parameter too small or too large can be observed in Figure 3. In practice, to balance those effects, we use the same fixed values for all processed images.

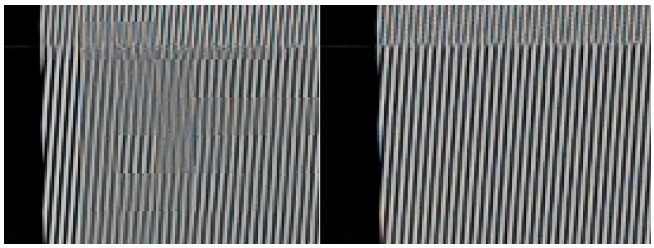

Fig. 4. **High frequency artifacts caused by the aperture problem: Left:** a high resolution and high frequency test chart image without the rejection logic described in Section S.3. Notice the numerous blocky artifacts visible when zoomed-in. **Right:** the same image with the rejection logic detecting variance loss showing no fusion artifacts, but some aliasing and color fringing.

S.3 High Frequency Artifacts Removal

Alignment algorithms (such as block matching or gradient based) fail to correctly align high frequency repetitive patterns (due to the aperture problem). Our robustness logic makes use of both low-pass filtering and comparing local statistics. Therefore, the algorithm as described is prone to producing blocky artifacts in regions containing *only* very high frequency signals, often observed on human-made test charts (Figure 4). To prevent this effect, we detect those regions by analyzing the local variance loss caused by local lowpass filtering. In particular, we compare the local variance before and after the lowpass filtering. When we detect variance loss and a large local variation in the alignment vector field (the same as used in the motion prior in Section 5.2.3), we mark those regions as incorrectly aligned and fully reject them. An example comparison with and without this logic is presented in Figure 4. This heuristic has a trade-off: in some cases, even properly aligned high frequency regions do not get merged.

S.4 Synthetic Data Quality Analysis

We show detailed box plots of our algorithm's performance compared to different demosaicing techniques in Figure 5.

S.5 Robustness Analysis

A PSNR analysis of the robustness on synthetic alignment corruption tests is shown in Figure 6. The strongest quality degradation (50% corrupted image tiles or wrong alignment with random offsets of 0.25 pixels) leads to our algorithm merging only a single frame and PSNR values comparable to simple demosaicing techniques. Additionally, we show examples of burst merging with and without the robustness model in real captured bursts in different difficult conditions in Figure 7.

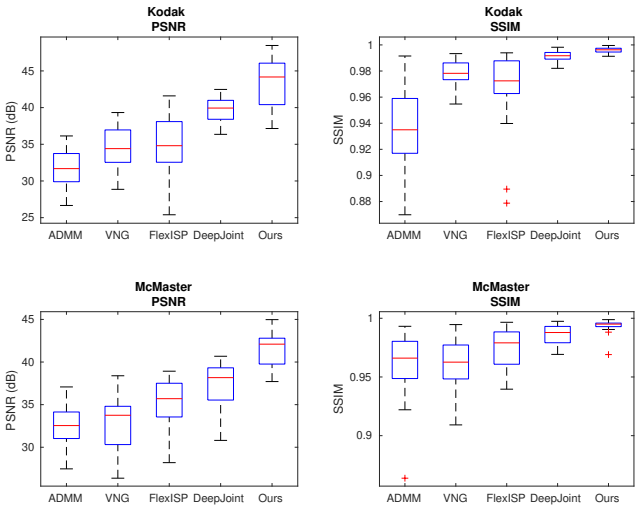

Fig. 5. **PSNR and SSIM comparisons on Kodak and McMaster dataset.** Performance of our algorithm compared to alternate approaches using PSNR and SSIM on synthetic bursts created from the Kodak and McMaster datasets. Our solution can use information present across multiple frames and is significantly better than all other techniques on both synthetic datasets.

