Decolorization: Is `rgb2gray()` Out?

Yibing Song Linchao Bao Xiaobin Xu Qingxiong Yang
City University of Hong Kong

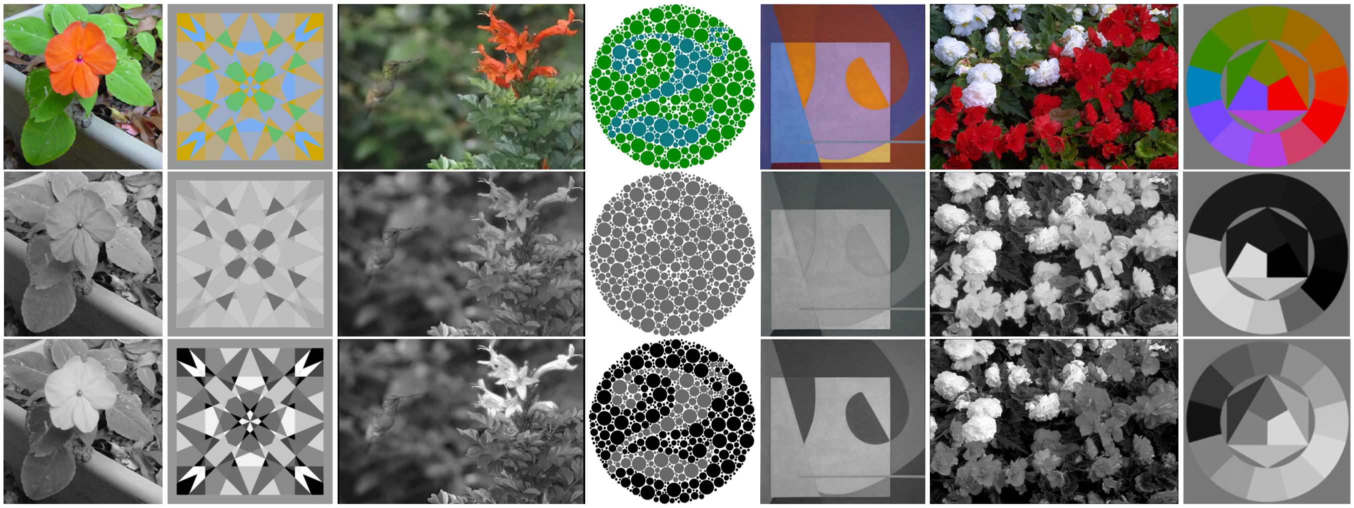


Figure 1: *First row: original color images. Second row: the failure cases of existing decolorization methods. Results are obtained by [Gooch et al. 2005], [Grundland and Dodgson 2007], [Smith et al. 2008], [Kim et al. 2009], [Ancuti et al. 2011], [Lu et al. 2012a] and [Lu et al. 2012b], respectively (from left to right). Third row: results of modified* `rgb2gray()` *with adjusted weights for R, G, and B channels.*

Abstract

Decolorization problems originate from the fact that the luminance channel may fail to represent iso-luminant regions in the original color image. Currently all the existing methods suffer from the same weakness – robustness: failure cases can be easily found for each of the methods. This prevents all these methods from being practical for real-world applications. In fact, the luminance conversion (i.e, `rgb2gray()` function in Matlab) performs rather well in practice only with exceptions for failure cases like the iso-luminant regions. Thus a thought-provoking question is naturally raised: can we reach a robust solution by simply modifying the `rgb2gray()` to avoid failures in iso-luminant regions? Instead of assigning fixed channel weights for all images, a more flexible strategy would be choosing channel weights depending on specific images to avoid indiscrimination in iso-luminant regions. Following this strategy, by considering multi-scale contrast preservation, we design an algorithm that can consistently produce "good" results for each color image, among which the "best" one preferred by users can be selected by further involving perceptual contrasts preferences. The results are verified through user study.

