

# COLOR TRANSFORMATION FOR COMPRESSIVE COMPUTING IN IMAGE FILTERING

Yoshihiro Maeda<sup>†</sup>, Norishige Fukushima<sup>‡</sup>, and Takayuki Hamamoto<sup>†</sup>

<sup>†</sup> Tokyo University of Science, Japan

<sup>‡</sup> Nagoya Institute of Technology, Japan

## ABSTRACT

In this paper, we propose an approximated acceleration framework for image filtering by subsampling with color space transformation. Image subsampling then processing is a simple acceleration approach for general image processing; however, the processing loses the resulting accuracy. To keep the high accuracy, the proposed framework converts a color space of the input image, i.e., as RGB, into an appropriate color space, such as YUV, DCT, and PCA. These kinds of the color spaces gather important signals to one channel. The channel information is biased; thus, we downsample other non-important signals, and then we perform filtering for the converted full-sample and downsample signals. The experimental results show that the proposed framework achieves high accuracy and fast computational time for various image filters.

**Index Terms**— image filtering, color space transformation, image subsampling, joint filtering, acceleration

## 1. INTRODUCTION

Image filtering is one of the fundamental tools in image processing. The typical image filters are box filters [1–3], Gaussian filters [1, 4, 5], Gabor filters [6], bilateral filters [7, 8], guided image filters [9], and so on. These filters are utilized for various applications, such as image denoising [1, 10], feature detection [6, 11, 12], stereo matching [13], high dynamic range imaging [14], up-sampling/super resolution [15], and haze removing [16].

These filters are defined as 2D finite impulse response (FIR) filters. The computational order is  $O(r^2)$  per pixel, where  $r$  is the kernel radius. Several algorithms reduce the order, such as separable algorithms, recursive algorithms, and infinite impulse response (IIR) filtering algorithms. The separable algorithms decompose the filtering kernel into 1D vertical and horizontal kernels, and its computational order is  $O(r)$  [17, 18]. Also, these filters are accelerated by recursive or IIR filtering, and the computational order is  $O(1)$  [4, 5, 19].

Image downsampling [20] is also one of the traditional acceleration approaches. In this process, an input image is

downsampled, and then the downsampled image is filtered. The filtered image is upsampled at last. The approaches do not reduce the computational order *per pixel*, but reduce the number of processing pixels. The approaches accelerate image processing; however, the accuracy of the resulting image is degraded.

The technique of the image downsampling is also utilized in the image compression context. The representative usage is chroma subsampling [21]. The method subsamples the chrominance components while does not subsample the luma component in the YUV color space. The chroma subsampling is based on the knowledge that the effect of the deterioration of the chrominance components is smaller than that of the luma component for human perception. We can regard the color space transformation as color decorrelation [22]. Other color spaces, such as discrete cosine transform and principal component analysis, also perform the color decorrelation. If we downsample images in the other color spaces like the chroma subsampling, they may have similar effects as YUV.

The chroma subsampling is discussed only in image compression, but it can also accelerate image filtering. However, filtering weights, which are based on the distance in RGB color space, is invalid, such as the bilateral filters and guided image filters. The reason is that the distance of the YUV color space is different from that of the RGB color space.

In this paper, we propose an approximated acceleration framework for image filtering by utilized color space transformation. We introduce concepts of the chroma subsampling, which are to concentrate important signals on one channel and subsample unimportant signals, to the proposed framework. Thus, the proposed framework can accelerate various image filtering while keeping accuracy. The proposed framework employs joint filtering for obtaining filtering weights correctly. By joint filtering, the proposed framework computes the weights based on the RGB color space. The advantage of the proposed framework is that the proposed framework and various acceleration filtering algorithms can be used in combination.

## 2. IMAGE SUBSAMPLING

The acceleration with image downsampling is defined as:

$$\bar{I} = (f(I_{\downarrow}) * I_{\downarrow})_{\uparrow}, \quad (1)$$

This work was supported by JSPS KAKENHI (JP21K17768, JP19K24368, and JP21H03465).

where  $I$  and  $\bar{I}$  are an input image and output image, respectively.  $f(\cdot)$  is the filtering weights function.  $\downarrow$  and  $\uparrow$  represent downsampling and upsampling operators, respectively.  $*$  is a convolution operator. The approach accelerates image filtering, but loses the accuracy.

