Interactive Photo Editing on Smartphones
via Intrinsic Decomposition
(Supplementary Material)

Sumit Shekhar[1] [iD], Max Reimann[1] [iD], Maximilian Mayer[1,2], Amir Semmo[1,2] [iD],
Sebastian Pasewaldt[1,2], Jürgen Döllner[1], and Matthias Trapp[1] [iD]

[1]Hasso Plattner Institute for Digital Engineering, University of Potsdam, Germany
[2]Digital Masterpieces GmbH, Germany

1. Ablation Study and Parameter Analysis

In this section, we demonstrate the efficiency of the intermediate steps and different parameter settings to achieve better results in terms of both specularity removal and intrinsic decomposition.

1.1. Specularity Removal

Our specularity removal approach described in Section 3.1 of the main paper consists of three steps namely,

1. identification of specularity,
2. intensity reduction of specular pixels, and
3. chroma inpainting of specular pixels.

The first step makes use of a single parameter to identify specular pixels, the threshold value τ (Section 3.1.1 of the main paper). In Fig. 1, we show how the specular mask changes with increasing value of τ. For the next two steps, only the pixels within the specular mask are processed to remove the inherent specularity. In order to address high- and low- frequency specularity separately, we scale down the positive coefficients of high-frequency and low-frequency intensity sub-bands using parameters κ_h and κ_l respectively (Section 3.1.2 of the main paper). The above intensity reduction step exposes the missing chroma information for saturated specular pixels, see Fig. 2(d). We fill in the missing chroma information by iterative bilateral filtering (Section 3.1.3 of the main paper). The initial chromaticity image with the missing information in specular pixels is considered as C^0, and after $k+1$ iteration the modified image is given as

$$C^{k+1}(\boldsymbol{p}) = \frac{1}{W_p} \sum_{\boldsymbol{q} \in M(\boldsymbol{p})} G_{\sigma_s}(||\boldsymbol{p}-\boldsymbol{q}||) \, G_{\sigma_r}(||C^k(\boldsymbol{p})-C^k(\boldsymbol{q})||) \, C^k(\boldsymbol{q}), \tag{1}$$

where the normalization factor W_p is computed as:

$$W_p = \sum_{\boldsymbol{q} \in M(\boldsymbol{p})} G_{\sigma_s}(||\boldsymbol{p}-\boldsymbol{q}||) \, G_{\sigma_r}(||C^k(\boldsymbol{p})-C^k(\boldsymbol{q})||). \tag{2}$$

The degree of inpainting in each iteration is controlled by parameters σ_s and σ_r for image C^k to get a seamless diffuse output Fig. 2(e). The effect of chroma inpainting with varying values of σ_s and σ_r are depicted in Fig. 3 and Fig. 4 respectively. To further motivate the intermediate steps, we use input and output images for both chroma and intensity in Fig. 5. Our filtering pipeline allows us to individually remove high frequency and/or low frequency specularities and can also be used to enhance it (refer to Fig. 6).

Input $\quad\quad$ $\tau = 0.04$ $\quad\quad$ $\tau = 0.12$ $\quad\quad$ $\tau = 0.24$
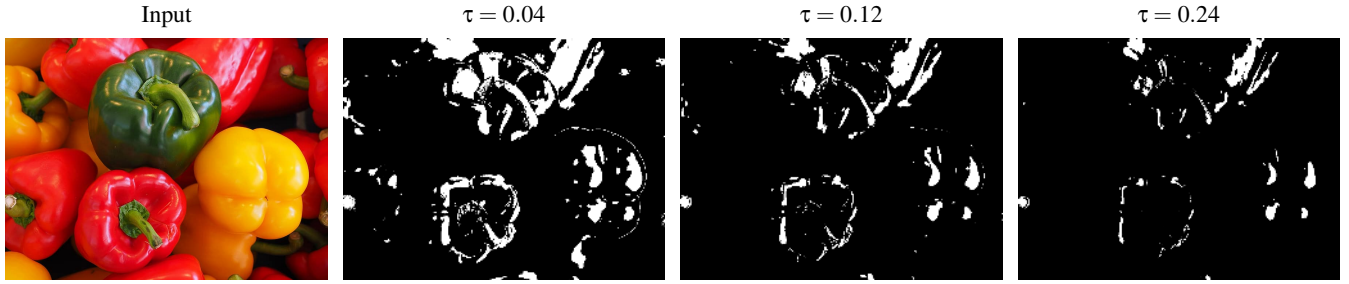
Figure 1: *The specular mask for the given input image with the increasing value of threshold parameter* τ.

(a) Input $\quad$ (b) Only Low Freq. $\quad$ (c) Only High Freq. $\quad$ (d) Both $\quad$ (e) w. Inpainting

Figure 2: *Diffuse outputs when we only reduce positive coefficients of (b) low frequency* $(-0.5 \leq \kappa_l \leq 0.2)$ *and (c) high frequency* $(-0.5 \leq \kappa_h \leq 0.2)$ *intensity subbands individually. The removal of both high and low frequency coefficients exposes the missing chroma information (d) which is then inpainted to get a seamless output (e).*

Initial $\quad\quad$ $\sigma_s = 1.5$ $\quad\quad$ $\sigma_s = 3.0$ $\quad\quad$ $\sigma_s = 7.25$
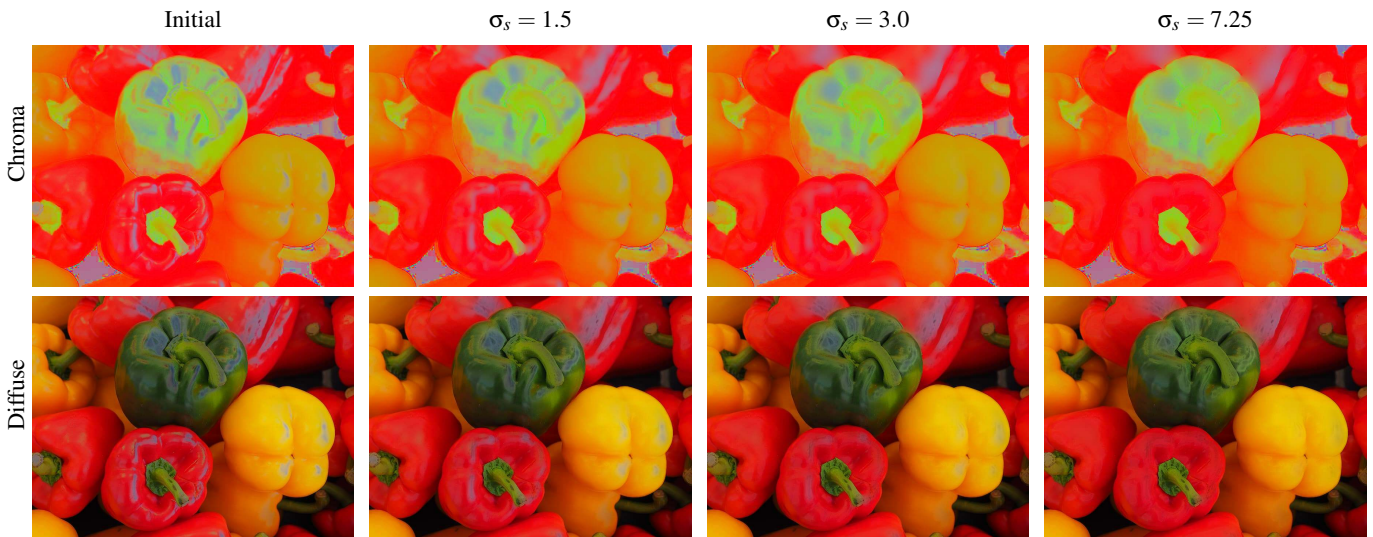
Figure 3: *The inpainted chroma and corresponding diffuse image output for different values of* σ_s, *for all the above cases we use* $\sigma_r = 3.0$ *and a kernel size of* 10×10 *pixels.*
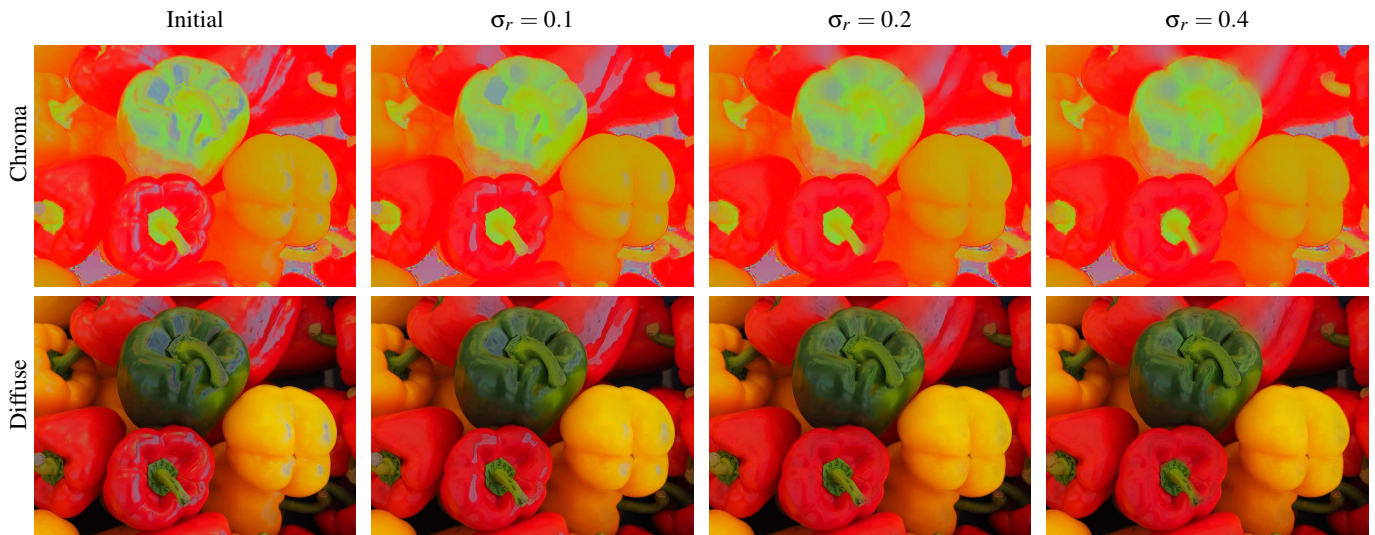
Figure 4: *The inpainted chroma and corresponding diffuse image output for different values of σ_r, for all the above cases we use $\sigma_s = 7.25$ and a kernel size of 10×10 pixels.*
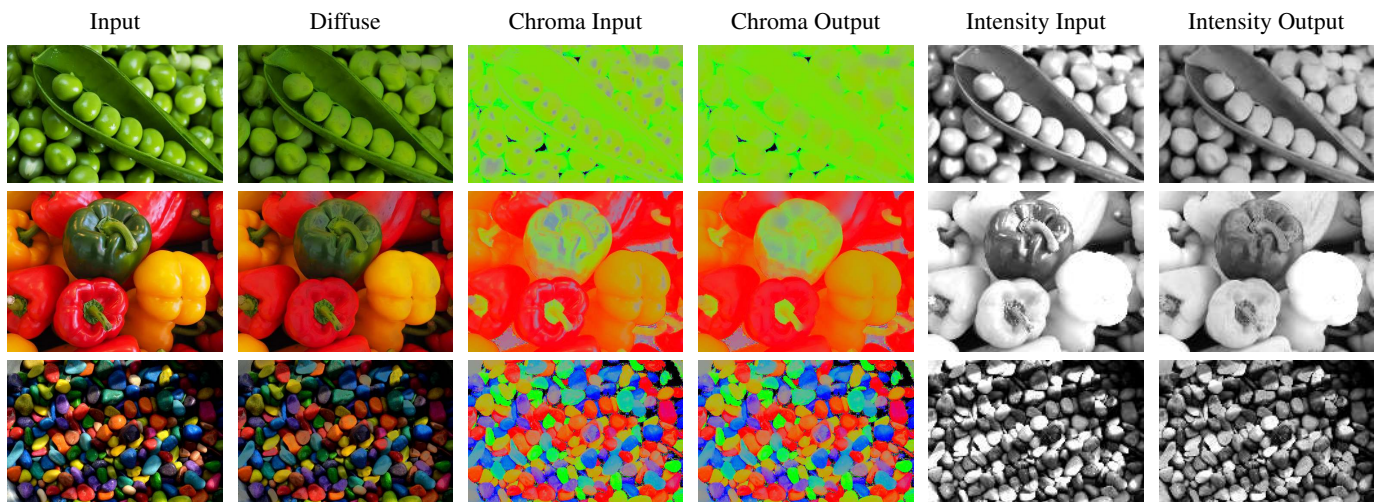
Figure 5: *Intermediate results of our diffuse-image generation pipeline. Note how the output—intensity and chroma—images are modified as compared to the input, please zoom-in for details.*