CR Categories: I.4.3 [Image Processing and Computer Vision]: Enhancement—Grayscale Manipulation;

Keywords: Decolorization, Visual Perception, Bilateral Filter

1 Introduction

Decolorization, a seemingly simple problem which aims to convert color images into grayscale images while preserving structures and contrasts in original color images, has recently received great attention in both graphics and vision society. Theoretically speaking, it is essentially a dimensionality reduction problem and hence is difficult. The baseline method to convert a color image into grayscale image is to extract its *luminance* channel[1] (e.g., CIE Y). If the color image is represented in RGB format, the luminance can be simply computed via a linear combination of R, G, and B channels with fixed weight (e.g., the `rgb2gray()` function in Matlab). For images having iso-luminant regions, the luminance channel will fail to represent structures or features in the color image, since the linear combination using fixed weights can produce the same result for some different groups of R, G, and B values.

Various of techniques categorized as local and global methods have been employed to better the baseline method. Local methods [Bala and Eschbach 2004; Gooch et al. 2005; Smith et al. 2008] alleviate the dimensionality reduction problem by employing different mapping functions in different local regions in one image, while global methods [Grundland and Dodgson 2007; Kim et al. 2009; Song et al. 2010; Ancuti et al. 2011; Lu et al. 2012a] strive to produce one mapping function for the whole image. Considering that local methods might cause unpleasant halo artifacts [Kim et al. 2009], global methods are more preferred in recent research work [Ancuti et al. 2011; Lu et al. 2012a]. In spite of the efforts of involving more complicated color models and computational models, all of the existing methods suffer from the same weakness – robustness: failure cases can be easily found for each of the methods from images in our daily life, either of missing major structures in original

[1] In the literature, *lightness* channel (e.g., L channel in CIELab color system) may also be regarded as the baseline method of decolorization.

color images or losing the perceptual plausibility. This prevents all these methods from being practical for real-world applications.

With the reflection on the current trend of involving more complicated color models (e.g., the nonlinear color model in [Kim et al. 2009] and the polynomial color model in [Lu et al. 2012a]) and computational models (e.g., the probabilistic graphical model in [Song et al. 2010] and nonlinear system model in [Lu et al. 2012a]) to solve the problem, a thought-provoking question will be naturally raised: can we reach a robust solution using the simplest color model and the most straightforward computational model? Recent work of [Lu et al. 2012b] gives us a positive answer along this line. Specifically, they approximate their previous optimization-based method [Lu et al. 2012a] and achieve real-time performance by confining the polynomial color model into a constrained, discrete linear color model. However, since their objective function is originally defined over continuous, polynomial space [Lu et al. 2012a], the approximated solution in confined search space might produce unsatisfactory results in special cases (see Figure 1). Nevertheless, the work shows the potential of the simplest conversion model, just like what is used in the classical Matlab function `rgb2gray()`, which we refer to as the RGB2GRAY *conversion model*:

Definition (RGB2GRAY *conversion model*) The grayscale output g is a constrained linear combination of R, G, and B channels of the input color image I, which is

$$
\begin{aligned}
g &= w_r I_r + w_g I_g + w_b I_b && (1)\\
\text{s.t.} \quad & w_r + w_g + w_b = 1, && (2)\\
& w_r \geq 0, w_g \geq 0, w_b \geq 0, && (3)
\end{aligned}
$$

where I_r, I_g, and I_b are input channels, respectively. Channel weights w_r, w_g, and w_b are non-negative numbers that sum to 1.

In the classical Matlab function `rgb2gray()`, the weights are fixed as $\{w_r = 0.2989, w_g = 0.5870, w_b = 0.1140\}$ for all images. A more flexible strategy would be choosing channel weights w_r, w_g, and w_b depending on specific input images. We in this paper show that high-quality results can be consistently found using this strategy with a straightforward computational framework for contrast preservation.

The major contributions of this paper are as follows. **First**, we design a novel decolorization algorithm that can take into account *multi-scale* contrast preservation in both *spatial* and *range* domain. **Second**, we conduct a user study on a commonly adopted decolorization dataset [Cadik 2008] to show that user-preferred "best" results among all the candidates produced by the (quantized) RGB2GRAY conversion model can be identified by our algorithm. Note that our algorithm produces several "good" results for each image, among which the actual "best" one can be selected by further involving perceptual preferences depending on specific applications. **Third**, our study shows the potential of the RGB2GRAY conversion model and provides the "best" results that can be obtained using this model, which can serve as the "ground truth" results of this model on Cadik's dataset.