S.6 Real Captured Bursts Additional Results

We show some additional comparisons with competing techniques on bursts captured with a mobile camera in Figure 9.

REFERENCES

Edward Chang, Shiufun Cheung, and Davis Y Pan. 1999. Color filter array recovery using a threshold-based variable number of gradients. In *Sensors, Cameras, and Applications for Digital Photography*, Vol. 3650. 36–44.

Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. 2016. Deep joint demosaicking and denoising. *ACM TOG* 35, 6 (2016), 191.

Samuel W Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. 2016. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM TOG* 35, 6 (2016), 192.

Mehdi S. M. Sajjadi, Raviteja Vemulapalli, and Matthew Brown. 2018. Frame-recurrent video super-resolution. In *Proc. CVPR*. 6626–6634.

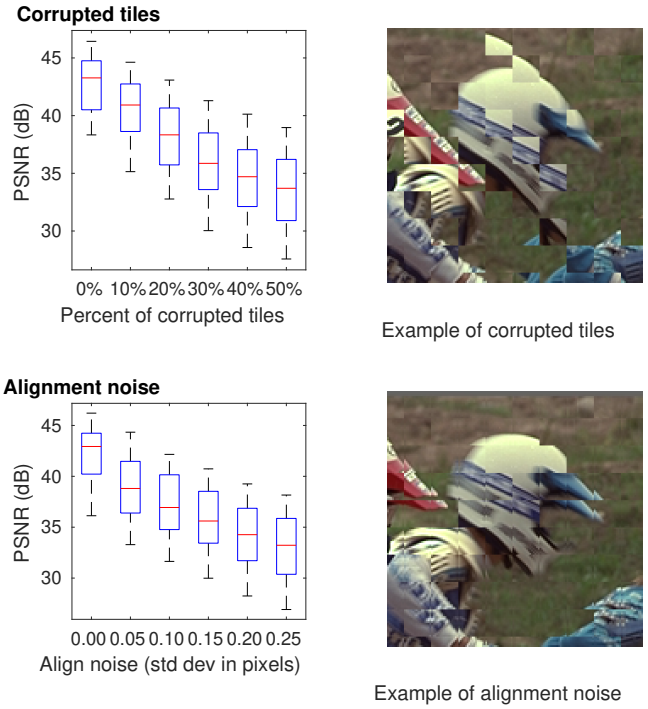

Fig. 6. **PSNR of image quality caused by alignment corruption of synthetic bursts created from Kodak dataset. Top-Left:** PSNR of our algorithm output caused by randomly corrupted and misaligned tiles. **Bottom-Left:** Visual demonstration of this type of distortion at the highest evaluated distortion value. **Top-Right:** PSNR of our algorithm output caused by noise added to the alignment vectors. **Bottom-Right:** Visual demonstration of this type of distortion at the highest evaluated distortion value. With increasing distortion rate we observe gradual quality degradation, as our algorithm rejects most of the frames in the synthetic burst and degrades to a simple gradient-based demosaicing technique.