Chroma subsampling [21] is a similar approach to the image downsampling. The method downsamples only the chrominance components in YUV color space images. The chroma subsampling needs to convert images into YUV images. For example, YUV4:2:2 does not downsample Y component, and subsamples U and V components in half. The technique is based on that human perception is more sensitive to the degradation of luminance than that of color; thus the method is utilized in the context of the bit-rate reduction.

The chroma subsampling reduces the cost of processing as a side effect because the number of the processing pixels is reduced. We aggressively utilize the characteristics for acceleration of image processing with various kinds of color space transformation.

### 3. COLOR SPACE TRANSFORMATION

#### 3.1. YUV Color Space

The YUV color space has three components, which are luma component (Y) and chrominance components (U and V). The Y signals mean luminance and are based on colorimetric considerations. The U and V signals are calculated as differences between Y signals and the B and G values, respectively. The RGB to YUV conversion is defined as follows:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.500 & -0.419 & -0.0813 \\ -0.169 & -0.331 & 0.500 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (2)$$

The  $3 \times 3$  matrix is a transformation matrix, which is defined in ITU-R BT. 601 [21]. The transformation matrix is a non-orthogonal matrix. Therefore, the inverse transformation is achieved by using the inverse matrix. The process is the following:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 1.407 & 0.000 \\ 1.000 & -0.714 & -0.344 \\ 1.000 & 0.000 & 1.772 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix}. \quad (3)$$

#### 3.2. 3-Point DCT

Discrete cosine transform (DCT) is one of the frequency transformation. The DCT exploits correlation in spatial and color dimensions. The DCT is used in frequency domain filtering [1, 23] and image compression [24]. The color decorrelation can also be achieved by performing DCT on the color space [22, 23]. The RGB to DCT transformation is defined as follows:

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (4)$$

The transformation matrix has a 3-point DCT basis. The inverse transformation is achieved by using the transpose operation of the matrix because the DCT transform matrix is orthogonal. The process is defined as:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}. \quad (5)$$

#### 3.3. Principal Component Analysis

Principal component analysis (PCA) converts possibly correlated variables into uncorrelated variables by an orthogonal matrix. The uncorrelated variables are called principal components. The PCA is a technique of the decorrelation and dimensionality reduction [22]. The first component represents the variability of the data to the maximum extent possible. The other components explain the remaining variability.

The PCA is obtained by decomposition of a covariance matrix. Let pixels of an input image be  $\mathbf{X} = [x_1, x_2, x_3]^T$ , where  $x_1$ ,  $x_2$ , and  $x_3$  are values of a pixel in each RGB channels, and  $T$  denotes a transpose operator. The covariance matrix  $\mathbf{C}$  is defined as:

$$\mathbf{C} = \frac{1}{N} \sum_{k=1}^N \mathbf{X}_k \mathbf{X}_k^T - \mathbf{m} \mathbf{m}^T, \quad (6)$$

where  $N$  is the number of pixels.  $\mathbf{m}$  is the mean vector, which is defined as follows:

$$\mathbf{m} = \frac{1}{N} \sum_{k=1}^N \mathbf{X}_k. \quad (7)$$

The transformation matrix  $\mathbf{A}$  is corresponding to eigenvectors of  $\mathbf{C}$ :

$$\mathbf{A} = [ \mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3 ]^T, \quad (8)$$

where  $\mathbf{e}_1$ ,  $\mathbf{e}_2$ , and  $\mathbf{e}_3$  are eigenvectors. The their eigenvalues,  $\lambda_i$ , have descending order so that  $\lambda_i > \lambda_{i+1}$ . The projection from the RGB space to the PCA space is obtained by:

$$\mathbf{P} = \mathbf{A} \mathbf{X}, \quad (9)$$

and the inverse projection is defined as:

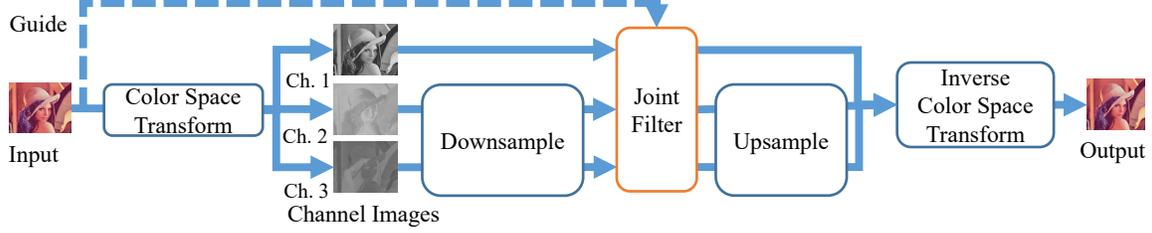
$$\mathbf{X} = \mathbf{A}^T \mathbf{P}. \quad (10)$$

### 4. PROPOSED FRAMEWORK

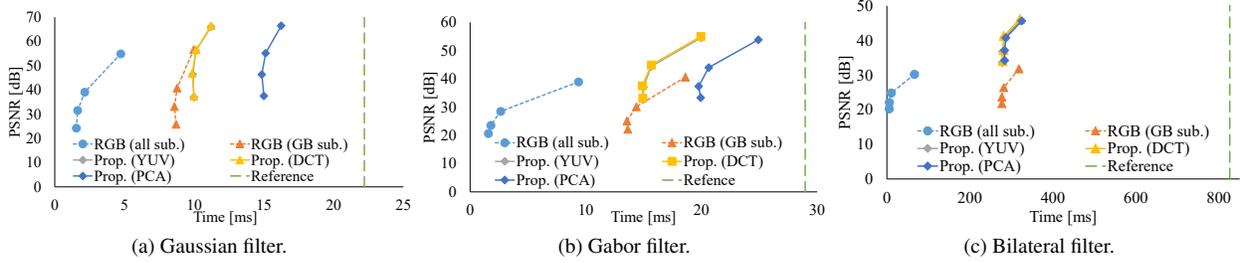
In this section, we propose an approximated acceleration framework for image filtering by utilizing color space transformation. Figure 1 shows the overview of the proposed framework. Firstly, the proposed framework converts the RGB color space into an appropriate color space. The color space transform is defined by:

$$\mathbf{J} = \phi(\mathbf{I}), \quad (11)$$

where  $\mathbf{I}$  is an input RGB image.  $\mathbf{J}$  is a color space converted image and a set of channel images as  $\mathbf{J} = \{\mathbf{J}_1, \mathbf{J}_2, \mathbf{J}_3\}$ . Note



**Fig. 1:** The overview of the proposed framework.



**Fig. 2:** PSNR w.r.t computational time. The kernel radius  $r$  is 32. Parameter of Gaussian filter is  $\sigma = r/3$ . Those of the Gabor filter are  $\sigma = 5$ ,  $\theta = 10$ ,  $\lambda = 30$ ,  $\gamma = 10$ , and  $\psi = 15$ . Those of the bilateral filter are  $\sigma_s = r/3$  and  $\sigma_r = 20$ .

that  $\mathbf{J}_i$  has more essential signals than  $\mathbf{J}_{i+1}$ .  $\phi(\cdot)$  denotes a function of color space transformation.

Secondly, we perform filtering for  $\mathbf{J}$ :

$$\bar{\mathbf{J}} = \begin{cases} f(\mathbf{I}) * \mathbf{J}_n & n = 1 \\ (f(\mathbf{I}_\downarrow) * \mathbf{J}_{n\downarrow})_\uparrow & n = 2, 3 \end{cases}, \quad (12)$$

where  $\bar{\mathbf{J}}$  is a filtered image. In the filtering processing,  $\mathbf{J}_1$  is not downsampled since  $\mathbf{J}_1$  is essential. On the other hand,  $\mathbf{J}_2$  and  $\mathbf{J}_3$  are downsampled.

When filtering weights are changed by image intensities  $\mathbf{I}$ , such as bilateral filtering, we cannot filter the converted image  $\mathbf{J}$ , directly. To consider the weight from  $\mathbf{I}$ , we should filter image as joint filtering [25,26]. If filtering weights are calculated based on the converted image, they are not correct in several filters, which have weights depending on pixel values in RGB color space. Therefore, in the proposed framework, the weights are calculated from the RGB image by using the joint filters.