(a) Input	(b) Diffuse	(c) Diffuse Low Freq.	(d) Diffuse High Freq.	(e) Specular Enhanced
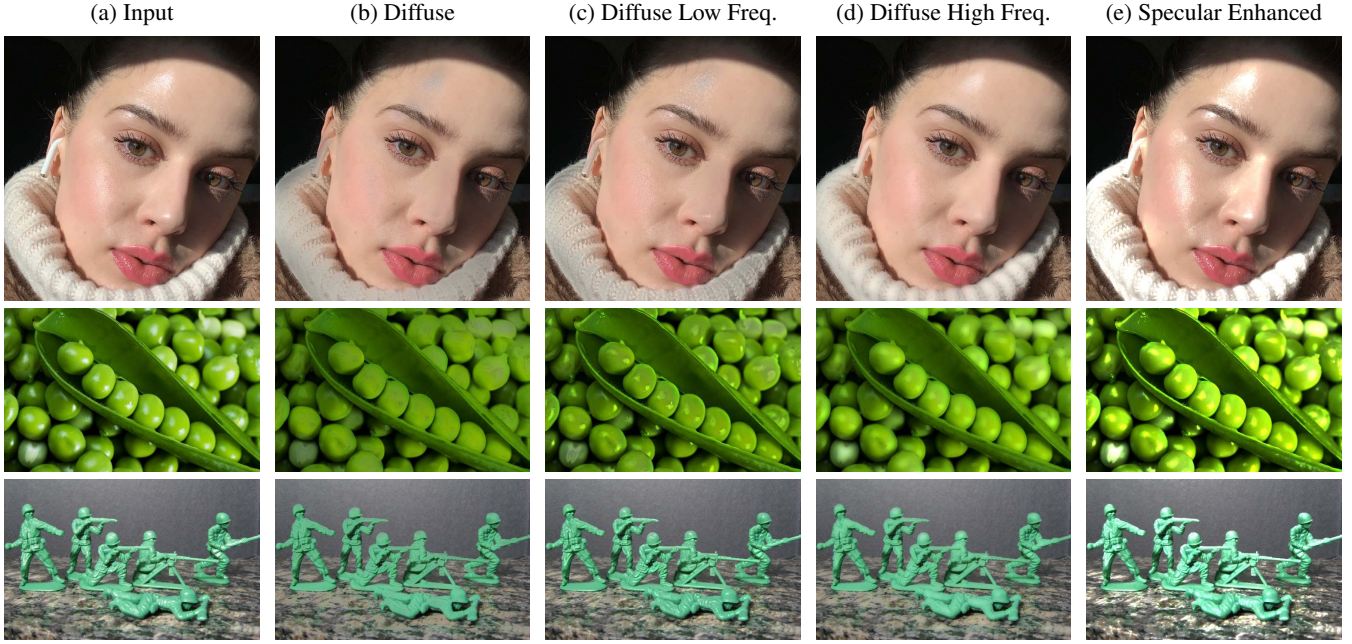
Figure 6: *To obtain the diffuse image (b), we remove both high and low frequency specularitites ($-0.5 \leq \kappa_h, \kappa_l \leq 0.2$). One can remove only low frequency specularities ($-0.5 \leq \kappa_l \leq 0.2$) or only high frequency specularities ($-0.5 \leq \kappa_h \leq 0.2$) to obtain (c) and (d) respectively. The specularities can also be seamlessly enhanced using our pipeline ($\kappa_h, \kappa_l > 1.0$) to obtain (e).*

The list of images (and their sources) used to demonstrate our specularity removal method:

- Face - https://cdn.repeller.com/wp-content/uploads/2020/02/YLM_THE_POUF_PC-848x1272.jpg
- Peas - https://img1.goodfon.com/wallpaper/nbig/3/19/zelenyy-goroshek-goroshiny.jpg
- Paprika - https://p0.pikist.com/photos/124/344/sweet-peppers-paprika-green-yellow-red-healthy-vitamins-pepper-veget
jpg
- Pebbles - https://c.pxhere.com/photos/ab/89/rocks_stones_colorful_colourful_pebbles-641429.jpg!d
- Toy Soldiers - https://2.bp.blogspot.com/-YvcgGOcL404/V1YLs-_LZFI/AAAAAAAACFA/xL2rEHvA9zI9EGKcTgqz7MVp4tgmvvJnwCLcB/s1600/lgam%2B1.jpg

1.2. Intrinsic Decomposition

For intrinsic decomposition, we solve for both albedo (*a*) and shading (*s*) simultaneously by minimizing the below energy function (Section 3.2 of the main paper),

$$E(\boldsymbol{x}) = \frac{1}{2}\left(\underbrace{\lambda_{ra}E_{ra}(\boldsymbol{x})}_{E_1} + \underbrace{\lambda_{rs}E_{rs}(\boldsymbol{x}) + \lambda_d E_d(\boldsymbol{x})}_{E_2}\right) + \underbrace{\lambda_{sp}||a(\boldsymbol{x})||_1}_{E_3} \tag{3}$$

where $\lambda_{ra}E_{ra}$, $\lambda_{rs}E_{rs}$, and $\lambda_d E_d$ are retinex-albedo smoothness, retinex-shading smoothness, and data terms respectively with their corresponding weights. We use a L_1 regularizer to enforce sparsity in the resulting albedo controlled by the weight λ_{sp}. We can further classify the energy in three parts: (*i*) E_1 – responsible for edge-preserving smoothing of initial albedo, (*ii*) E_2 – is the contribution of shading smoothing and its coupling with albedo optimization and (*iii*) E_3 – that controls the sparsity in the resulting albedo, as depicted in Eqn. (3). For the ablation study, we build our energy one component at a time and analyze how it improves the final output, see Fig. 7. Further, to understand the effect of individual energy term on the final result, we vary their weight parameter and analyze how it affects the output albedo or shading. For the given input image, we get the best results with the following parameter values: for energy formulation – $\lambda_{ra} = 5.0$, $\alpha_{ra} = 10.0$, $\lambda_{rs} = 50.0$, $\alpha_{rs} = 100.0$, $\lambda_d = 0.02$, and for optimization solver – $\alpha = 0.0025$, $\beta = 0.01$, $\lambda_{sp} = 0.14$ respectively. We then vary the weights of individual energy terms and analyze the output albedo/shading as shown in Fig. 8, Fig. 9, Fig. 10, and Fig. 11.