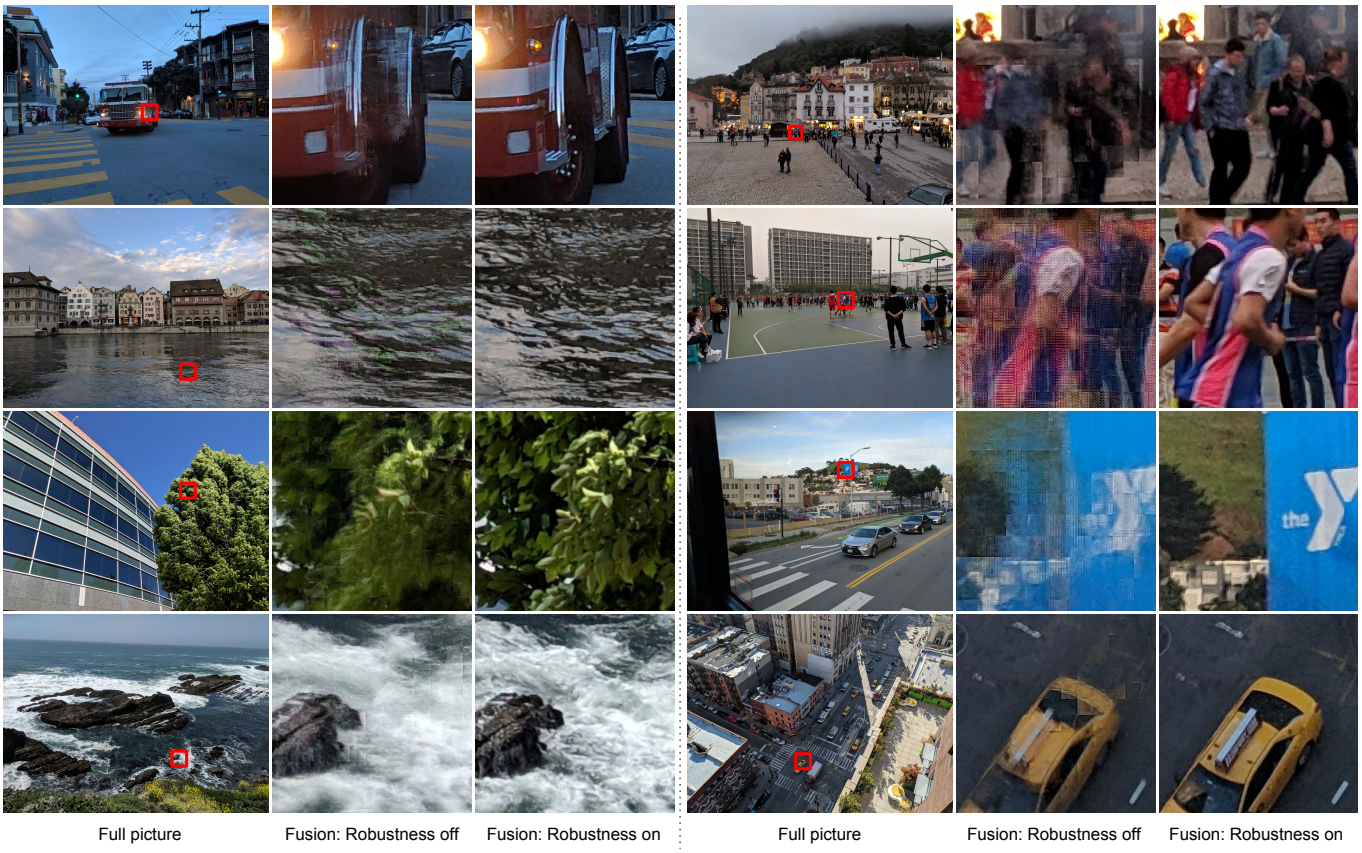

Fig. 7. **Robustness examples: Left:** Full photo. **Middle:** Crop of the photo merged without our robustness model. **Right:** Same region of the photo merged with our robustness model. In real captured bursts, our algorithm is able to handle challenging scenarios including local scene motion, parallax or scene changes like water rippling.

