Ground-truth dataset and baseline evaluations for image base-detail separation algorithms

Xuan Dong
UCLA
dongxuan8811@gmail.com

Boyan Bonev
Synaptics
bonev@ucla.edu

Weixin Li
UCLA
lwx@cs.ucla.edu

Weichao Qiu
UCLA
qiuwch@gmail.com

Xianjie Chen
UCLA
cxj@ucla.edu

Alan Yuille
UCLA
yuille@stat.ucla.edu

November 24, 2015

Abstract

Base-detail separation is a fundamental computer vision problem consisting of modeling a smooth base layer with the coarse structures, and a detail layer containing the texture-like structures. One of the challenges of estimating the base is to preserve sharp boundaries between objects or parts to avoid halo artifacts. Many methods have been proposed to address this problem, but there is no ground-truth dataset of real images for quantitative evaluation. We proposed a procedure to construct such a dataset, and provide two datasets: Pascal Base-Detail and Fashionista Base-Detail, containing 1000 and 250 images, respectively. Our assumption is that the base is piecewise smooth and we label the appearance of each piece by a polynomial model. The pieces are objects and parts of objects, obtained from human annotations. Finally, we proposed a way to evaluate methods with our base-detail ground-truth and we compared the performances of seven state-of-the-art algorithms.

1 Introduction

Base-detail separation has been traditionally considered as a specific computer graphics task. But its scope is much more general. We argue that base-detail separation is a fundamental problem in computer vision, image processing, signal processing, and related fields. It defines a simplified coarse representation of an image with its basic structures (base layer), and a detailed representation which may contain texture, fine details or just noise (detail layer), as shown in Fig 2. This definition leaves open what is detail and what is base. For instance, in an image of a crowd of people, we could argue that their heads and faces form a texture, or detail, over a common surface which could be their average color. At a less coarse level, we could say that each individual head is composed of two base regions: the hair and the face, where the details are the hair texture, the eyes, nose and mouth. We could go into still more detail and argue that the mouth of a person could also be separated into a smooth surface (or base), and the details of the lips, if there is enough resolution. There is no general rule that can establish exactly how fine or coarse base and detail should be. For this reason, in this work we argue that base-detail separation should be formulated as a hierarchical problem. There is no ideal hierarchy depth that can be selected and, in practice, the cutoff thresholds of the base-detail algorithms require different settings depending on the application. Roughly speaking, from fine to coarse, the hierarchical base-detail separation can be classified as pixel level, sub-part level, part level, and object level. Fig. 1 shows an example of hierarchical base-detail separation of an image.

Base-detail separation is useful for a number of appli-
Figure 1: Example of hierarchical base-detail separation. The original image (left) is decomposed into base (first row) and detail (second row). Columns are sorted from coarser to finer taxonomy in the hierarchy. Depending on the hierarchy level, details contain different information, e.g., the nose of the dog can be considered as detail, or it can be considered as a part and the nostrils are the detail (see third row). Note that detail is represented with a very high contrast here, for better visualization.

Figure 2: Base-Detail separation is a fundamental vision problem. It relates to a wide range of tasks, from low-level image processing to higher-level problems, like the separation of texture from structure. (Here, D is shown with increased contrast).

Many separation algorithms have been proposed, such as the Gaussian filter, and the bilateral filter [3]. These filters smooth the image to obtain the base, and the residual is the detail. Filtering is not usually formulated in terms of base detail separation. So it is not obvious at which level of the hierarchy they are performed. But, in practice, their performance at part level is very important for applications like contrast management and exposure correction. It is because in these applications, base-detail separation needs performing at part level and failing to do it correctly will introduce halo artifacts into the final enhancement results. Fig. 3 shows the types of errors, such as halo artifacts, in exposure correction enhancement [2], which result from incorrect separation. Some examples of the incorrect separation are shown in Fig. 4.

Benchmark datasets are becoming important in computer vision. For most computer vision problems, like optical flow [4], stereo [5], object recognition [6], and edge detection [7], there exist datasets used as benchmarks for evaluation and comparison. These datasets have driven innovation and rigor to those problems. However, there is a lack of a common dataset for image base-detail separation at coarser levels of the hierarchy, in particular, at a part level. This makes it hard to draw conclusions and to compare quantitatively.