(a) Input   (b) Only E_1 (c) $E_1 + E_2$ (d) $E_1 + E_2 + E_3$
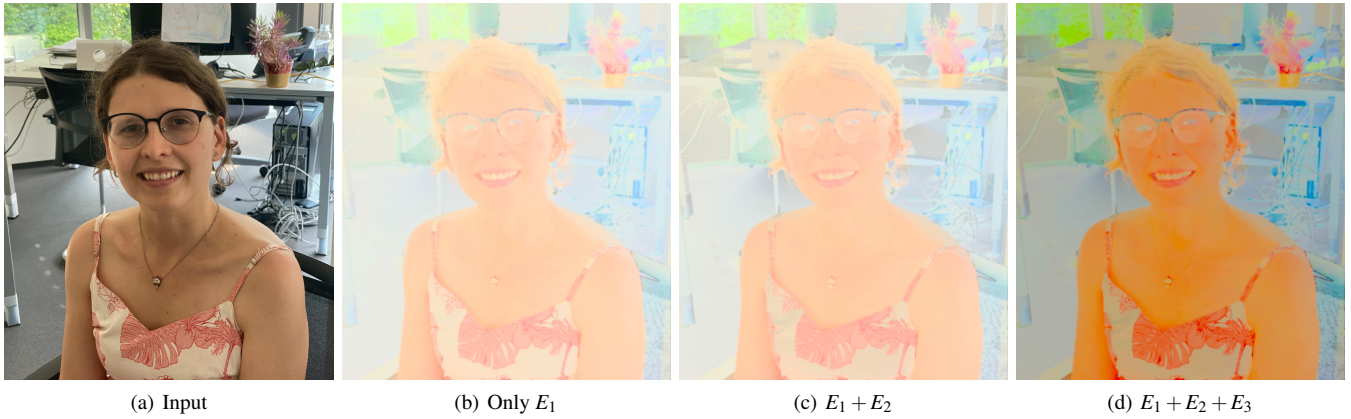
Figure 7: *Input image and corresponding albedo as we add different components to the energy formulation. Note, how the fine details in the resulting albedo becomes more visible with each component.*


(a) Input   (b) $\lambda_{ra} = 2.32$ (c) $\lambda_{ra} = 15.0$ (d) $\lambda_{ra} = 30.63$
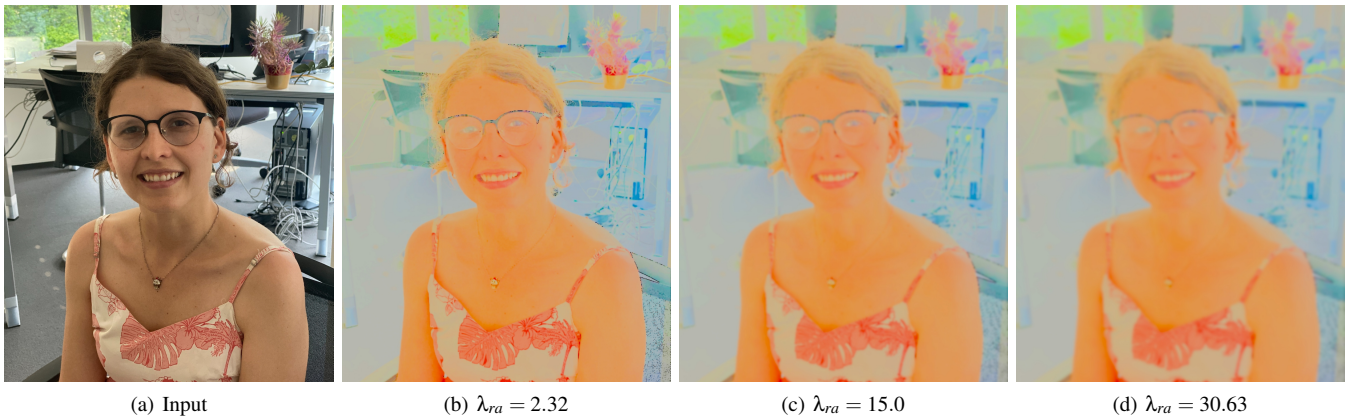
Figure 8: *Input image and corresponding albedo output with increasing value of threshold λ_{ra}. The "Retinex-Albedo smoothness" term smooths the output albedo while preserving the edges. However, if the smoothing is more it starts blurring important details, see (d).*

(a) Input   (b) $\lambda_{rs} = 16.12$ (c) $\lambda_{rs} = 50.0$ (d) $\lambda_{rs} = 90.83$

Figure 9: *Input image and corresponding shading output with increasing value of threshold λ_{rs}. The "Retinex-Shading smoothness" term smooths the output shading while preserving the edges. However, if the smoothing is more it starts blurring important details, see (d).*

(a) Input   (b) $\lambda_d = 0.005$ (c) $\lambda_d = 0.024$ (d) $\lambda_d = 0.046$
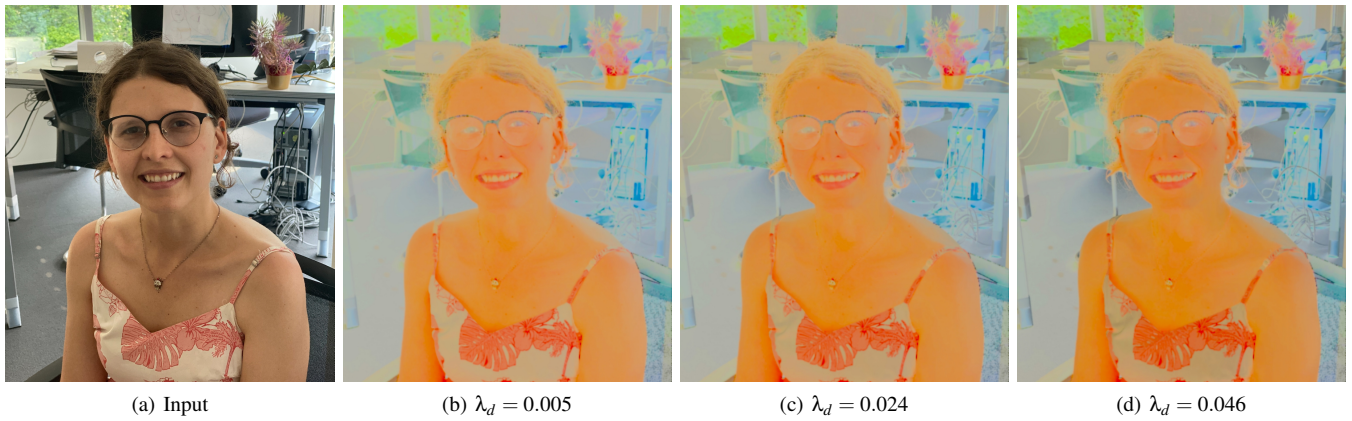
Figure 10: *Input image and corresponding albedo with increasing value of threshold λ_d. The "Data" term couples the separate optimization solvers of albedo and shading. However, if this coupling is too strong it introduces shading details in the output albedo, see (d).*