Finally,  $\bar{\mathbf{J}}$  is converted back to the RGB color space:

$$\bar{\mathbf{I}} = \phi^{-1}(\bar{\mathbf{J}}), \quad (13)$$

where  $\bar{\mathbf{I}}$  is an output image and  $\phi^{-1}(\cdot)$  denotes a function of inverse color transformation.

The proposed framework accelerates image filtering by image subsampling and achieves high accuracy by the color space transformation. The image filtering is performed on the essential channel without downsampling, and the other channels are filtered after downsampling. Moreover, the proposed framework solves the problem of weight computation by using joint filtering. Therefore, we can deal with various image filters and accelerate the filters while keeping accuracy.

In addition, the proposed framework does not change filtering algorithms itself. Therefore, we can also utilize special-

ized acceleration algorithms for each channel filtering, such as FFT and recursive acceleration.

## 5. EXPERIMENTAL RESULTS

We compared the proposed acceleration framework with the simple image subsampling approach and the proposed framework without color space transformation, which downsamples R and G channels, as competitive methods. Also, we evaluated three filters, such as Gaussian filtering, Gabor filtering, and bilateral filtering. The weight of the Gaussian filtering is defined as:

$$f(\mathbf{p}, \mathbf{q}) := \exp\left(\frac{\|\mathbf{p} - \mathbf{q}\|_2^2}{-2\sigma^2}\right), \quad (14)$$

where  $\mathbf{p}$  and  $\mathbf{q}$  are current and reference position, respectively.  $\|\cdot\|_2$  denotes the L2 norm.  $\sigma$  is a smoothing parameter. The weight of the Gabor filtering is defined as:

$$f(x, y) := \exp\left(\frac{x'^2 + \gamma^2 y'^2}{-2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right), \quad (15)$$

where  $x$  and  $y$  are horizontal and vertical parts of  $\mathbf{p} - \mathbf{q}$ , respectively, and  $x' := x \cos \theta + y \sin \theta$  and  $y' := -x \sin \theta + y \cos \theta$ .  $\lambda$  denotes the wavelength.  $\theta$  denotes the orientation of the normal to the parallel stripes.  $\psi$  and  $\gamma$  are the phase offset and the spatial aspect ratio, respectively. The weight of the bilateral filtering is defined as:

$$f(\mathbf{p}, \mathbf{q}) := \exp\left(\frac{\|\mathbf{p} - \mathbf{q}\|_2^2}{-2\sigma_s^2}\right) \exp\left(\frac{\|\mathbf{I}(\mathbf{p}) - \mathbf{I}(\mathbf{q})\|_2^2}{-2\sigma_r^2}\right), \quad (16)$$

where  $\sigma_s$  and  $\sigma_r$  are smoothing parameters for space and color, respectively. The weights of the Gaussian filtering and Gabor filtering depend on only spatial distance; thus, the weights are invariant on all pixels. That of the bilateral filtering depend on spatial and color distance; thus, the weights

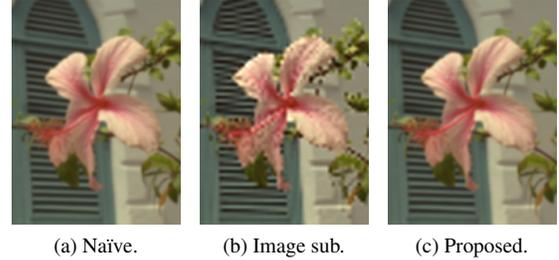
**Table 1:** PSNR [dB] of the Gabor filtering. Parameters are  $\sigma = 5$ ,  $\theta = 10$ ,  $\lambda = 30$ ,  $\gamma = 10$ , and  $\psi = 15$ . The downsampling ratio is  $1/16$ .