Thus, we generate a ground-truth base-detail dataset at part level in this paper. At part level, a fundamental assumption of this paper is that the base layers are piecewise smooth. It is piecewise because of the different objects and parts present in the image. For each part at part level, the separation results should not be affected by their neighboring parts, hence the methods should successfully preserve the sharp boundaries. Otherwise, halo artifacts
will be introduced into both of the base and the detail, as shown in Fig. 4.

To get the ground-truth data, we manually segment each image into parts, and for each part, we label the base and detail layers. Segmenting images into parts is challenging because the shapes of parts and their appearances vary a lot. Automatic segmentation is not ideal for producing ground truth datasets because it can fail in many cases, i.e., its segments might not respect the actual part or object boundaries. It is safer to rely on manual annotations of the segments at pixel level. Within each part, labeling the base-detail separation is also challenging because it requires pixel-wise annotation for the intensity of the base layer in the RGB color space (or, more generally, in any color space). We use a polynomial model of different orders to separate the image signal into several possible base layers and let humans select which order of the polynomial separation is the correct base layer separation. The residual of the selected base layer is the detail.

The main contributions of this paper are: 1) a quantitative evaluation of base-detail separation methods; 2) two natural image datasets providing base and detail ground truth (Pascal [8] with extended custom annotations and Fashionista [9]); and 3) the evaluation of several state-of-the-art base-detail separation algorithms based on the provided ground truth.

The rest of the paper is organized as follows. In section 2 we discuss the related work. In section 3 we describe the construction of our ground-truth dataset and proposed a way to evaluate algorithms based on it. In section 4 we present a baseline evaluation of several state-of-the-art methods. We conclude with section 5.

2 Related work

2.1 Image base-detail separation algorithms and applications

As described above, we argue that the separation of image base-detail layers is hierarchical. And different image base-detail separation algorithms [13, 14, 15, 16, 17, 18, 12, 19, 11, 20, 1, 21, 22, 23, 24, 25, 26, 27, 28] focus on separation at different levels of the hierarchy, aiming at different applications.

Base-detail is mainly a concept in computer graphics. We now give some examples of it. One example of image base-detail separation at the fine level is signal-noise separation, because the separated detail is of very high frequency. In noisy images the detail layer is occupied by noise while the base layer contains the image signal. Different algorithms have been proposed for applications of denoising [29, 16, 20], joint upsampling [18, 12], compression artifacts removal [18, 1], matting [19, 12], disparity estimation / optical flow [24], dehazing [12], etc.

Image base-detail separation at coarse level aims to separate the image’s basic coarse structures, such as the luminance and color of parts, and texture-like detail, like high frequency and mid-high frequency information. Different algorithms are proposed for applications of image base-detail separation at coarse level including exposure correction [2], HDR/tone mapping [30, 15, 17, 12, 23], face recognition and background correction for cDNA microarray images [31], style transfer [17, 24], segmentation [16], edge detection [1, 21], colorization/recolor [16, 11], rendering effects [11, 20, 24], edit propagation [19], etc.
Figure 4: Halo effects produced by different filters, (a) to (g), as opposed to the proposed ground-truth (h). Top row: base. Bottom row: detail, with increased contrast for visualization. Filters: (a) adaptive manifold, (b) domain transform, (c) L0 smooth, (d) rolling guidance, (e) bilateral, (f) guided, and (g) Gaussian. The different halo effects are easily observed between the sky and the vehicle, and most of them produce blurry boundaries in the base layer and a padding artifact in the detail layer. Another effect is that some methods represent the cloud as a broken texture (detail layer), and not as a blob-like structure (groundtruth detail, (h)).

2.2 Related datasets and quantitative evaluation

There are some quantitative evaluation methods for base-detail separation at fine level, like image denoising [29, 16, 12, 20], upsampling [18], and compression artifacts removal [18, 1]. For evaluation, the methods add spot noise to the original images, down-sample them or compress them to get noisy images. Then, they recover the noisy images and compute the difference between original images and recovered results. This can show the algorithms’ performance in separating the fine detail, such as spot noise or structured noise. But the performance of base-detail at fine level has little relationship with the performance at coarse level. This is because ignoring the piecewise smoothness assumption does not have big effects on the results at fine level, but can have a very big effect at coarse level. For example, artifacts like halos are frequently introduced when the base-detail separation of a part is contaminated by neighboring parts.