(a) Input   (b) $\lambda_{rs} = 0.045$ (c) $\lambda_{rs} = 0.145$ (d) $\lambda_{rs} = 0.222$
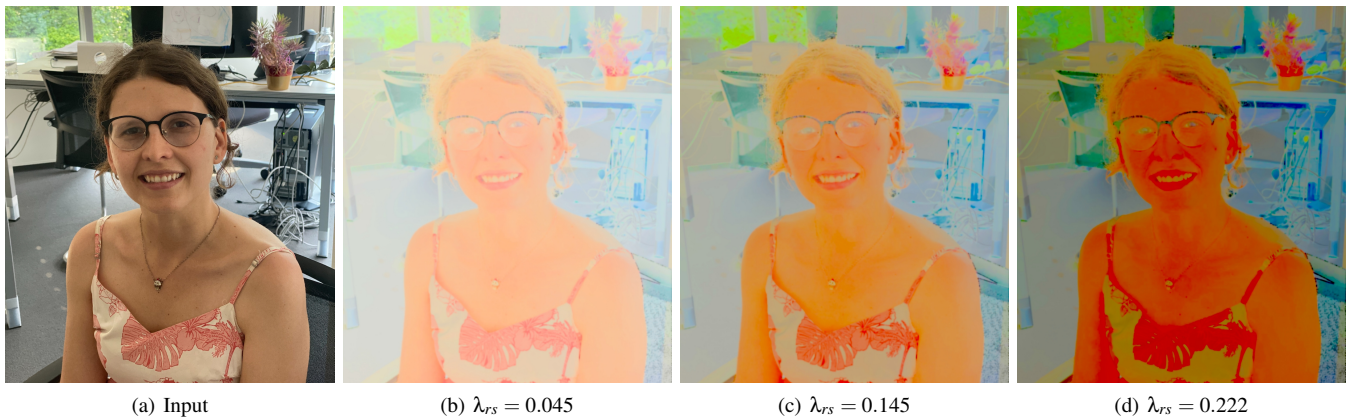
Figure 11: *Input image and corresponding albedo with increasing value of threshold λ_{sp}. The "L_1 Sparsity" term makes the output albedo more sparse however, an excess of sparsity leads to spatially inconsistent patch-like artifacts in the output albedo, see (d).*

2. Material Appearance Editing – More Results

Figure 12: *Showcasing more results for material editing. Please zoom in for better visualization of tattoo results.*
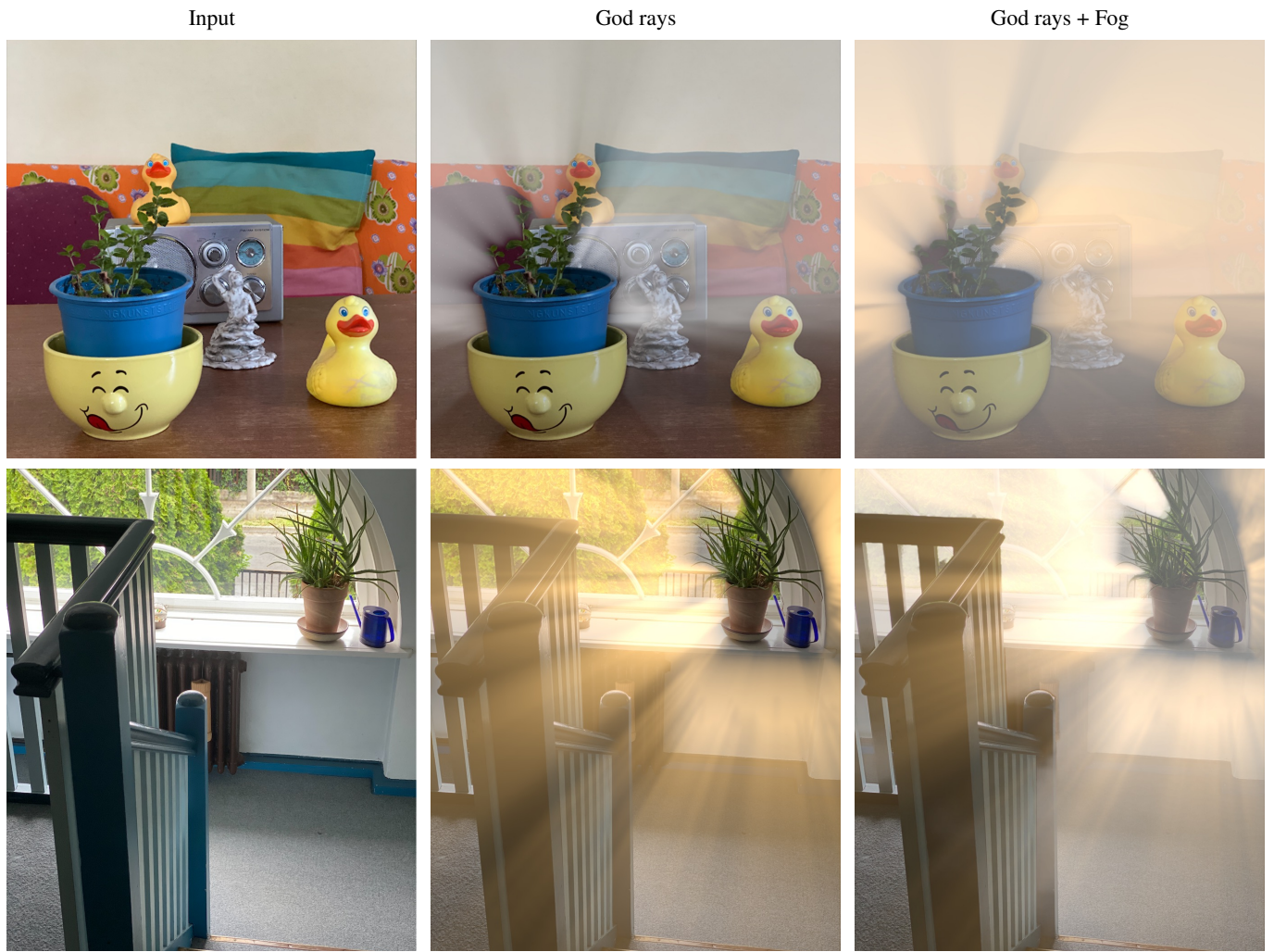
3. Atmospheric Appearance Editing – More Results

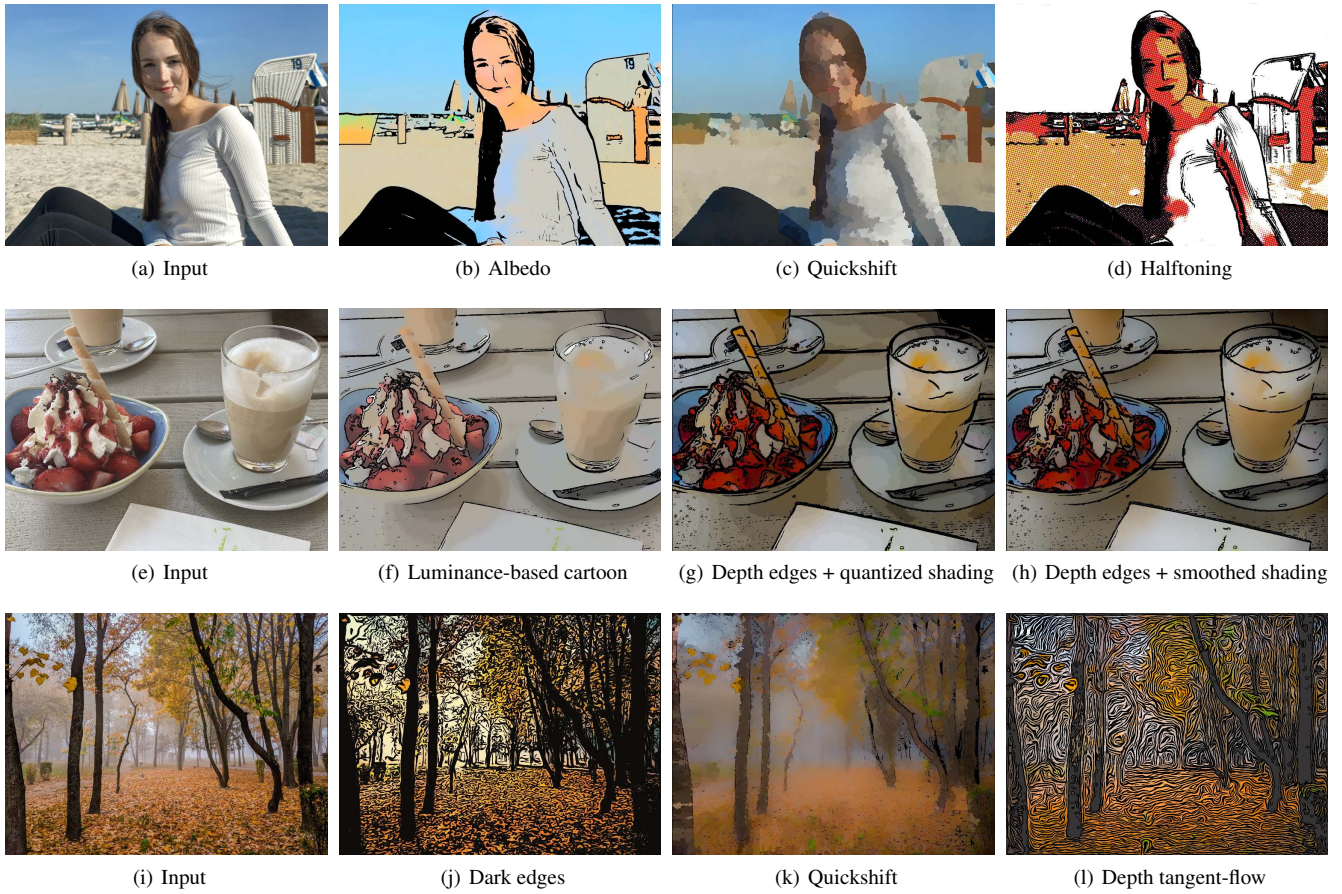| Input	Low-density Fog	High-density Fog

Figure 13: *Showcasing more results for virtual fog.*

Input   God rays   God rays + Fog

Figure 14: *Showcasing more results for God rays.*

4. Image Stylization – More Results

(a) Input   (b) Albedo   (c) Quickshift   (d) Halftoning

(e) Input   (f) Luminance-based cartoon   (g) Depth edges + quantized shading   (h) Depth edges + smoothed shading