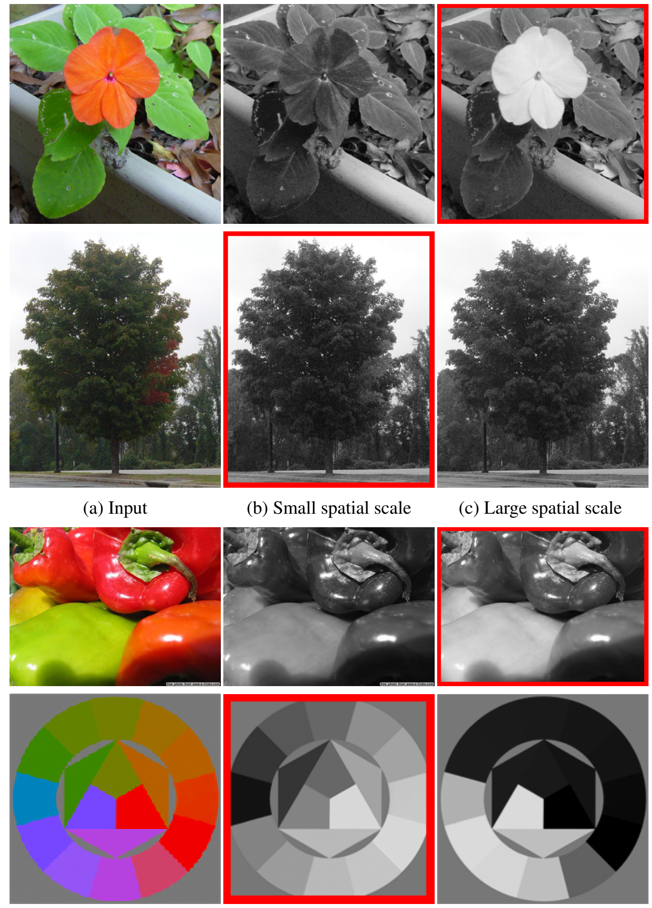
2 Our Approach

In this section, we begin with describing the motivation of our approach, followed by introducing the key tools and strategies employed in our approach. Finally, we summarize our approach in the end of this section.

2.1 Multi-Scale Contrast Preservation

In the decolorization process, *contrast preservation* is often regarded as the key ingredient to avoid indiscrimination between different colors [Gooch et al. 2005; Kim et al. 2009; Lu et al. 2012a]. The motivation of our approach stems from the observation on human perceptual preferences of contrasts (through user study, see Section 3): when evaluating decolorization results, users tend to pay more attention on contrast preservation of different *spatial* and *range*[2] scales for different images, depending on the image contents. For example, in the first row of Figure 2, by preserving contrasts in small-scale, local regions, the details of the flower petals are well preserved in (b), but the contrast between red flower and green leaves are lost. By preserving larger-scale contrasts in the whole image, the red flower becomes prominent in the grayscale image in (c), which is the user-preferred result. However, in the second row of Figure 2, the user-preferred result is the one in (b) that can preserve small-scale contrast, since the small regions of red leaves will get lost when larger-scale contrasts are targeted to be preserved (see (c)).

When it comes to the range domain, the diversity of user preferences remains true. The last two rows of Figure 2 shows two examples. In the "peppers" example (third row), when contrast preserva-



(a) Input (b) Small spatial scale (c) Large spatial scale

(d) Input (e) Small range scale (f) Large range scale

Figure 2: *Examples of multi-scale contrast preservation in spatial and range domain. User-preferred results are marked inside red squares. First two rows: contrasts of small spatial scale are preserved in (b), while contrasts of large spatial scale are preserved in (c). Last two rows: similar to the above rows but shows the contrast preservation for different range scales.*

[2]The *range* domain means the image color/intensity domain, as is usually referred to in literature of bilateral filtering [Paris et al. 2009].