Figure 5: Similarity (top row) and difference (bottom row) between base-detail separation and intrinsic image decomposition. In base-detail separation the small shading patterns, like the ones on the teabag, are considered detail. Columns 1 to 3: Original, Base, Detail. Columns 4 and 5: Synthetic shading and texture [32].

In the field of intrinsic image separation, where intensity patterns are decomposed into shading and reflectance components [32, 33], there are some intrinsic image datasets, including the MIT Intrinsic Image dataset
and the Cornell Intrinsic Image dataset [33]. The MIT Intrinsic Image dataset uses computer graphics techniques to produce the ground-truth data. The Cornell Intrinsic Image dataset uses human annotation to get the ground-truth data of indoor images. The analogy between intrinsic images and base-detail is that, for an ideal image generated by adding sharp texture to smooth shading, base and detail would correspond to shading and texture. However, their problem, i.e. shading-texture separation, is different from our problem. In base-detail, the base layer may contain the basic structure of the shading components while the detail layer may also contain shading patterns and fine-level structures. An example of this is shown in the second row of Fig. 5. Another major difference is that base-detail separation depends on how coarse the base is defined to be, i.e. the hierarchy, while, ideally, intrinsic images should be independent of this.

3 The ground-truth base-detail separation dataset

Our goal of this paper is to construct a ground-truth base-detail separation dataset at coarse level. In other words, we separate images into base-detail at part level instead of pixel level, so the detail will be at a coarse level, e.g., in a person’s face the detail layer will contain the eyes, nose and mouth, or in a t-shirt, the detail will be its texture patterns and creases. We assume that the separated base layer is piece-wise smooth (validated by our annotators). It means that the separation of pixels of one part is only determined by the part itself, and the neighboring parts do not affect it. This avoids the halo effects between parts, see Fig. 3 and 4. Thus, to get the ground-truth base-detail separation dataset, 1) we proposed to use annotated images at part level, and 2) within each part, we annotate the base layer in the RGB color space by using a polynomial model to separate each part into several possible base layers and letting humans select the correct one.

3.1 Annotation for segmentation

The annotation for segmentation at part level consists of drawing the exact boundaries of each part in the image. Automatic segmentation is usually not a good approximation for this task. Fortunately, there exist some well-known datasets with this kind of annotations, like the Fashionista [9] and Pascal Part UCLA dataset [8]. They provide human-made annotations of parts at pixel level. We reuse their part-level annotations. In Fashionista, the labeled regions are parts of human beings, such as hair, glasses, shirt, shoes, socks, etc. The parts are appearance-based, rather than functional parts, e.g., “skin” and “shirt” are annotated as two classes, instead of a single class “arm” which could contain both skin and shirt appearances. In the Pascal Part UCLA dataset, the labeled images have many classes, such as dog, person, car, and background classes such as sky, grass, etc. Twenty of the most frequently appearing foreground objects have part annotations as well, such as the person’s “arm”, “head”, etc., or the car’s doors, windshield, and tires. In the Pascal dataset, regions are labeled according to semantic meaning and do not necessarily enforce the appearance consistency. In summary, the Fashionista dataset only has annotations of human beings images but the segmented parts have appearance consistency. In the Pascal Part UCLA dataset some of the part annotations might contain several regions with different visual appearances but it has the advantage of having all pixels annotated with more classes, and more annotated images than the Fashionista dataset. It is also worth insisting on the generality of the Pascal dataset, which offers a large variety of photographs, both indoors and outdoors. Thus, the results on the Pascal dataset are more likely to be extended to new images.

As mentioned above, the definition of a part is different in different applications and datasets because the base-detail separation is a hierarchical structure. This explains why the Fashionista and Pascal Part UCLA datasets used different strategies for parts annotation. In our opinion, this diversity of the labeling of parts is a good property because we can evaluate different base-detail separation algorithms in different conditions so that we can have a better understanding of the performance of different algorithms.

3.2 Annotation for base-detail separation

According to the piece-wise smoothness assumption, we separate images into base and detail layers within each part. The part comes from the ground-truth annotations. It is difficult for humans to directly annotate the base and detail layers in the RGB color space. Thus, we pro-
posed to first separate the image signals of each part into several possible results using polynomial models of different orders. Then, from the set of possible separation results, we let humans select which order of the polynomial results gives the best base-detail separation of each part. This can work for a subset of the images in the whole dataset, for which the base layer can be described by some order of the polynomial model. For the other images, we remove them from the dataset. Thus, from the whole datasets of Fashionista and Pascal, we get a subset as the ground-truth dataset of our work. Our strategy reduces the labeling work a lot and makes it much easier to generate the ground-truth datasets for base-detail separation at part level.