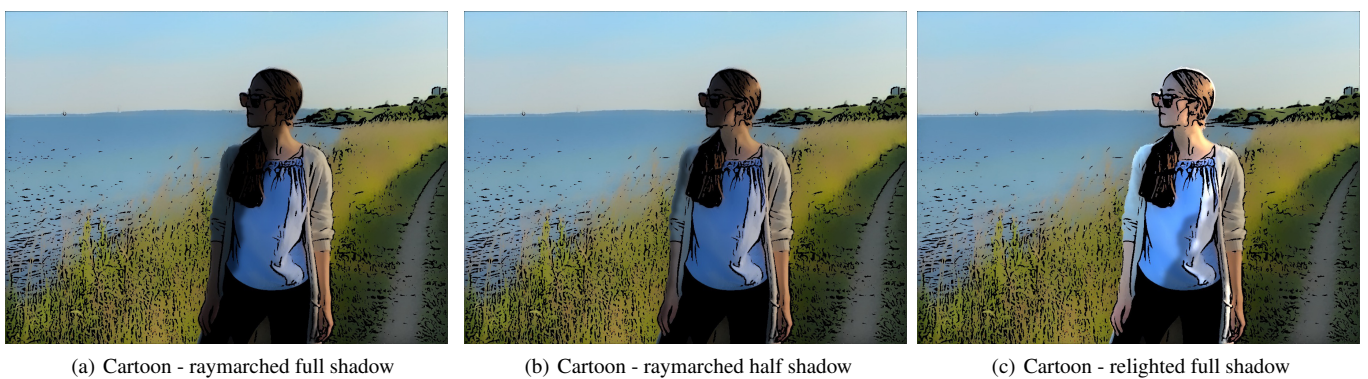(i) Input   (j) Dark edges   (k) Quickshift   (l) Depth tangent-flow

Figure 15: *More results for full stylization pipeline. The first row demonstrates variants of using decomposition information, by (b) only using albedo, (c) adding segmented shading using quickshift [VS08] and (d) halftoning the shading. The second row compares the color luminance-based cartoon filter by Winnemöller et al. [WOG06, WKO12] (f) with depth-edge enhanced cartoon (g) & (h), the depth edges are especially noticeable on the glass. Also, (g) & (h) show variants of albedo & shading recombination for a posterize effect and an airbrush-like effect. The third row demonstrates stylization in outdoor scenes without a foreground subject. Increasing the blackness and edge thickness creates a dark/moody effect (j) and using quickshift shading increases the fogginess in the image (k). While the captured depth accuracy is limited, it can be used to create abstract effects such as visualizing depth (l) by increasing the depth-normal boost factor ω in ST_ϕ.*

(a) Input Image

(b) Cartoon - luminance edges only

(c) Cartoon - depth edges only

(d) Cartoon - combined edges

Figure 16: *Luminance-based edges and depth edges are combined to reduce obstructions (as visible in the face) and keep thick outlines around actual borders*

(a) Cartoon - raymarched full shadow

(b) Cartoon - raymarched half shadow

(c) Cartoon - relighted full shadow

Figure 17: *Raymarched shadows can be used to shadow (a), (b) or relight (c) the foreground subject.*

(a) Cartoon - no shadow   (b) Cartoon - hard shadow   (c) Cartoon - halftoned shadows
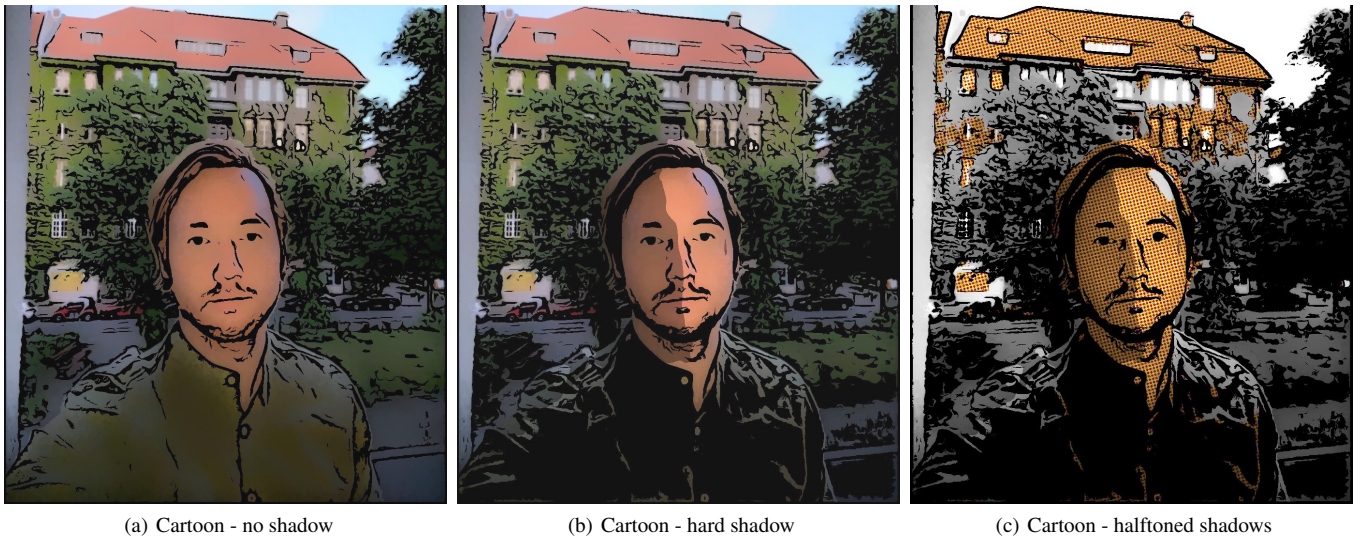
Figure 18: *Normal-angle based hard shadows can be used for scene emphasis (b), and can be combined with halftoning to create a "noir" cartoon style (c).*

(a) Input   (b) Very fine ($\sigma = 14, \tau = 2.0$)   (c) Fine ($\sigma = 14, \tau = 4.0$)

(d) Medium ($\sigma = 15.0, \tau = 6.0$)   (e) Coarse ($\sigma = 18.0, \tau = 8.0$)   (f) Very Coarse ($\sigma = 20.0, \tau = 10.0$)