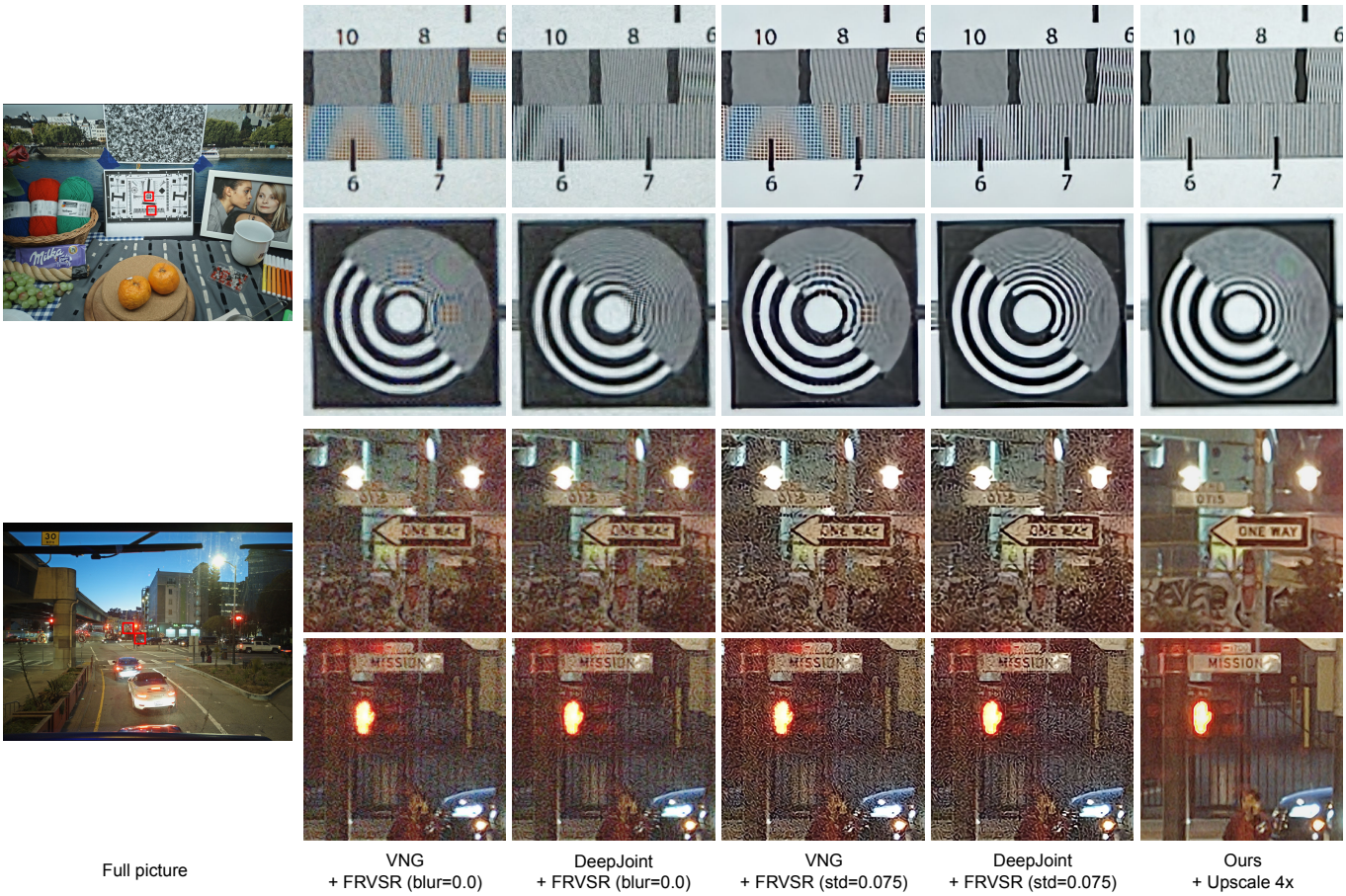

Fig. 8. **Additional comparison with video super-resolution.** Our method compared with *FRVSR* [Sajjadi et al. 2018] applied to bursts of images demosaiced with VNG [Chang et al. 1999] or *DeepJoint* [Gharbi et al. 2016]. Readers are encouraged to zoom aggressively (300% or more).