To annotate the base layer within each part, the first step is to separate the image signals of each part into several possible base layers with the polynomial model with different orders. Within each part \( P_i \), we fit polynomial models on each color channel separately. The number of parameters \( \omega \) depends on the order \( k \) of the polynomial. The polynomial approximations are of form:

\[
\begin{align*}
    b_k(\vec{x}, \vec{\omega}) &= \vec{x}^T \vec{\omega} \\
    k = 0 : \vec{x} &= 1, \vec{\omega} = [\omega_0]
    \\
    k = 1 : \vec{x} &= [1, x_1, x_2], \vec{\omega} = [\omega_0, \omega_1, \omega_2]
    \\
    k = 2 : \vec{x} &= [1, x_1, x_2, x_1^2, x_2^2, x_1 x_2], \vec{\omega} = [\omega_0, ..., \omega_5]
    \\
    k = 3 : \vec{x} &= [1, x_1, x_2, x_1^2, x_2^2, x_1 x_2, x_1^3, x_2^3, x_1 x_2^2, x_1^2 x_2], \vec{\omega} = [\omega_0, ..., \omega_9]
\end{align*}
\]

The estimation of the parameters \( \vec{\omega} \) of the polynomial is performed by linear least squares QR factorization [34]. We limit our polynomial approximations to the third order, \( k = 3 \), to prevent over-fitting the data. See Fig. 6, where different segments are approximated by different orders \( k \).

The second step is to select the ground-truth base layer for each part. For each part, there are four possible ground-truth base layers, and we let annotators select the ground-truth base layer by choosing one from the four layers. When none of them is correct, annotators will mark it and the image will be removed from the dataset.

After the previous two steps, we can get the ground-truth base layer with the polynomial model’s separation results and the annotation of annotators on each part. For an image, if the base layer of even one part cannot be described by any of the polynomial results, this image will be rejected. In total, we select about 1000 and 250 images in the Pascal and Fashionista datasets, respectively.

## 4 Evaluation

### 4.1 Ground truth datasets

The ground-truth datasets that we use are the subset of images from the Fashionista and Pascal Part UCLA datasets, as described in Sec. 3. For simplification, in this paper, we still call the subset of images as the Fashionista and Pascal Part UCLA datasets. See examples of both datasets in Fig. 7 and Fig. 8.

### 4.2 Separation methods

The separation methods we use in the evaluation include the Gaussian filter, bilateral filter (BL) [3], guided filter (GD) [12], adaptive manifold filter (AM) [10], domain transform filter (DT) [11], L0 smooth filter (L0) [1], rolling guidance filter (RG) [25]. The Gaussian filter is a linear filter, and the smoothing only considers the distance between neighboring pixels. The bilateral filter improves Gaussian filter to an edge preserving filter because the smoothing considers both the distance and the intensity difference between neighboring pixels. The guided filter performs a linear transform within a local window and applies the linear transform to all local windows to get the output. The rolling guidance filter performs scale-aware separation, similar with our hierarchical base-detail structure. The L0-smooth filter is an optimization based filter. It uses L0 gradient minimization, which globally locates

![Figure 6: Different polynomial orders \( k \) in the same base layer. Left: original. Center: polynomial base approximation. Right: order of the polynomial. Dark-blue: \( k = 0 \), light-blue: \( k = 1 \), yellow: \( k = 2 \); red: \( k = 3 \).](image)
important edges instead of considering local features. The adaptive manifold filter performs high-dimensional filtering and is flexible, capable of approximating bilateral, Gaussian or other filters. The domain transform filter transforms 2D image to a real line and filter the transformed signal with a 1D kernel.

4.3 Error metric

A direct way for evaluation is to compute the mean squared error (MSE) between ground-truth base-detail layers and estimated base-detail layers. MSE is defined as:

$$MSE(J_1, J_2) = \frac{1}{\sum_{i,c}} \sum_{i,c} (J_1(i,c) - J_2(i,c))^2,$$

where $J_1$ and $J_2$ are two images, $i$ is the pixel position, $c$ is the color channel in the RGB space.