Figure 19: *Different levels of coarseness in quickshift shading. Here, σ controls the kernel size of the Gaussian distribution, while τ is the distance threshold used when comparing neighboring pixels and affects the size of the kernel window. For parameter details, please refer to Vedaldi et al. [VS08].*
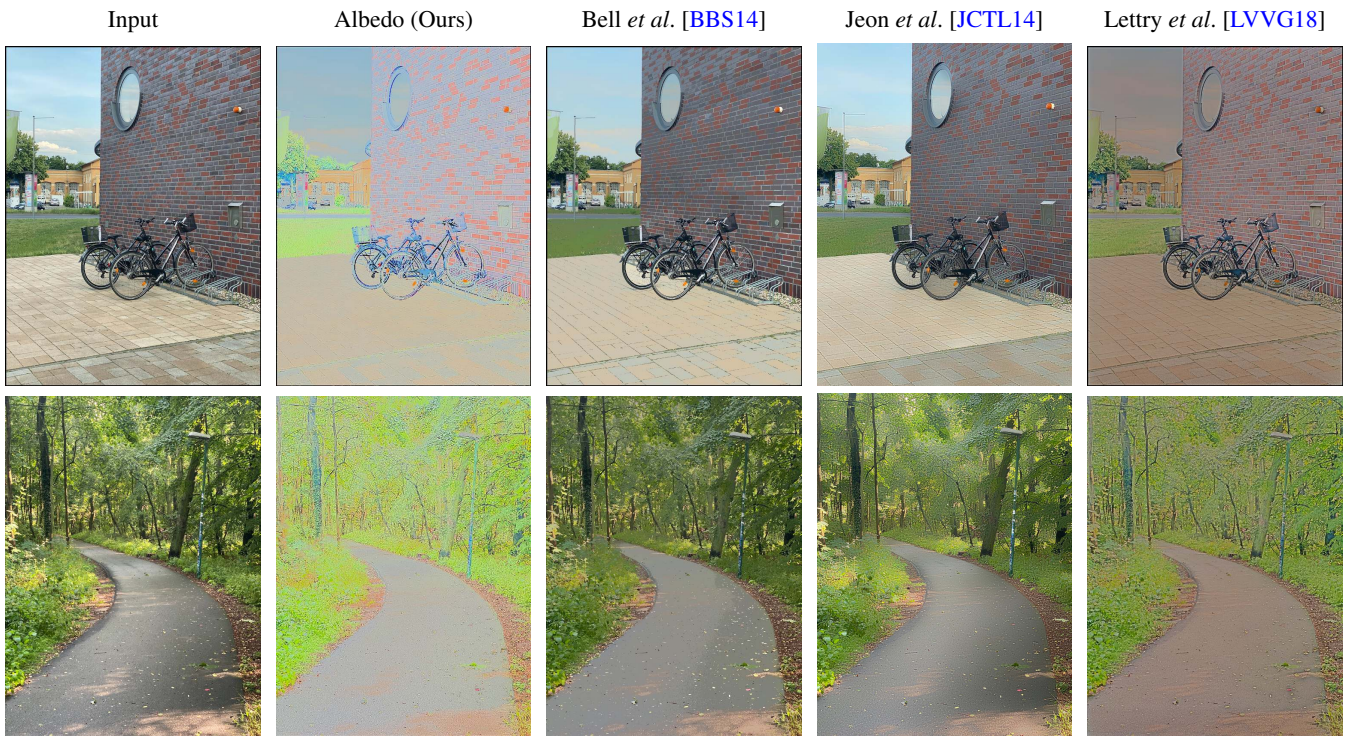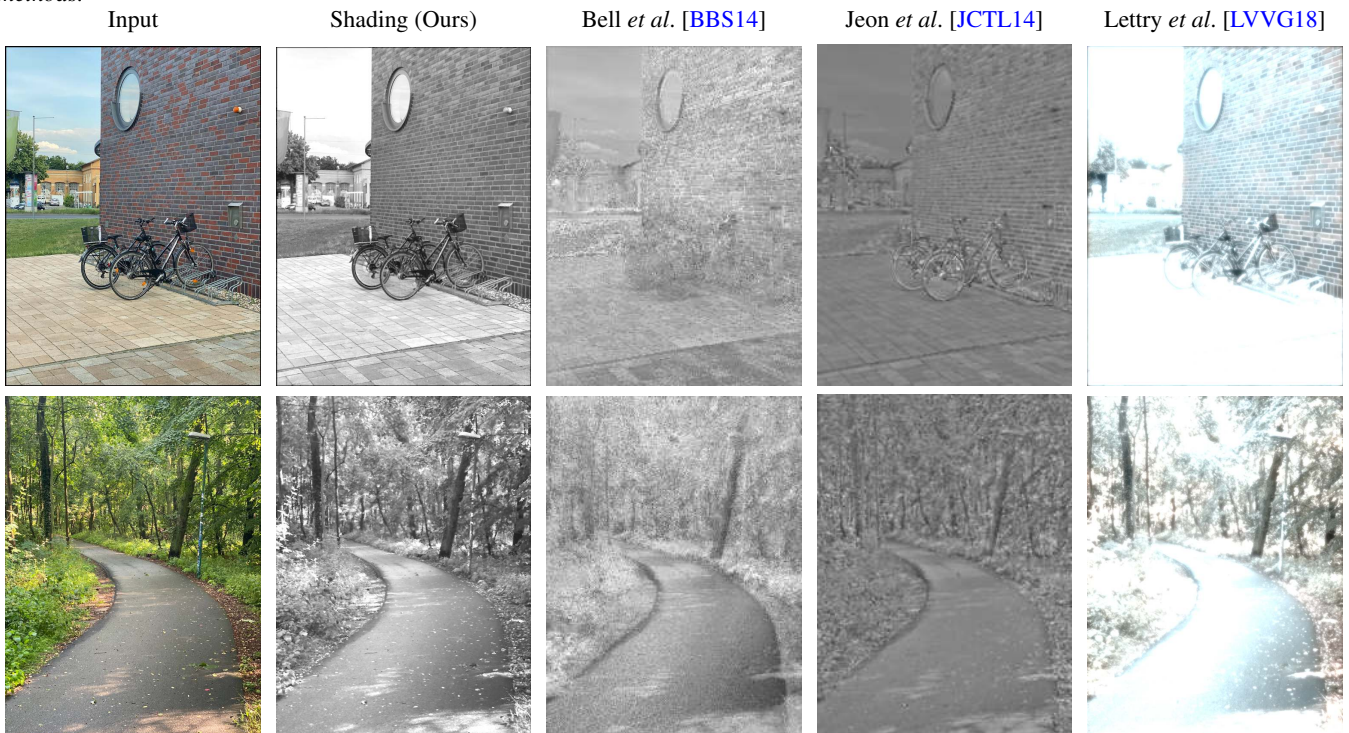
5. Intrinsic Decomposition – More Results

Input	Shading (Ours)	Bell *et al.* [BBS14]	Jeon *et al.* [JCTL14]	Lettry *et al.* [LVVG18]
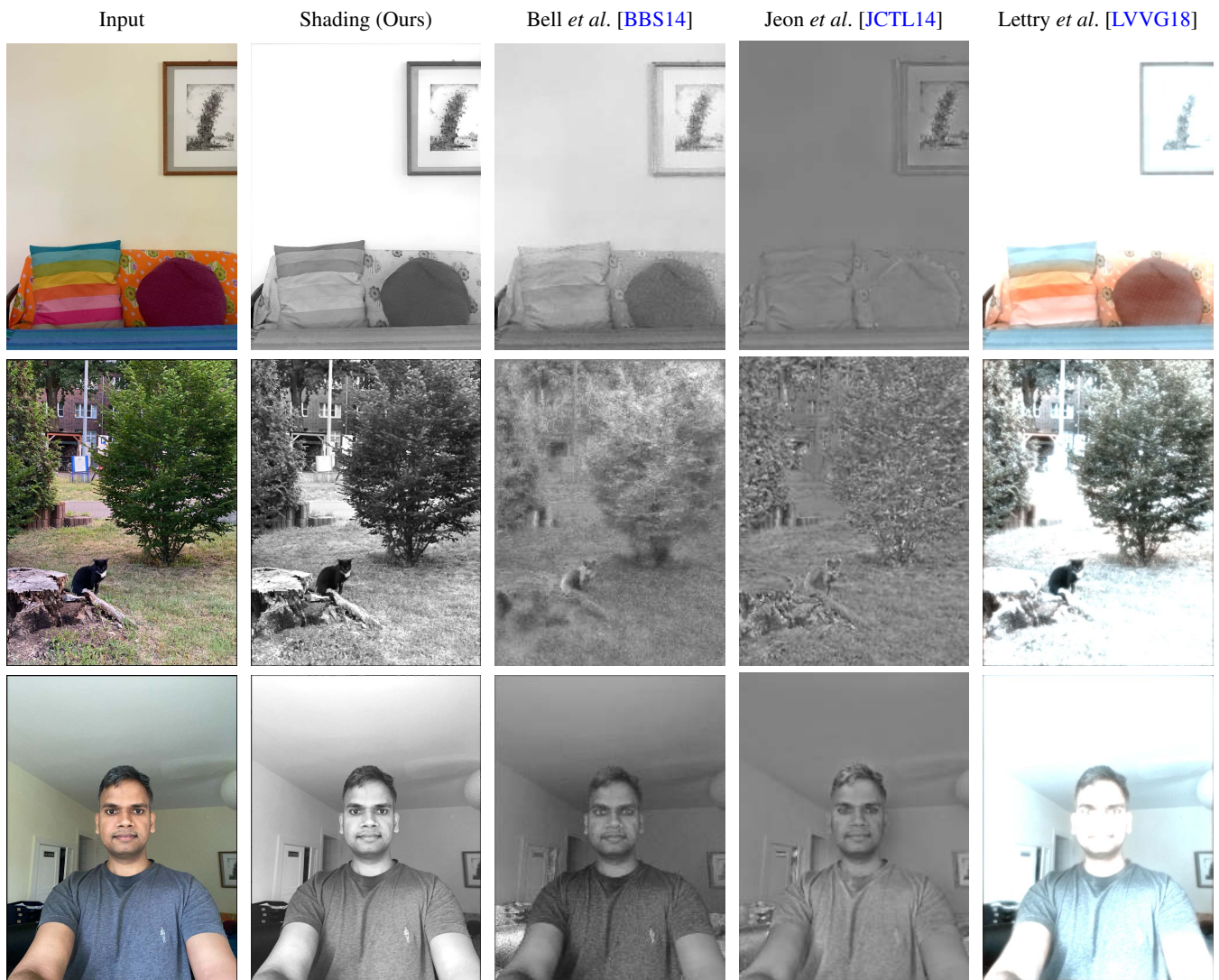
Figure 20: *Comparison of intrinsic decomposition with other methods. The figure contains input image and the corresponding shading (for Fig.12 in main paper) obtained using ours, Bell et al. [BBS14], Jeon et al. [JCTL14], and Lettry et al. [LVVG18] intrinsic decomposition methods.*

Input	Albedo (Ours)	Bell *et al*. [BBS14]	Jeon *et al*. [JCTL14]	Lettry *et al*. [LVVG18]

Figure 21: *Comparison of intrinsic decomposition albedo with other methods for outdoor scenes. The figure contains input image and the corresponding albedo obtained using ours, Bell et al. [BBS14], Jeon et al. [JCTL14], and Lettry et al. [LVVG18] intrinsic decomposition methods.*