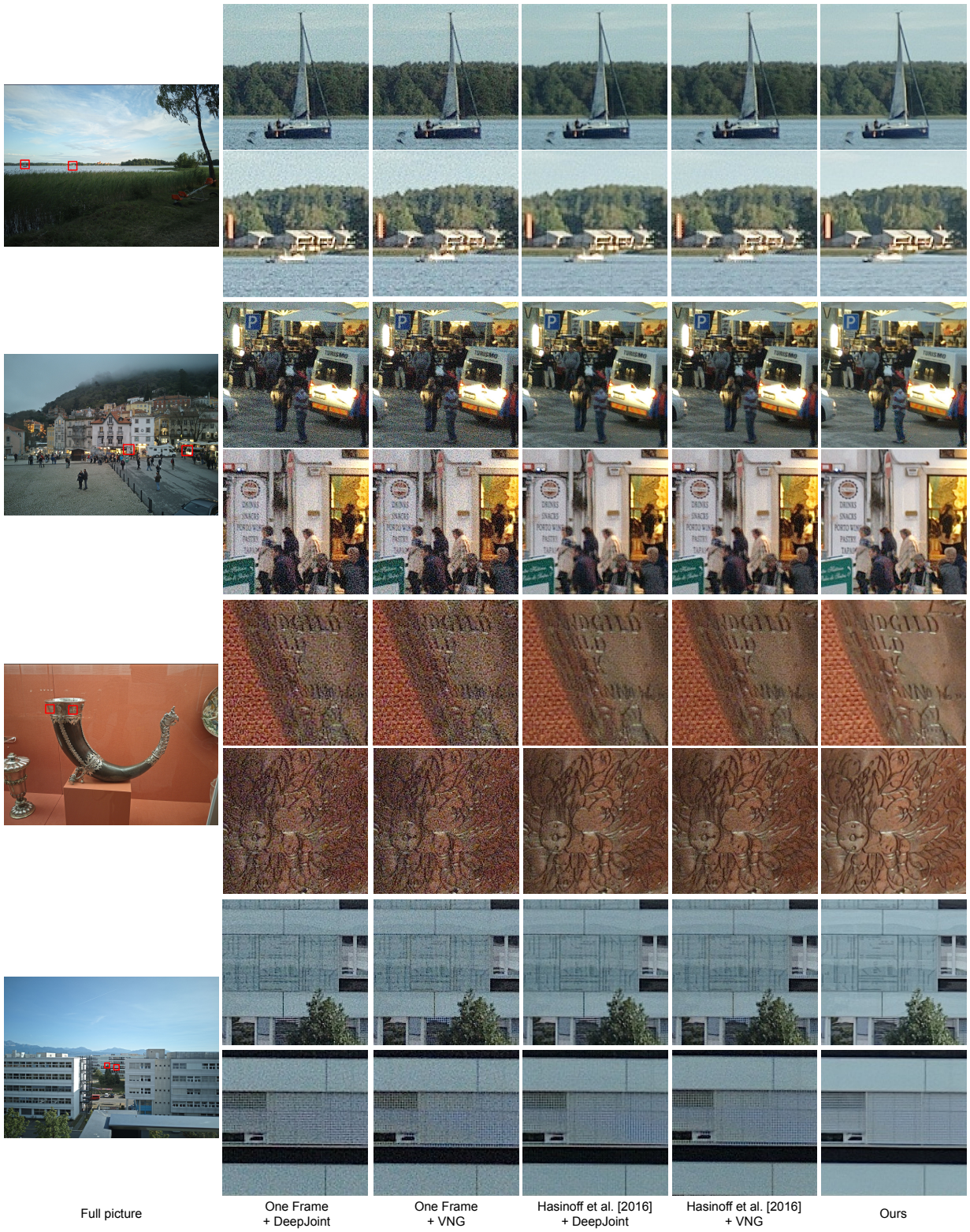

Fig. 9. **Addional comparison with demosaicing techniques:** Our method compared with dcraw's Variable Number of Gradients [Chang et al. 1999] and *DeepJoint* [Gharbi et al. 2016]. Both demosaicing techniques are applied to either one frame from a burst or result of burst merging as described in Hasinoff et al. [2016]. Readers are encouraged to zoom in aggressively (300% or more).