However, we found that between well-exposed images and low lighting images, the same amount of error will
cause much different MSE values. For example, the intensities of a pixel in a well-exposed image and low lighting image are 200 and 20, respectively. If the errors are the same, for example, the estimation values are 210 and 30, respectively, the MSE values of the estimation will be 4100 and 500, respectively. The reason is for well-exposed images, because the RGB intensities of pixels are high, some small errors will lead to large MSE values. But for low lighting images, because the RGB intensities are low, even if there exist large errors, the MSE will not be very high. So, directly using MSE will lead to bias to the evaluation results and the errors of well-exposed images will have more weights.

To avoid the bias, we proposed the relative mean squared error (RMSE) as the error metric between the ground-truth base-detail layers and the estimated base-detail layers. The metric is similar with the local mean squared error (LMSE) [32]. RMSE is defined as

$$\text{RMSE}(B_{GT}, D_{GT}, B_E, D_E) = \frac{1}{2} \left( \frac{\text{MSE}(B_{GT}, B_E)}{\text{MSE}(B_{GT}, 0)} + \frac{\text{MSE}(D_{GT}, D_E)}{\text{MSE}(D_{GT}, 0)} \right)$$

where, $B_{GT}$ is the ground truth base layer, $D_{GT}$ is the ground-truth detail layer, $B_E$ is the estimated base layer, and $D_E$ is the estimated detail layer. Because 1) RMSE considers errors of both detail and base layers, and 2) for each layer, it measures the relative error, i.e. the ratio of $\text{MSE}(GT, E)$ and $\text{MSE}(GT, 0)$, it can avoid the bias between low lighting images and well exposed images. According to the definition of RMSE, if the RMSE value is lower, the estimation base and detail layers are more accurate. If the RMSE value is 0, the estimation layers are exactly the same as the ground truth layers.

### 4.4 Algorithms and parameter settings

For an input image, we use different algorithms to smooth it to get the base layers. Then, we compute the RMSE between the filtered result and the ground-truth data.

The seven separation methods have parameters to control the smoothing. For different goals, like separation at coarse level or fine level, the parameters should be different. In general, high values of the parameters tend to mean coarse level smoothing. The values of the parameters are usually set experimentally. Here, we select the best parameters for each filter to enable a fair comparison between them. We use the same parameter value for the whole dataset (one parameter setting for the Pascal Parts UCLA and another for the Fashionista datasets). The parameters range of the filters and the optimal parameters for the datasets are shown in Table 1. The parameters for the Fashionista and Pascal datasets are different, and the reason is that, as described in Sec. 3.1, these two datasets used different strategies for part annotation. The segmented parts of the Pascal dataset are usually coarser than those of the Fashionista dataset. As a result, as shown in Table 1, the optimal parameters for the Pascal dataset are usually larger than those for the Fashionista dataset.

### 4.5 Results

For all images, we first average each separation method’s RMSE scores, with all images weighted equally. Secondly, we rank all the algorithms on each image according to the RMSE scores and average the ranks. The reason of using the average rank is that ranking is more robust in the case of images with extreme level of difficulty. The results are shown in Fig. 10. Some example results of each method can be seen in Figs. 4, 9.

### 4.6 Analysis

In the experiments, we can see that most of the edge-preserving filters perform better than the Gaussian filter (we can consider the Gaussian filter as a classical baseline). This is because the Gaussian filter does not preserve edges, and the high parameters (e.g., the variance) lead to the smoothing results of each part affected heavily by neighboring parts. Most of the other filters preserve edges better than the Gaussian filter, so they have better performance.

The bilateral filter is edge-preserving, and consistently with the standard intuition, so it performs better than the Gaussian filter. The adaptive manifold filter and domain transform filter have better performance than the bilateral filter on average because they are flexible, and have more potential to perform well in our datasets if the parameters are selected carefully. The rolling guidance filter and guided filter also have good performances in the experiments.
### Table 1: Parameters settings of different filters for the Fashionista and Pascal Part UCLA datasets.