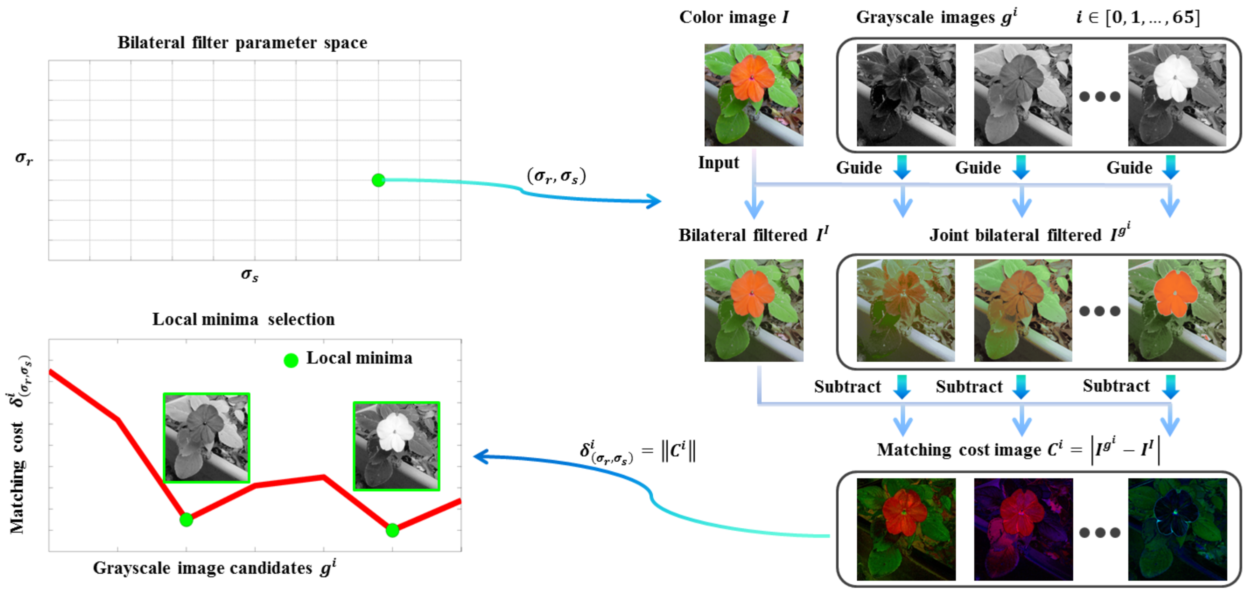


Figure 3: *Overview of our approach. For a given parameter setting (σ_s, σ_r), cost δ^i for each grayscale image g^i in candidate set is computed, and candidate with a local minimum cost value is voted once (local minima is selected by comparing each candidate to its neighboring candidates). After processing all (σ_s, σ_r) parameter settings, candidates with more votes than a threshold are selected as output.*

tion of small range scale is enhanced (see Figure 2(e)), small color variations within one pepper are well preserved, but the contrasts between different peppers get weakened. According to our user study (see Section 3), users actually prefer large contrasts between peppers with different colors, as is shown in Figure 2(f), where only large color contrasts are targeted to be preserved (i.e., contrast preservation of large range scale). An opposite example is shown in the last row of Figure 2, where the preservation of small color variations are preferred by users (contrast preservation of small range scale).

As analyzed above, the diversity of user preferences in the contrast preservation in both spatial and range domain makes the decolorization difficult to consistently produce high-quality results. By exploring multi-scale contrast preservation, our approach alleviates such difficulty.

2.2 Bilateral Filtering for Contrast Preservation

We use bilateral filtering [Yang et al. 2009] to capture the center-surround contrast for each pixel in an image. Note that other fast edge-preserving filtering algorithms [Gastal and Oliveira 2011] can also be employed for speed concern, but we here use bilateral filtering for its simplicity and directness. The (joint) bilateral filtering is defined as follows. Let $\mathbf{I}(\mathbf{p})$ be the color at pixel $\mathbf{p}$ and $\mathbf{I}^{\mathbf{J}}(\mathbf{p})$ be the filtered value, then we have

$$\mathbf{I}^{\mathbf{J}}(\mathbf{p}) = \frac{\sum_{\mathbf{q}\in\Omega_{\mathbf{p}}} G_{\sigma_s}(||\mathbf{p}-\mathbf{q}||) G_{\sigma_r}(||\mathbf{J}(\mathbf{p})-\mathbf{J}(\mathbf{q})||)\mathbf{I}(\mathbf{q})}{\sum_{\mathbf{q}\in\Omega_{\mathbf{p}}} G_{\sigma_s}(||\mathbf{p}-\mathbf{q}||) G_{\sigma_r}(||\mathbf{J}(\mathbf{p})-\mathbf{J}(\mathbf{q})||)}, \tag{4}$$

where $\mathbf{q}$ is a pixel in the neighborhood $\Omega_{\mathbf{p}}$ of pixel $\mathbf{p}$, and G_{σ_s} and G_{σ_r} are the spatial and range filter kernels measuring the spatial and color/intensity similarity, $\mathbf{J}$ is the guidance image (which can be either the input image $\mathbf{I}$ itself or other images).

