

带有客观评估的像素级别非局部图像平滑

Jun Xu[#], Zhi-Ang Liu[#], Ying-Kun Hou, Xian-Tong Zhen, Ling Shao, Ming-Ming Cheng^{*}

摘要—最近, 图像平滑由于其在其他图像处理任务 (例如, 图像增强和编辑) 中的先决作用而受到越来越多的关注。但是, 图像平滑算法的评估通常是通过没有相应真实值的图像进行主观观察来进行的。为了促进图像平滑算法的发展, 在本文中, 我们构建了一个新颖的 Nankai Smoothing (NKS) 数据集, 其中包含 200 图像, 并混合了多种结构图像和自然纹理。结构图像本质上是平滑的, 自然会被视为真实值。在我们的 NKS 数据集上, 我们全面评估了 14 种流行的图像平滑算法。此外, 我们提出了一种像素级非局部平滑 (PNLS) 方法, 通过利用自然图像之前的像素级非局部自相似性来很好地保留平滑图像的结构。在几个基准数据集上进行的大量实验表明, 我们的 PNLS 在图像平滑任务上优于以前的算法。消融研究还揭示了我们的 PNLS 在图像平滑上的工作机制。为了进一步证明其有效性, 我们将 PNLS 应用于多种应用程序, 例如语义区域平滑, 细节/边缘增强和图像抽象。数据集和代码在 <https://github.com/zal0302/PNLS>。

Index Terms—图像平滑, 基准数据集, 性能评估, 像素级非局部自相似性。

I. 引言

图像平滑是一项重要的多媒体技术, 旨在将图像分解为分段的平滑 (结构) 层和细节 (纹理) 层 [14]。光滑层反映图像的结构内容, 而纹理层显示图像中的残留细节。分解后的图层可以分别进行操作, 并以不同的方式进行重组, 以实现诸如图像增强 [22], [45], 图像消除 [10], [11] 和图像抽象 [5], [54] 等特定应用。

在过去的十年中, 从局部滤波器 [31], [34], [37], 全局滤波器 [33], [55], [58], 和深层滤波器 [21], [27], [53]

This work was supported in part by the Major Project for New Generation of AI under Grant No. 2018AAA0100400, the Fundamental Research Funds for the Central Universities, Nankai University (63201168) and Project (92022104), National Natural Science Foundation of China (61922046), and Tianjin Natural Science Foundation (18ZXZNGX00110). Jun Xu and Zhi-Ang Liu are joint first authors and contributed equally to this work. Ming-Ming Cheng (cmm@nankai.edu.cn) is the corresponding author.

J Xu, ZA Liu, and MM Cheng are with TKLNDST, College of Computer Science, Nankai University, Tianjin, China. YK Hou is with School of Information Science and Technology, Taishan University, Tai'an, China. XT Zhen and L Shao are with Inception Institute of Artificial Intelligence and Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE.



图 1. 我们 NKS 数据集中的一些例子。

等的角度提出了许多图像平滑算法。局部滤波器通过以局部加权的方式平均像素强度来平滑输入图像 [19], [34], [37]。它们的计算效率很高, 但是会产生梯度反转和光晕伪影, 尤其是在边缘上 [19]。全局滤波器 [14], [55], [58] 通过以一种原则的方式对整个图像进行优化来减弱逆转和伪影。但是, 全局滤波器通常会浪费时间和内存 [59]。深层滤波器 [21], [27], [53] 用成对的自然和“真实值”图像训练平滑网络。但是“真实值”图像通常是由其他图像平滑方法生成的 [52], [59], 从而阻碍了深度滤波器的域广义性能。

尽管它们具有令人鼓舞的性能, 但是由于缺乏合理的基准, 很难对这些图像平滑算法进行客观评估。尽管为图像平滑任务收集了几个数据集 [30], [52], [59], 他们的“真实值”要么是通过其他图像平滑方法生成的 [52], [59], 要么是与卡通图像和合成纹理混合 [30]。一方面, 现有平滑方法生成的“真实值”存在很大偏差。也就是说, 基于“真实值”计算的度量结果不反映平滑方法的平滑性能, 而是其结果与几种平滑方法生成的“真实值”的接近度。另一方面, 在卡通图像和合成纹理上训练的算法在平滑不同于训练数据的自然纹理方面不能始终如一地表现良好。为了促进图像平滑算法的发展, 在这项工作中, 我们构建了一个新颖的 Nankai Smoothing (NKS) 数据集, 其中包含 200 个多用途图像, 这些图像由结构图像 (真实值) 和自然纹理混合而成。

图中显示了我们的 NKS 数据集的一些示例 1. 在我们的 NKS 数据集上, 我们对 14 种流行的图像平滑算法进行了基准测试, 并使用常用的指标进行了广泛的性能分析。此外, 我们基于自然图像之前的像素级非局部自相似性 (NSS), 提出了一种像素级非局部平滑 (PNLS) 方法 [20]. 在几个基准上进行了广泛的实验数据集 (包括我们的 NKS) 表明, 我们的 PNLS 在客观指标和主观视觉质量上比以前的图像平滑方法具有更好的性能。为了显示其广泛的实用性, 我们将 PNLS 应用于四个图像处理任务: 显着区域平滑, 图像细节增强, 图像边缘增强和图像抽象等。总而言之, 我们的主要贡献是多方面的:

- 我们构建一个 **Nankai Smoothing (NKS)** 数据集包含 200 个按结构和纹理图像混合的图像。结构图像自然被视为评估图像平滑方法的基础。我们还在 NKS 数据集上对 14 种流行的图像平滑算法进行了基准测试。
- 我们提出了一种新颖的像素级非局部平滑 (PNLS) 方法通过利用自然图像之前的像素级非局部自相似性 (NSS)。
- 实验表明, 我们的 PNLS 实现了令人满意的图像平滑性能在几个基准数据集上, 通过主观评估和客观指标。通过将其应用于各种图像处理任务, 我们还展示了 PNLS 的广泛实用性。

本文的其余部分安排如下。在 §II, 我们审查相关工作。在 §III, 我们介绍了 Nankai Smoothing (NKS) 数据集, 并在其上对 14 种流行的图像平滑算法进行了基准测试。然后, 我们在下面介绍 PNLS 平滑方法 §IV. 在 §V, 我们在几个数据集上进行了广泛的实验, 以证明 PNLS 与以前的平滑方法相比的优势。我们还提供 PNLS 在以下几种图像处理任务上的更多应用 §VI. 结论在 §VII.

II. 相关工作

A. 图像平滑方法

局部滤波器以一步或迭代方式显式过滤每个像素作为其邻域像素的加权平均值。双边滤波器 (BF) [31] 是这一类别中的一种简单直观的方法, 并广泛应用于其他图像处理任务 [16], [24], [43]。但是, 由于并非所有像素周围都有足够的相似像素, 因此加权平均值将受到异常像素的影响, 从而导致梯度反转伪影 [19]. 它也被概括为联合双边滤波器 (JBF) [34], 其中权重是在另一个引导

图像上而不是在输入图像本身上计算的。借助指导图像的观点后来在导向滤波器 (GF) [19] 中快速发展。滚动导向滤波器 (RGF) [57] 通过完全控制细节平滑过滤图像, 使用 BF 来进行滤波, 并且以迭代的方式实现滚动引导。GF 启发了很多方法, 因为对于一张 N 个像素点的图像来说, 它的复杂度是 $O(N)$ 的。然而, 它不能解决关于是否平滑某些边缘的模糊性。借助 ℓ_0 梯度最小化 [52] 进行校正, Su *et al.* [37] 使用降级方案来平滑小规模纹理, 并使用联合双边滤波器来抑制纹理。

全局滤波器 [14], [52], [54] 减弱局部滤波器的局限性, 例如梯度反转和光晕伪影 [14]。这些方法原则上在整个图像上解决了优化功能。该函数通常由用于数据拟合的保真度项和用于使平滑度正规化的先前项组成。在这些方法中, 加权最小二乘 (WLS) [14] 根据图像梯度调整矩阵亲和力, 并产生无光晕的平滑结果。后来, 提出了 WLS [28] 的半全局扩展, 以节省时间和内存的方式求解线性系统。 ℓ_0 梯度最小化 [52] 全局控制非零渐变的数量, 这些渐变与逼近输入图像的突出结构有关。但是, 一个不可避免的问题是, 在平滑细节时, 它们容易过度磨削边缘 [14], [28], [52]. Zhou *et al.* [58] 提出了一种迭代优化滤波器, 可以选择性地抑制较小尺度特征的梯度, 同时在有限的迭代中保留大规模强度变化。简而言之, 全局滤波器通常在计算上很昂贵, 并且常常牺牲局部边缘保留效果以获得更好的全局性能。

深层滤波器在着眼于最先进的局部或全局滤波器 (例如 BF [31], WLS [14]) 的同时, 着重于加速。深度边缘感知滤波器 (DEAF) [53] 是该类别中的开拓性工作。它在梯度域中训练网络, 并从深层网络产生的精细梯度中重建滤波后的输出。在 [27] 中, 提出了一种基于递归滤波器的混合神经网络, 其系数可以通过深度网络进行学习。Li *et al.* [21] 提出了一种基于学习的方法来构建基于卷积神经网络 (CNN) 的联合滤波器。

在 [12] 中, 作者首先使用网络来基于梯度预测边缘/结构置信度图, 然后使用其输出来指导滤波网络。Shen *et al.* [36] 引入了卷积神经金字塔网络来提取不同比例的特征, 并从输入图像中获得更大的接收场。[5] 的工作利用上下文聚合网络来包含更多上下文信息。Lu *et al.* [30] 开发了结构和纹理数据集并训练了纹理和结构感知网络。[13] 的工作引入了无监督的学习 CNN, 它有助于产生灵活的平滑效果。一个普遍的问题是, 所有这些方法都将现有滤波器的输出视为“真实值”, 因