	conventional		propose		
	image sub.	RG sub.	YUV	DCT	PCA
kodim01	24.48	26.24	44.72	<b>45.11</b>	41.57
kodim02	31.50	32.94	39.31	39.33	<b>39.53</b>
kodim03	31.86	33.19	43.99	44.54	<b>44.63</b>
kodim04	31.16	32.92	43.53	<b>43.67</b>	41.44
kodim05	24.78	26.35	40.85	41.43	<b>41.54</b>
kodim06	26.79	28.31	45.49	<b>46.06</b>	44.55
kodim07	29.21	30.93	44.16	<b>44.79</b>	44.60
kodim08	22.75	24.41	43.73	<b>44.15</b>	43.30
kodim09	30.23	31.82	46.75	<b>47.41</b>	45.93
kodim10	30.59	32.26	47.17	<b>47.75</b>	46.87
kodim11	28.14	29.71	45.49	<b>45.77</b>	44.05
kodim12	29.95	31.53	46.82	<b>47.48</b>	46.19
kodim13	23.81	25.48	43.53	<b>44.25</b>	43.96
kodim14	27.77	29.23	39.41	<b>39.72</b>	38.59
kodim15	30.53	32.17	43.99	<b>44.33</b>	43.62
kodim16	30.71	32.28	49.65	50.16	<b>50.28</b>
kodim17	30.78	32.51	48.46	49.00	<b>49.20</b>
kodim18	27.69	29.30	43.40	<b>43.95</b>	39.94
kodim19	26.85	28.37	47.22	47.72	<b>48.15</b>
kodim20	29.65	31.22	45.41	<b>45.97</b>	45.70
kodim21	27.56	29.17	46.15	<b>46.64</b>	44.39
kodim22	29.63	31.31	43.06	<b>43.60</b>	43.13
kodim23	31.36	33.00	43.04	<b>43.49</b>	43.47
kodim24	26.48	28.45	41.49	<b>41.99</b>	41.96

depend on values of RGB pixels. Therefore, the bilateral filtering requires joint filtering, and the Gaussian filtering and Gabor filtering do not require. Note that the bilateral filtering was applied to each channel image because we assumed that each channel images does not correlate. The proposed framework utilized three kinds of color spaces, such as YUV, DCT, and PCA. The approximation accuracy was evaluated with the peak signal noise ratio (PSNR) between the approximation and naïve algorithm. Filtering radius was  $r = 32$ . For evaluation, we used 24 images ( $768 \times 512$ ), which are Kodak images. We used Intel Core i7-7800X 3.5GHz (6 cores, 12 threads) and Visual Studio 2017 as a compiler.

Figure 2 shows the relation between PSNR and computational time with changing the downsampling ratio. The proposed framework accelerates these filters while keeping high accuracy. The proposed framework is the highest accuracy than the conventional methods of image subsampling and RG subsampling. For the Gabor filtering and the bilateral filtering, the proposed framework is more effective. The weight of the Gaussian filtering is simple, but those of the Gabor filtering and the bilateral filtering are complex.

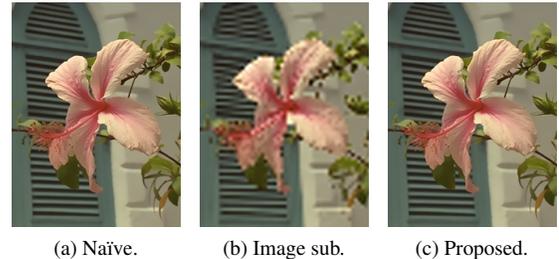
In comparison with the color spaces, the accuracy of the DCT is the highest among the three kinds of color space, but these are almost the same accuracy. The DCT and the YUV are almost the same computational time, and the PCA is the slowest among them. The PCA requires to calculate the mean and covariance of the input image; thus, the transform matrix depends on the image. On the other hand, the transform matrices of DCT and YUV are invariant regardless of the image. Table 1 shows that PSNR of the Gabor filtering. In most cases, the accuracy of the DCT is the best. However, in a



**Fig. 3:** Result images of Gaussian filtering.



**Fig. 4:** Result images of Gabor filtering.



**Fig. 5:** Result images of bilateral filtering.

small number of images, the PCA is more accurate. Therefore, the PCA is not necessarily ineffective.

Figures 3 - 5 depict filtering results of an image. In visually, the proposed framework can keep high approximate accuracy.