Given an input color image $\mathbf{I}$ and a grayscale conversion result g, we perform bilateral filtering on $\mathbf{I}$ with itself and g as guidance image, respectively, to get $\mathbf{I}^{\mathbf{I}}$ and $\mathbf{I}^g$. Ideally if all the details in the color image can be reproduced in the grayscale image, the bilateral filtered results $\mathbf{I}^{\mathbf{I}}$ and $\mathbf{I}^g$ should be identical. However, this will not be the case in reality, since the dimensionaslity reduction process probably will cause contrast loss for most images. Nevertheless, the matching cost between the two results can serve as a good metric to measure the contrast preservation quality of the grayscale conversion. Specifically, we compute the matching cost image $\mathbf{C}$ as follows,

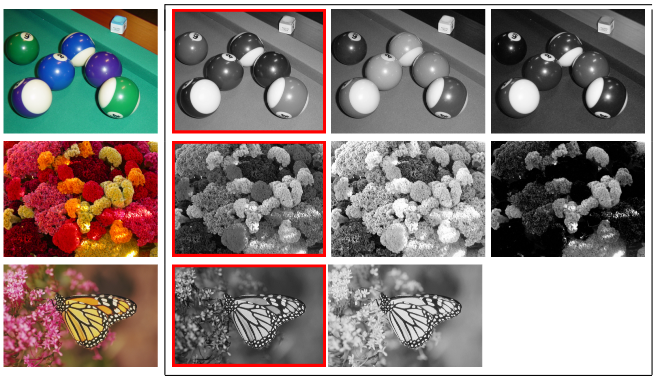
$$\mathbf{C} = |\mathbf{I}^g - \mathbf{I}^{\mathbf{I}}|. \tag{5}$$

Summing up all the pixels of all channels in the matching cost image yields the cost δ, which we adopt as a metric to measure the contrast preservation quality of a grayscale conversion (the lower, the better).

Note that the filtering is performed with two specific parameters: σ_s and σ_r. Thus the metric δ can only be used to measure the contrast preservation in one specific spatial and range scale. Specifically, it captures the contrasts within a small spatial neighborhood for each pixel when σ_s is small, while it can take into account larger neighborhood when σ_s becomes larger. Similarly, when σ_r is small, it favors grayscale image that can capture all small color variations in color image; and when σ_r becomes larger, it becomes more tolerant on small color variations. Jointly considering multiple scales in both domain can actually simulate human preferences. Next we describe our strategy for taking into account multi-scale contrast preservation.

2.3 Local Minima Voting

We use the quantized RGB2GRAY conversion model to generate grayscale image candidates. Following [Lu et al. 2012b], we discretize w_r,w_g,w_b in the range of $[0, 1]$ with interval 0.1. This yields a candidate set containing 66 grayscale images for each input color image. The candidates are actually uniformly sampled from a triangular plane in the w_r-w_g-w_b space (see the constraint in Equations (2) and (3)). Note that finer quantization is unnecessary for most images since it can only produce results with almost invisible differences from these 66 candidates.



(a) Input (b) Our results

Figure 4: *Results of our approach on Cadik's dataset. The user-preferred results are successfully identified by our approach (marked with red squares). Note that our approach produces several results for each input image, among which the ones with top three (if there is) most votes are shown.*

The pipeline of our approach is shown in Figure 3. First, we quantize the σ_s-σ_r parameter space of bilateral filtering. For a given parameter setting (σ_s, σ_r), we compute cost δ^i for each grayscale image g^i in the candidate set using the above described method, then the candidates with local minimum cost values are voted. Here the local minima is selected by comparing the cost of each candidate to its neighboring candidates (see Figure 3 for a 1-D illustration of local minima selection, note the selection is on a 2-D triangle plane in a 3-D w_r-w_g-w_b space). After processing all (σ_s, σ_r) parameter settings, we count the votes of each candidate and select the candidates with more votes than a threshold as the output.