Input	Shading (Ours)	Bell *et al*. [BBS14]	Jeon *et al*. [JCTL14]	Lettry *et al*. [LVVG18]

Figure 22: *Comparison of intrinsic decomposition shading with other methods for outdoor scenes. The figure contains input image and the corresponding shading obtained using ours, Bell et al. [BBS14], Jeon et al. [JCTL14], and Lettry et al. [LVVG18] intrinsic decomposition methods.*

6. Specularity Removal – More Results

Input	Ground Truth	Ours	Fu *et al.* [FZS*19]	Akashi *et al.* [AO16]
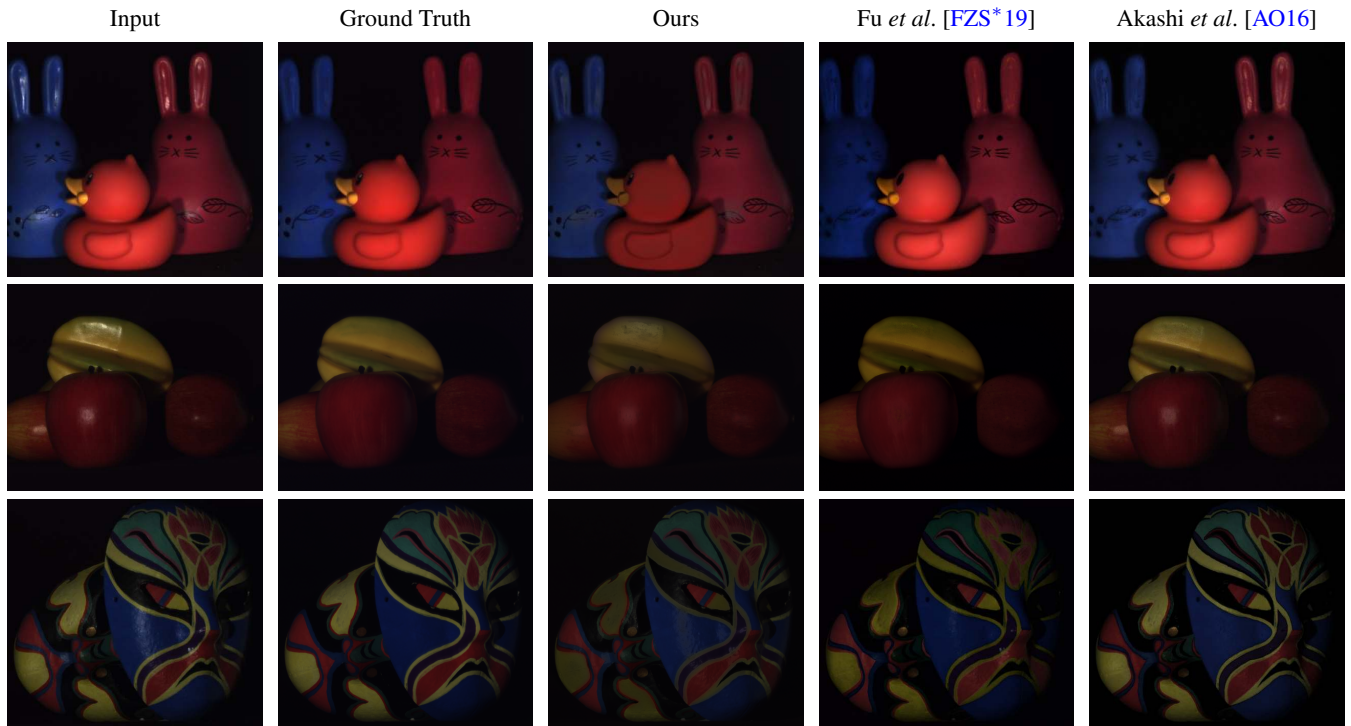
Figure 23: *Comparison of specularity removal for images captured in a lab setting. The figure contains input images, its respective ground truth diffuse version, and the corresponding diffuse images obtained using ours, Fu et al. [FZS*19], and Akashi et al. [AO16] specularity removal methods.*

7. Image Editing

In this section we show image three different type of image-editing results based on intrinsic decomposition. For ground truth comparison we make use of the intrinsic image dataset of Bonneel *et al.* [BKPB17]. In order to introduce image edits using intrinsic layers we make use of the appearance-editing framework from Beigpour *et al.* [BSM*18].

As the first edit we **Recolor** the below image using intrinsic layers obtained from competing methods and compare that with the ground truth,
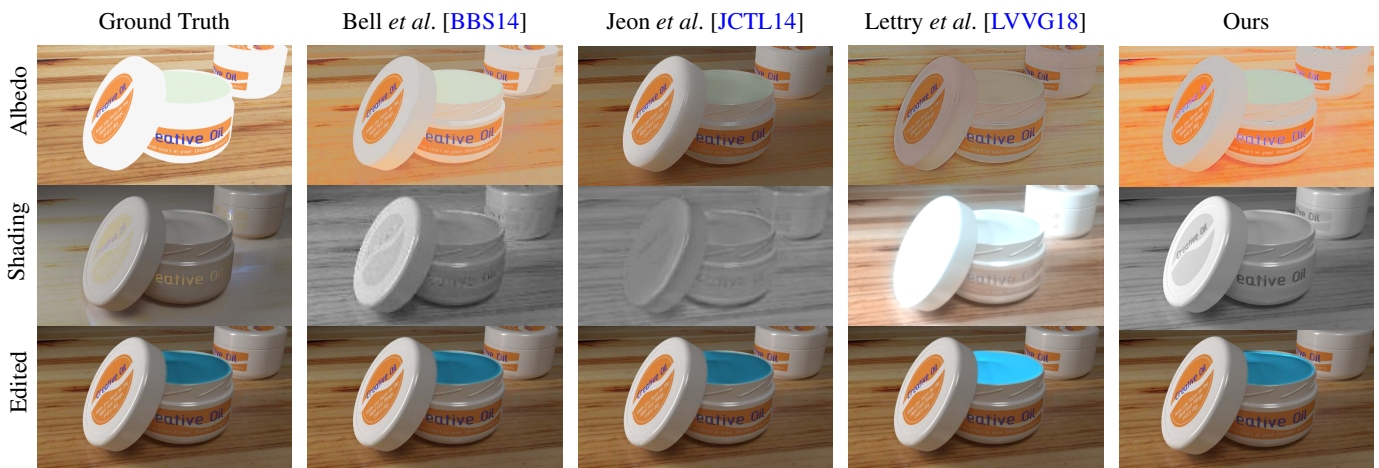

Figure 24: *Cream Box*

Figure 25: *The computed intrinsic layers and corresponding edited image. Note that the recolored image looks quite different for the method of Lettry et al. due to inaccuracies in the computed albedo.*

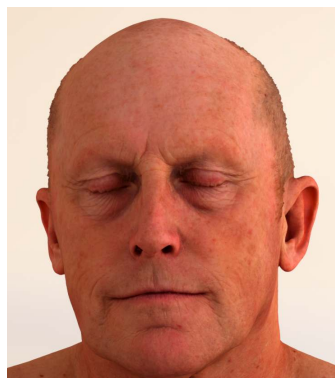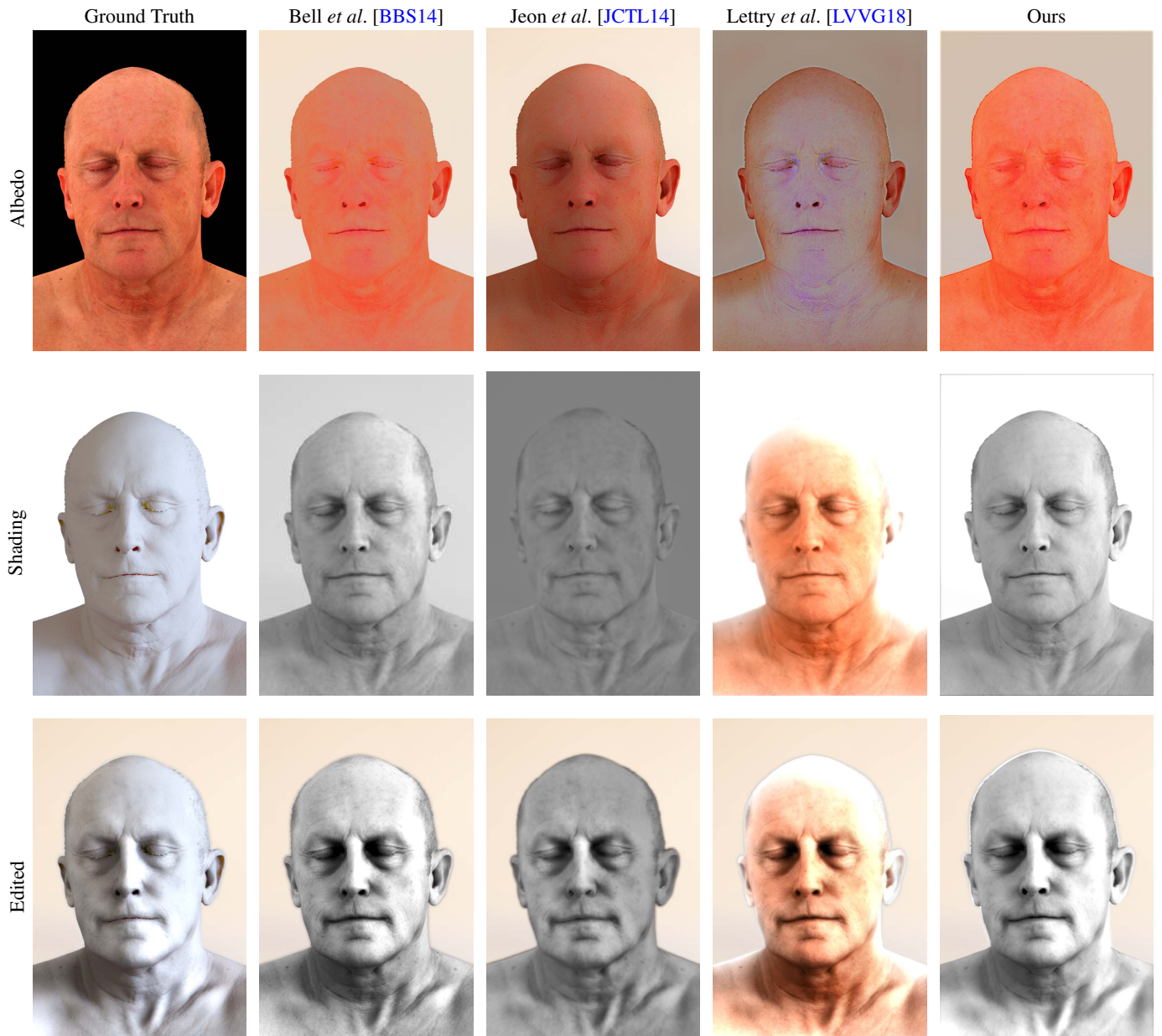For the next edit we perform **Material-Editing** for the below image,

