Multi-frame interpolation of Bayer images using optical flow

Toshiki Sato and Norishige Fukushima*
Nagoya Institute of Technology, Japan
Email: *fukushima@nitech.ac.jp

ABSTRACT

A recent image sensor in a camera contains millions of pixels. The pixel itself records only the intensity of a single light signal of red, green, or blue in the Bayer patterned sensor. When the camera moves position slightly, however, the different kinds of wavelength light signals can be recorded in a different pixel. In other words, we can complement the missing color signals in the Bayer pattern by the captured ray from the moved camera. Consequently, a full-color image can be generated without demosaicing, but the capturing method may cause a false-color. In our proposed method, we take more than 50 images to complement the pixels because the camera position is moved artificially rather than mechanically. Besides, we use the dense optical flow to track the movement of all pixels between the reference image and the other images. All images are converted from Bayer images to grayscale images to calculate the optical flow. Then, the pixels are linked according to the optical flow, the missing color pixel values in reference Bayer image are complemented from the other corresponding Bayer images. In our experiments, we generated sharper images than the demosaiced method.

Keywords: Multi-frame interpolation, Demosaicing, Optical flow

INTRODUCTION

A current camera capture an image as Bayer patterned images. Then, the Bayer image passes the image processing pipeline to obtain a color image. The image processing pipeline consists of demosaicing[1], denoising[2, 3, 10], color correction[4] and white balance correction[5], and edge enhancement. In recent years, smartphones use an image processing pipeline that uses continuous burst images to generate high-quality images[6]. In this case, the usual image processing pipeline is not efficient because we can capture various kinds of signals from the multiple captured images. The paradigm-changing affects the demosaicing and denoising in the image processing pipeline.

Therefore, in this paper, we revisit what kind of image processing pipeline is better in the image processing pipeline with burst-capturing. In particular, we examine whether combining multiple frames can perform functions equivalent to demosaicing and denoising.

PROPOSED METHOD

Interpolation using optical flow

Taking multiple images with Bayer pattern while shifting the camera slightly, different kinds of wavelength color can be probably captured at the same pixel position. In this case, the captured multiple images need to link the corresponding pixels. Usually, multi-frame based interpolation uses the corresponding points[7]. In the proposed method, we also use the point by using dense optical flow in Bayer images.

Optical flow estimation methods usually use grayscale or color image, not directly compute on Bayer image; thus, we prepare images converted to grayscale by demosaicing at first. We use edge-aware interpolation for demosaicing[8]. After determining, we select one reference image, and then we calculate the optical flow from the reference image to the other images. We regard the optical flow in the grayscale image as the flow of its Bayer image. Then, the pixels in multi-frame Bayer images are linked to a reference image by the flow for interpolation, i.e., we just copy signals from the other frames. Note that we do not use the demosaicing image, but use Bayer images. The numerical accuracy of optical flow has a sub-pixel accuracy level of about 0.25 pixels; thus, we handle the subpixel flows in the next section.

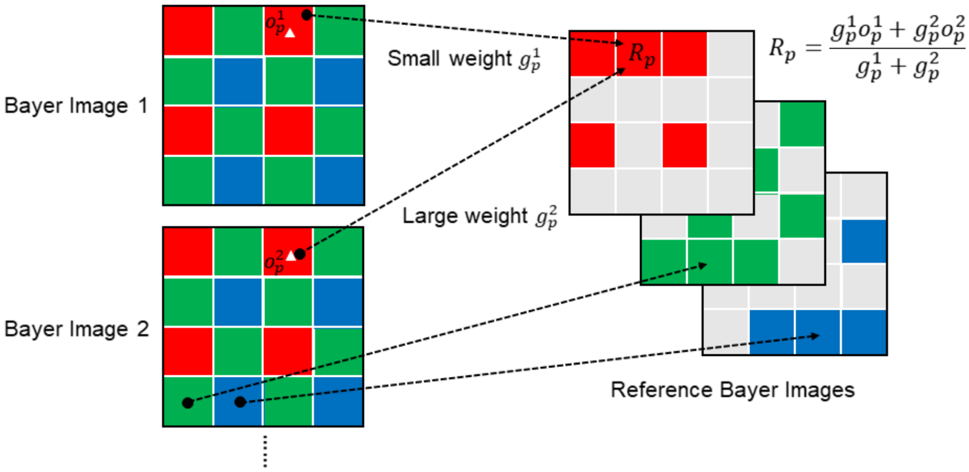

Figure 1. According to the optical flow, the reference Bayer image is complemented by the other Bayer images. The white triangle is the center of the pixel. The black circle is the destination of the optical flow from p. If multiple pixels overlap like o_p^1 and o_p^2, we use the weighting method. If the pixels do not overlap, the pixel value is taken without weighting. In this case of o_p^1, because the distance between the white triangle and the black circle is far, g_p^1 is small weight.