3 User Study and Experiments

To obtain human preferences on grayscale conversion and verify our approach, we conduct a user study on Cadik's decolorization dataset [Cadik 2008] which contains 24 color images. We restrict the candidates for each color image to the 66 grayscale images generated by the quantized RGB2GRAY conversion model. To reduce the efforts of the observers, we manually remove some seemingly similar candidates to reduce obvious ambiguity. We adopt a similar perceptual evaluation setting as described in [Cadik 2008]: each time two grayscale candidates are randomly displayed along with the input color image in a high resolution display. Observers are asked to choose one of the two candidates that better matches the color image from their own preference. In our study, 20 observers participated and a total of around 7500 pair-wise comparisons are completed. Finally, for each color image, the grayscale candidate with the largest number of votes from observers is selected as the "best" user-preferred result.

In our algorithmic experiments, we quantize the parameters σ_s as $\{0.1, 0.2, 0.3, ..., 1.0\}$ and σ_r as $\{0.01, 0.05, 0.1, 0.2\}$. After running our algorithm on Cadik's dataset, we collect all the grayscale results with nonzero votes for each color image. For most of the color images in the dataset, our algorithm produces less than 10 results (out of 66 candidates). Actually, for some of the images, the results only contain 2-3 candidates (e.g., the "butterfly" image shown in Figure 4 only has two candidates left). Most importantly, all of the user-preferred results are contained in our results, which indicates the robustness of the multi-scale contrast preservation simulating the human perception.

4 Concluding Remarks

We in this paper present a novel decolorization algorithm that can take into account multi-scale contrast preservation in both spatial and range domain. The algorithm is based on bilateral filtering for mimicking human contrast perception. A local minima voting scheme enables our algorithm to produce several results for an input color image, among which the user-preferred one can be consistently contained. We believe that, by involving more ingredients depending on specific applications, the results for each input image can be further reduced to a final desired one.

Another contribution of this paper is that, through our user study and experiments, we show the potential of the simple RGB2GRAY conversion model. Although the result candidates for each input image are restricted to only 66, high-quality results can be consistently found among the very restricted candidate set (see the last row of Figure 1). In addition, the outcome of our user study can serve as the "ground truth" user-preferred results of RGB2GRAY conversion model on Cadik's dataset, which can benefit future research on this model.

References

ANCUTI, C., ANCUTI, C., AND BEKAERT, P. 2011. Enhancing by saliency-guided decolorization. In *CVPR*.

BALA, R., AND ESCHBACH, R. 2004. Spatial color-to-grayscale transform preserving chrominance edge information. In *Color Imaging Conference*.

CADIK, M. 2008. Perceptual evaluation of color-to-grayscale image conversions. *Computer Graphics Forum*.

GASTAL, E., AND OLIVEIRA, M. 2011. Domain transform for edge-aware image and video processing. *ACM Transaction on Graphics (SIGGRAPH)*.

GOOCH, A., OLSEN, S., TUMBLIN, J., AND GOOCH, B. 2005. Color2gray: salience-preserving color removal. In *ACM Transactions on Graphics (SIGGRAPH)*.

GRUNDLAND, M., AND DODGSON, N. 2007. Decolorize: Fast, contrast enhancing, color to grayscale conversion. *Pattern Recognition*.

KIM, Y., JANG, C., DEMOUTH, J., AND LEE, S. 2009. Robust color-to-gray via nonlinear global mapping. *ACM Transactions on Graphics (SIGGRAPH Asia)*.

LU, C., XU, L., AND JIA, J. 2012. Contrast preserving decolorization. In *ICCP*.

LU, C., XU, L., AND JIA, J. 2012. Real-time contrast preserving decolorization. In *SIGGRAPH Asia Technical Briefs*.

PARIS, S., KORNPROBST, P., TUMBLIN, J., AND DURAND, F. 2009. Bilateral filtering: Theory and applications. *Foundations and Trends in Computer Graphics and Vision*.

SMITH, K., LANDES, P., THOLLOT, J., AND MYSZKOWSKI, K. 2008. Apparent greyscale: A simple and fast conversion to perceptually accurate images and video. *Computer Graphics Forum*.

SONG, M., TAO, D., CHEN, C., LI, X., AND CHEN, C. 2010. Color to gray: visual cue preservation. *PAMI*.

YANG, Q., TAN, K.-H., AND AHUJA, N. 2009. Real-time o(1) bilateral filtering. In *CVPR*.