| Alg. | $\theta_1$ | $\theta_2$ | Ranges of parameters $\theta_1$ | Optimal parameters ($\theta_1, \theta_2$) |
|------|-------------|-------------|---------------------------------|----------------------------------------|
| AM   | 2,4,8,16,32,64 | 0.1,0.2,0.4,0.8,1.2 | (8,0.8) | (32,0.8) |
| DT   | 4,8,16,32,64,128 | 0.1,0.2,0.4,0.8,1.2 | (16,2) | (64,2) |
| L0   | 0.005,0.01,0.02,0.04,0.08,0.1,0.2,0.4,0.8,1 | | 0.08 | 0.1 |
| RG   | 2,4,8,16 | 0.01,0.02,0.04,0.08,0.1,0.2,0.4 | (8,0.1) | (16,0.2) |
| BL   | 2,4,8,16,32,64 | 0.1,0.2,0.4,1.2,4 | (8,0.4) | (32,0.4) |
| GD   | 4,8,16,32,64,128 | 0.01,0.02,0.04,0.1,0.2,0.4 | (16,0.1) | (64,0.1) |
| GS   | 4,8,16,32,64,128 | 1.2,4,8,16,32 | (32,4) | (64,4) |

Left: The parameters range in our experiments for the adaptive manifold filter (AM), the domain transform filter (DT), the L0 smooth filter (L0), the rolling guidance filter (RG), the bilateral filter (BL), the guided filter (GD), and the gaussian filter (GS). For GS and GD, $\theta_1, \theta_2$ denote window size and $\sigma$ variance, respectively. For BL, AM, DT, and RG, $\theta_1, \theta_2$ denote $\sigma$ spatial and $\sigma$ range, respectively. For L0, $\theta_1$ denotes $\lambda$. Right: The optimal parameters of the seven methods for the Fashionista and Pascal Part UCLA datasets.

Figure 9: Results of the base-detail separation methods tested on an example image of the Fashionista dataset. Top: base; Bottom: detail. See Table 1 with the names of the algorithms and their parameters. Last column (GT) is the proposed ground-truth.
The L0 smooth filter uses gradient to separate base and detail. However, in our datasets, the parts are segmented semantically, and areas with large gradient do not always mean the edges of parts. In addition, among different images, the strength of the edges between parts are also different. Using the uniform parameter to filter thousands of images results in poor performance.

From the results of both the Pascal and Fashionista datasets, we can see that the average RMSE results and the average rank results of the filters are consistent between the datasets.

5 Conclusions

Quantitative evaluations are fundamental to the advances of any research field. The base-detail ground-truth datasets we provide here are a necessary starting point for extensive quantitative comparisons of filtering and base-detail algorithms. We argue that, ideally, base-detail annotation should be hierarchical and hand-made, and we proposed an intermediate solution, which is practical (i.e., ready for use) by now and extensible in the future. The images in our datasets are annotated at part level, coming from the Pascal Part UCLA dataset (complete labeling) and the Fashionista dataset.

While the piecewise smoothness of the base layer is the most obvious solution to the halo effects, the modeling within each part seems arguable. Many plausible models can be formulated, as long as they can handle smooth appearance changes due to illumination, shape, etc. Here we choose a simple one: polynomial approximations.

It is understandable that different filtering methods may differ in their objectives and applications. Still, performing extensive quantitative benchmarking can bring important progress to the field.

In the future we would like to extend the annotations to a deeper level in the hierarchy and perform psychophysics validation experiments.