Weighting by the distance from the center of the nearest pixel

We perform weighted interpolation using the fractional values of the optical flow. Figure 1 shows our proposed method. When there is a corresponding pixel, we just copy the pixel. When we have two or more corresponding pixels, we perform the weighted interpolation. The weight depends on the distance to the nearest integer pixel, i.e., rounding the value of optical flow. Then the distance is weighted by Gaussian function. The weight is defined as follows;

$$d_p^i = \left\| \mathrm{R}(\boldsymbol{f}_p^i) - \boldsymbol{f}_p^i \right\|_2,$$

$$g_p^i = \exp\left(-\frac{{d_p^i}^2}{2\sigma^2}\right),$$

where p is a pixel in the reference image, $\boldsymbol{f}_p^i$ is the optical flow, which includes horizontal and vertical flows and estimated from p to the i-th Bayer image, d_p^i is the rounding distance of the optical flow, $\mathrm{R}(\cdot)$ is the function that rounds off each element. g_p^i is Gaussian weights by the d_p^i, σ is the standard deviation of the Gaussian function. Then we interpolate the pixels by using the weight. The color signals at the pixel p, R_p, G_p and B_p, are defined as follows;

$$R_p = \frac{\sum_{i=1}^{m} g_p^i\, o_p^i \mathrm{J}_r}{\sum_{i=1}^{m} g_p^i\, \mathrm{J}_r}, \qquad \mathrm{J}_r = \begin{cases} 1 & \textit{if the fetched pixel is red pixel} \\ 0 & \textit{otherwise} \end{cases}$$

$$G_p = \frac{\sum_{i=1}^{m} g_p^i\, o_p^i \mathrm{J}_g}{\sum_{i=1}^{m} g_p^i\, \mathrm{J}_g}, \qquad \mathrm{J}_g = \begin{cases} 1 & \textit{if the fetched pixel is green pixel} \\ 0 & \textit{otherwise} \end{cases}$$

$$B_p = \frac{\sum_{i=1}^{m} g_p^i\, o_p^i \mathrm{J}_b}{\sum_{i=1}^{m} g_p^i\, \mathrm{J}_b}, \qquad \mathrm{J}_b = \begin{cases} 1 & \textit{if the fetched pixel is blue pixel} \\ 0 & \textit{otherwise} \end{cases}$$

where o_p^i is the pixel value of i-th destination Bayer image referenced by the integer value of the optical flow on pixel p. m is the number of input Bayer images. J is a switching function for judging the color of the referred pixel in the Bayer pattern. After the interpolation, we can obtain the demosaicing image, and also, the image is denoised by multiple images averaging of this interpolation function.

EXPERIMENTAL RESULTS

In this experiment, we compared the edge-aware demosaiced image from a single Bayer image and the result of the proposed method. We corrected the black level and white balance in the Bayer images at first. At the end of each method, we applied gamma correction and color correction using a color correction matrix. Note that we did not apply denoising and edge enhancement in this image processing pipeline. In the proposed method, when taking multiple images, the camera fixed on a tripod was set up on the trolley, we moved the camera position in parallel with vertical and horizontal directions. A dense optical flow algorithm used in the proposed method was dense inverse search[9]. The used camera is the Nikon D7200.

Figure 3 shows the image that is applied edge-aware demosaicing from the Bayer image, and Figure 4 and Figure 5 show the images combined with 50 Bayer images. Figure 4 shows that the method uses a pixel for interpolation. The method collects the pixels closest to the destination of the optical flow, i.e., we assign the most nearest Bayer value to the integer coordinate. Figure 5 uses the proposed weighting method. As a result, the proposed method removed the noise in Figure 4 and Figure 5 compared to Figure 3. Figure 4 contains striped edges; however, Figure 5 suppresses the striped pattern because of the weighting function.

Figure 3. Result of edge-aware demosaicing of the reference image.

Figure 4. Result of the method that collects the pixels closest to the destination of the optical flow.

Figure 5. Result of the proposed weighting method.

CONCLUSION

In this paper, we proposed the more straightforward method that interpolates missing color signals by taking multiple images. We indicated that combining multiple frames can perform functions equivalent to demosaicing and denoising by using the proposed weighting method according to the distance between the destination of the optical flow and the center of the nearest pixel.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI (JP17H01764, JP18K19813).

REFERENCES

[1] Kimmel, R., "Demosaicing: Image reconstruction from color CCD samples," IEEE Transactions on Image Processing, 8(9):1221–8, Sept. 1999.

[2] Buades, A., Coll, B., and Morel, J-M., "A non-local algorithm for image denoising," in Proc. IEEE Computer Vision and Pattern Recognition (CVPR), 2005.

[3] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K., "Image denoising by sparse 3-d transform-domain collaborative filtering," IEEE Trans. on Image Processing, vol. 16, no. 8, pp. 2080–2095, 2007.

[4] Takahashi, K., Monno, Y., Tanaka, M., and Okutomi, M., "Effective color correction pipeline for a noisy image," Proc. IEEE International Conference on Image Processing (ICIP), pp. 4002-4006, 2016.

[5] Lukac, R., [Single-sensor imaging: methods and applications for digital cameras], CRC Press, 2008.

[6] Liba, O., Murthy, K., Tsai, Y-T., Brooks, T., Xue, T., Karnad, N., He, Q., Barron, J.T., Sharlet, D., Geiss, R., Hasinoff, S.W., Pritch, Y., and Levoy, M., "Handheld Mobile Photography in Very Low Light," arXiv:1910.11336, 2019.

[7] Farsiu, S., Elad, M., and Milanfar, P., "Multiframe demosaicing and super-resolution of color images," IEEE transactions on image processing 15.1 (2005): 141-159.

[8] Bradski, G. and Kaehler, A., [Learning OpenCV: Computer vision with the OpenCV library], O'Reilly Media, Inc., 2008.

[9] Kroeger, T., Timofte, R., Dai, D., and Van Gool, L., "Fast Optical Flow Using Dense Inverse Search," Proc. ECCV 9908, 471-488 (2016).

[10] Fukushima, N., Kawasaki, Y., and Maeda, Y., "Accelerating Redundant DCT Filtering for Deblurring and Denoising," in Proc. IEEE International on Image Processing (ICIP), 2019.