## 6. CONCLUSION

In this paper, we proposed an acceleration framework for image filtering by image subsampling. The proposed framework uses characteristics of the color decorrelation to downsample channel images, which do not have essential signals. The joint filtering is used to correctly calculate weights in transformed images, which is different from RGB color space. The experimental results showed that the proposed framework accelerates the Gaussian filtering, the Gabor filtering, and the bilateral filtering while keeping high approximate accuracy.

Also, the proposed framework does not change algorithms of image processing itself. If we construct the proposed method as a class library or processing framework, we can accelerate various image processing with slight code modification, because we can directly use optimized code of each image processing. In our experiment, we use the optimized code based on OpenCV 4.0 without modification, actually.

## 7. REFERENCES

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, 2008.
- [2] F. C. Crow, “Summed-area tables for texture mapping,” in *Proc. ACM SIGGRAPH*, 1984, pp. 207–212.
- [3] N. Fukushima, Y. Maeda, Y. Kawasaki, M. Nakamura, T. Tsumura, K. Sugimoto, and S. Kamata, “Efficient computational scheduling of box and gaussian fir filtering for cpu microarchitecture,” in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2018.
- [4] P. Getreuer, “A survey of gaussian convolution algorithms,” *Image Processing On Line*, vol. 2013, pp. 286–310, 2013.
- [5] I. T. Young and L. J. van Vliet, “Recursive implementation of the gaussian filter,” *Signal processing*, vol. 44, no. 2, pp. 139–151, 1995.
- [6] C. Liu and H. Wechsler, “Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition,” *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 467–476, 2002.
- [7] P. Kornprobst and J. Tumblin, *Bilateral filtering: Theory and applications*, Now Publishers Inc., 2009.
- [8] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 1998, pp. 839–846.
- [9] K. He, J. Sun, and X. Tang, “Guided image filtering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.
- [10] A. Buades, B. Coll, and J. M. Morel, “A non-local algorithm for image denoising,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 60–65.
- [11] J. S. Chen, A. Huertas, and G. Medioni, “Fast convolution with laplacian-of-gaussian masks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 4, pp. 584–590, 1987.
- [12] M. Basu, “Gaussian-based edge-detection methods—a survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 32, no. 3, pp. 252–260, Aug 2002.
- [13] T. Matsuo, N. Fukushima, and Y. Ishibashi, “Weighted joint bilateral filter with slope depth compensation filter for depth map refinement,” in *Proc. International Conference on Computer Vision Theory and Applications (VISAPP)*, 2013.
- [14] F. Durand and J. Dorsey, “Fast bilateral filtering for the display of high-dynamic-range images,” *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 257–266, 2002.
- [15] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, “Joint bilateral upsampling,” *ACM Transactions on Graphics*, vol. 26, no. 3, 2007.
- [16] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 2341–2353.
- [17] T. Q. Pham and L. J. V. Vliet, “Separable bilateral filtering for fast video preprocessing,” in *Proc. International Conference on Multimedia and Expo (ICME)*, 2005.
- [18] N. Fukushima, S. Fujita, and Y. Ishibashi, “Switching dual kernels for separable edge-preserving filtering,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.
- [19] W. M. Wells, “Efficient synthesis of gaussian filters by cascaded uniform filters,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 2, pp. 234–239, 1986.
- [20] L. Lou, P. Nguyen, J. Lawrence, and C. Barnes, “Image perforation: Automatically accelerating image pipelines by intelligently skipping samples,” *ACM Transactions on Graphics*, vol. 35, no. 5, pp. 153:1–153:14, 2016.
- [21] ITU-R BT.601-7, “Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios,” 2011.
- [22] M. S. Drew and S. Bergner, “Spatio-chromatic decorrelation for color image compression,” *Signal Processing: Image Communication*, vol. 23, no. 8, pp. 599–609, 2008.
- [23] G. Yu and G.apiro, “DCT Image Denoising: a Simple and Effective Image Denoising Algorithm,” *Image Processing On Line*, vol. 1, pp. 292–296, 2011.
- [24] G. K. Wallace, “The jpeg still picture compression standard,” *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.
- [25] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen, and K. Toyama, “Digital photography with flash and no-flash image pairs,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 664–672, 2004.
- [26] E. Eisemann and F. Durand, “Flash photography enhancement via intrinsic relighting,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 673–678, 2004.