References

[1] L. Xu, C. Lu, Y. Xu, and J. Jia, “Image smoothing via l0 gradient minimization,” TOG, 2011. 2, 3, 4, 6
[2] L. Yuan and J. Sun, “Automatic exposure correction of consumer photographs,” Proc. European Conference on Computer Vision, pp. 1–14, 2012. 2, 3
[3] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, “Bilateral filtering: Theory and applications,” Foundations and Trends in Computer Graphics and Vision, vol. 4, no. 1, pp. 1–73, 2008. 2, 6
[4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, “A naturalistic open source movie for optical flow evaluation,” ECCV, 2012. 2
[5] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” IJCV, 2002. 2
[6] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories,” CVPR Workshop on Generative Model Based Vision, 2004. 2
[7] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” TPAMI, vol. 33, no. 5, pp. 898–916, 2011. 2
[8] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille, “Detect what you can: Detecting and representing objects using holistic models and body parts,” 2014. 3, 5
[9] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg, “Parsing clothing in fashion photographs,” CVPR, 2012. 3, 5
[10] E. S. L. Gastal and M. M. Oliveira, “Adaptive manifolds for real-time high-dimensional filtering,” TOG, 2012. 3, 6
[11] ———, “Domain transform for edge-aware image and video processing,” TOG, 2011. 3, 6
[12] K. He, J. Sun, and X. Tang, “Guided image filtering,” ECCV, 2010. 3, 4, 6
[13] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, “Edge preserving decompositions for multi-scale tone and detail manipulation,” TOG, 2008. 3
[14] R. Fattal, M. Agrawala, and S. Rusinkiewicz, “Multiscale shape and detail enhancement from multi-light image collections,” TOG, 2007. 3
[15] K. Subr, C. Soler, and F. Durand, “Edge-preserving multiscale image decomposition based on local extrema,” TOG, 2009. 3
[16] A. Criminist, T. Sharp, C. Rother, and P. Perez, “Geodesic image and video editing,” TOG, 2010. 3, 4
[17] M. Aubry, S. Paris, S. W. Hasinoff, J. Kautz, and F. Durand, “Fast local laplacian filters: Theory and applications,” TOG, 2014. 3
[18] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless, “Gradienteshop: A gradient domain optimization framework for image and video filtering,” TOG, 2010. 3, 4
[19] Z. Farbman, R. Fattal, and D. Lischinski, “Diffusion maps for edge-aware image editing,” TOG, 2010. 3
[20] L. Karacan, E. Erdem, and A. Erdem, “Structure preserving image smoothing via region covariances,” TOG, 2013. 3, 4
[21] L. Xu, Q. Yan, Y. Xia, and J. Jia, “Structure extraction from texture via relative total variation,” TOG, 2012. 3, 4
[22] M. Kass and J. Solomon, “Smoothed local histogram filters,” TOG, 2010. 3
[23] S. Paris, S. W. Hasinoff, and J. Kautz, “Local laplacian filters: Edge-aware image processing with a laplacian pyramid,” TOG, 2011. 3
[24] Q. Zhang, L. Xu, and J. Jia, “100+ times faster weighted median filter,” CVPR, 2014. 3, 4
[25] Q. Zhang, X. Shen, L. Xu, and J. Jia, “Rolling guidance filter,” ECCV, 2014. 3, 6
[26] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” PAMI, 1990. 3
[27] J. van de Weijer and R. van den Boomgaard, “Local mode filtering,” CVPR, 2001. 3
[28] B. Weiss, “Fast median and bilateral filtering,” TOG, 2006. 3
[29] T. Brox and D. Cremers, “Iterated nonlocal means for texture restoration,” SSVM, 2007. 3, 4
[30] F. Durand and J. Dorsey, “Fast bilateral filtering for the display of high dynamic range images,” TOG, 2002. 3
[31] W. Yin, D. Goldfarb, and S. Osher, “Image cartoon-texture decomposition and feature selection using the total variation regularized l1 functional,” VLSM, 2005. 3
[32] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman, “Ground truth dataset and baseline evaluations for intrinsic image algorithms,” in Proceedings of the International Conference on Computer Vision (ICCV), 2009. 4, 5, 8
[33] S. Bell, K. Bala, and N. Snavely, “Intrinsic images in the wild,” ACM Trans. on Graphics (SIGGRAPH), vol. 33, no. 4, 2014. 4, 5
[34] G. H. Golub and C. F. V. Loan, “Matrix computations,” JHU Press, vol. 3, 2012. 6
This figure "fig1.png" is available in "png" format from:

http://arxiv.org/ps/1511.06830v1
This figure "fig2.png" is available in "png" format from:

http://arxiv.org/ps/1511.06830v1
This figure "fig3.png" is available in "png" format from:

http://arxiv.org/ps/1511.06830v1
This figure "fig4.png" is available in "png" format from:

http://arxiv.org/ps/1511.06830v1
This figure "fig5.png" is available in "png" format from:

http://arxiv.org/ps/1511.06830v1
This figure "fig6.png" is available in "png" format from:

http://arxiv.org/ps/1511.06830v1
This figure "fig7.png" is available in "png" format from:

http://arxiv.org/ps/1511.06830v1
This figure "fig8.png" is available in "png" format from:

http://arxiv.org/ps/1511.06830v1
This figure "fig10.png" is available in "png" format from:

http://arxiv.org/ps/1511.06830v1