Figure 26: *Human Face*

Figure 27: *For this edit, we try to give the skin a shiny silver-metal like look using the intrinsic layers. For the method of Lettry et al. the shading is multi-channel and also contains details about skin texture and tone thus preventing a plausible edit.*

For the next edit we perform **Retexturing** for the below image,

Figure 28: *Corridor*

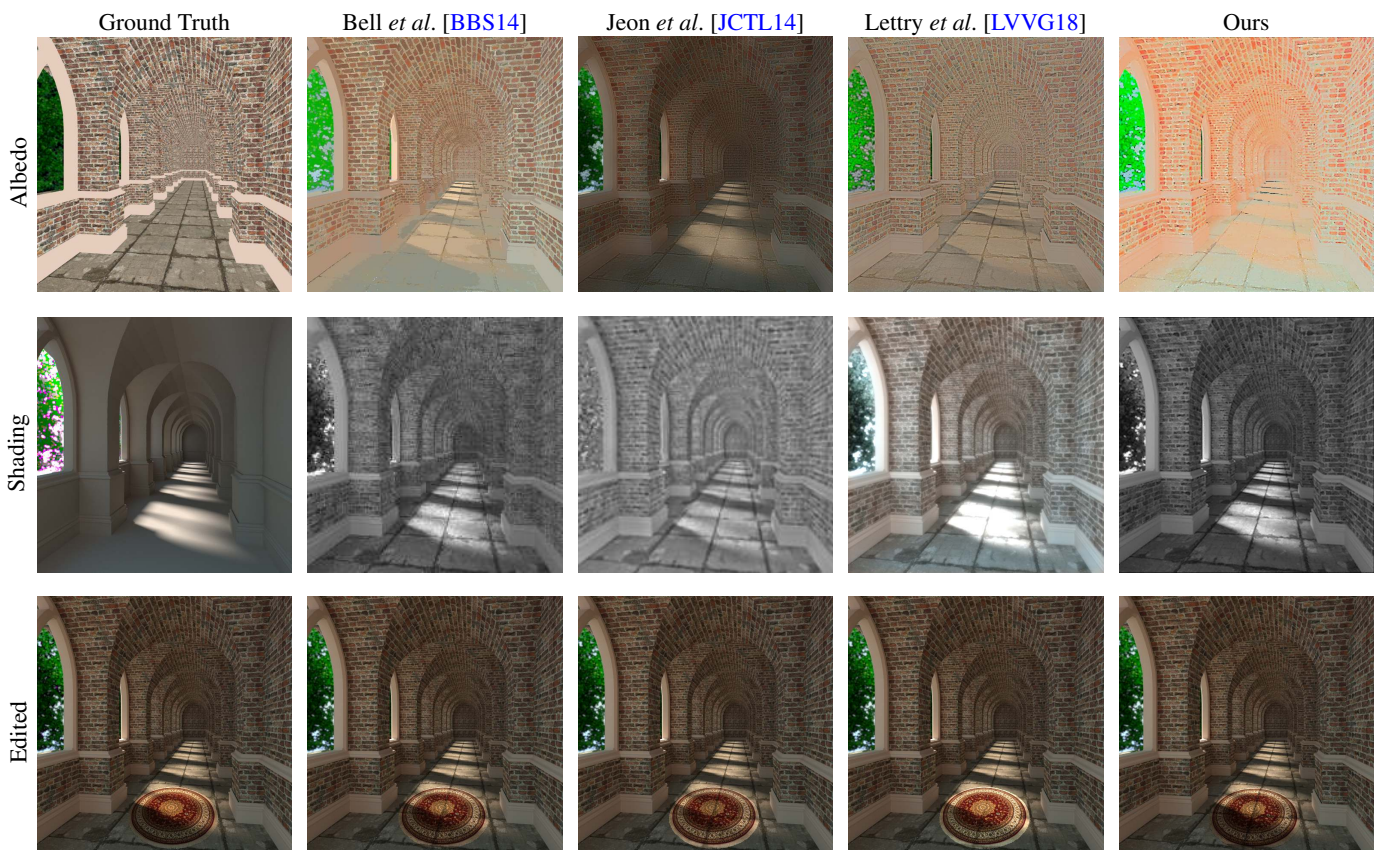| Ground Truth	Bell *et al*. [BBS14]	Jeon *et al*. [JCTL14]	Lettry *et al*. [LVVG18]	Ours

Figure 29: *As part of this edit we retexture the floor of the corridor by compositing the image of a carpet in the albedo intrinsic layer. It is a challenging edit due to both shading and albedo variations in the given region. None of the method achieves a flawless result, however except the method of Lettry et al. the others are able to preserve shading variations in the output.*

References

[AO16] AKASHI Y., OKATANI T.: Separation of reflection components by sparse non-negative matrix factorization. Computer Vision and Image Understanding 146, C (May 2016), 77–85. `doi:10.1016/j.cviu.2015.09.001`. 15

[BBS14] BELL S., BALA K., SNAVELY N.: Intrinsic images in the wild. ACM Transactions on Graphics 33, 4 (July 2014). `doi:10.1145/2601097.2601206`. 13, 14, 16, 17, 18

[BKPB17] BONNEEL N., KOVACS B., PARIS S., BALA K.: Intrinsic decompositions for image editing. Computer Graphics Forum 36, 2 (May 2017), 593–609. `doi:10.1111/cgf.13149`. 16

[BSM*18] BEIGPOUR S., SHEKHAR S., MANSOURYAR M., MYSZKOWSKI K., SEIDEL H.-P.: Light-field appearance editing based on intrinsic decomposition. Journal of Perceptual Imaging 1, 1 (2018), 15. `doi:10.2352/J.Percept.Imaging.2018.1.1.010502`. 16

[FZS*19] FU G., ZHANG Q., SONG C., LIN Q., XIAO C.: Specular highlight removal for real-world images. Computer Graphics Forum 38, 7 (2019), 253–263. `doi:10.1111/cgf.13834`. 15

[JCTL14] JEON J., CHO S., TONG X., LEE S.: Intrinsic image decomposition using structure-texture separation and surface normals. In European Conference on Computer Vision (ECCV) (2014), pp. 218–233. `doi:10.1007/978-3-319-10584-0_15`. 13, 14, 16, 17, 18

[LVVG18] LETTRY L., VANHOEY K., VAN GOOL L.: Unsupervised deep single-image intrinsic decomposition using illumination-varying image sequences. Computer Graphics Forum 37, 7 (2018), 409–419. `doi:10.1111/cgf.13578`. 13, 14, 16, 17, 18

[VS08] VEDALDI A., SOATTO S.: Quick shift and kernel methods for mode seeking. In European Conference on Computer Vision (ECCV) (2008), pp. 705–718. `doi:10.1007/978-3-540-88693-8_52`. 10, 12

[WKO12] WINNEMÖLLER H., KYPRIANIDIS J. E., OLSEN S. C.: Xdog: an extended difference-of-gaussians compendium including advanced image stylization. Computers & Graphics 36, 6 (2012), 740–753. `doi:10.1016/j.cag.2012.03.004`. 10

[WOG06] WINNEMÖLLER H., OLSEN S. C., GOOCH B.: Real-time video abstraction. ACM Transactions On Graphics (TOG) 25, 3 (2006), 1221–1226. 10