此很难超越这些“导师”滤波器。

B. 图像平滑基准

数据集. 最初收集了 BSDS500 [32], DIV2K [1], MIT5K [3] 的几个数据集用于图像分割和图像恢复 [11], [23], [35] 任务, 但也用于表示图像平滑性能。但是这些图像没有相应的平滑基础事实。使用提议的图像平滑算法 RTV [54] 发布了一个数据集, 但是该数据集也没有真实值。Zhu *et al.* [59] 构造了图像平滑数据集。遗憾的是, 地面真值平滑图像是由现有的平滑算法生成的。这些“真实值”易于主观化, 因为实际上我们只能评估新算法与这些精选现有算法之间的性能差异。在我们的 Nankai Smoothing (NKS) 数据集中, 我们收集结构图像作为真实值, 并通过混合图像获取样本。

评估指标. 峰值信噪比 (PSNR) 是在图像恢复任务中广泛使用的客观指标, 用于计算原始图像和变形图像之间的误差。但是, PSNR 专注于两个图像之间的像素级差异, 而忽略了它们在视觉特性上的相似性 [42]。为了填补这一空白, 开发了结构相似度索引 (SSIM) [42], 以从亮度, 对比度和结构方面全面测量图像相似度。SSIM 考虑到结构块的相关性而不是像素, 因此更符合人眼对图像质量的判断。由于并非图像中的所有像素都具有相同的重要性, 因此特征相似度索引 (FSIM) [56] 使用低级特征来评估参考图像和失真图像之间的距离。一种常见的评估方式是通过视觉质量主观评估平滑图像, 但是这种方式可能缺乏准确的测量值。尽管在 [59] 中使用了加权均方根误差 (RMSE) 和加权均值绝对误差 (WMAE), 但它们可能会被现有平滑方法产生的“真实值”所困扰。在本文中, 我们将 PSNR, SSIM [42] 和 FSIM [56] 用作评估指标, 因为它们具有与人的视觉感知一致的性能。

III. 提出的 NANKAI SMOOTHING DATASET

在本部分中, 我们将为图像平滑任务开发一个 Nankai Smoothing (NKS) 数据集, 并使用三种常用指标, 即 PSNR, SSIM [42], 和 FSIM [56]。

A. 构建 NKS 数据集

动机. 由于手动注释图像的结构是主观且昂贵的, 因此直接从自然图像中提取结构真实值是困难的。实际上, 图像平滑非常接近图像去噪任务 [25], [26], [40]: 两者的

目的都是要从图像中滤除小规模成分 (纹理或噪声)。尽管对于图像平滑 [30], [59], 真实值的收集尚不明确, 但构建图像去噪数据集 [39], [47], [48] 显然是可行的。也就是说, 我们将噪点添加到干净的图像中, 并通过去除噪点从合成噪点图像中恢复出来 [46], [49]–[51]。在评估中, 自然会以相应的干净图像作为基本事实来计算客观指标, 例如 PSNR 和 SSIM [42]。类似地, 在图像平滑中, 我们可以混合结构和纹理图像以生成测试图像, 并且可以通过从混合图像中删除纹理来恢复结构图像。

尽管 [30] 的作者朝着这个方向努力, 但它有两个局限性: 1) 合成纹理不像真实世界的彩色图像那样自然; 2) 该数据集未公开发布, 因此其他人很难评估该数据集上的新颖平滑方法。

收集结构和纹理图像. 如图 2 所示, 我们观察到矢量图像是平滑的, 可以合理地用作结构图像, 并通过将矢量图像和纹理图像混合来构造 NKS 数据集。具体来说, 我们在 Pixabay 网站 [7] 上搜索“vector”的关键词, 并从数千个自由结构图像中选择 20 个高度逼真的矢量图像。此外, 我们从 Pixabay 网站 [7] 上手动选择了 10 个自然纹理图像。所选的结构图像和自然纹理分别显示在 Figures 2 和 3 中。我们观察到矢量图像是平滑的, 结构清晰, 因此可以视为图像平滑中的真实值。

生成混合图像. 为了生成混合的结构和纹理图像, 我们以适当的比例混合了 20 个结构图像和 10 个自然纹理。我们将结构图像的比例设置为 0.7 到 0.85, 以确保每个融合图像足够真实。该过程与图像降噪中的噪声图像的生成非常相似 [44], [46], [51], 只是我们需要设置适当比例的结构和纹理图像以使混合图像的视觉质量自然。原因是, 将结构和纹理图像直接相加会导致像素值溢出, 以及混合图像的外观不自然。将结构图像作为由结构图像和 10 种自然纹理混合而成的相应图像的真实值。这样, 我们在 NKS 数据集中收集了总共 $20 \times 10 = 200$ 个图像, 其中有 20 个结构图像作为真实值。

数据集统计. 我们的 NKS 数据集包含 200 种通用场景的图像。我们在 Figures 2 和 3 中显示了结构和纹理图像, 从中可以看到我们的 NKS 数据集包含多种内容, 例如人 (比如, 儿童, 妇女, ...), 人工制品 (比如, 锅, 粉笔, ...), 风景 (比如, 森林, 海滩, ...) 和动物 (比如, 猫, 昆虫……), 等等。我们还在表格中显示了我们的 NKS 数据集的统计信息 I。通过默认的 Matlab 函数“imresize”, 所有图像都按比例调整为 512 的宽度。

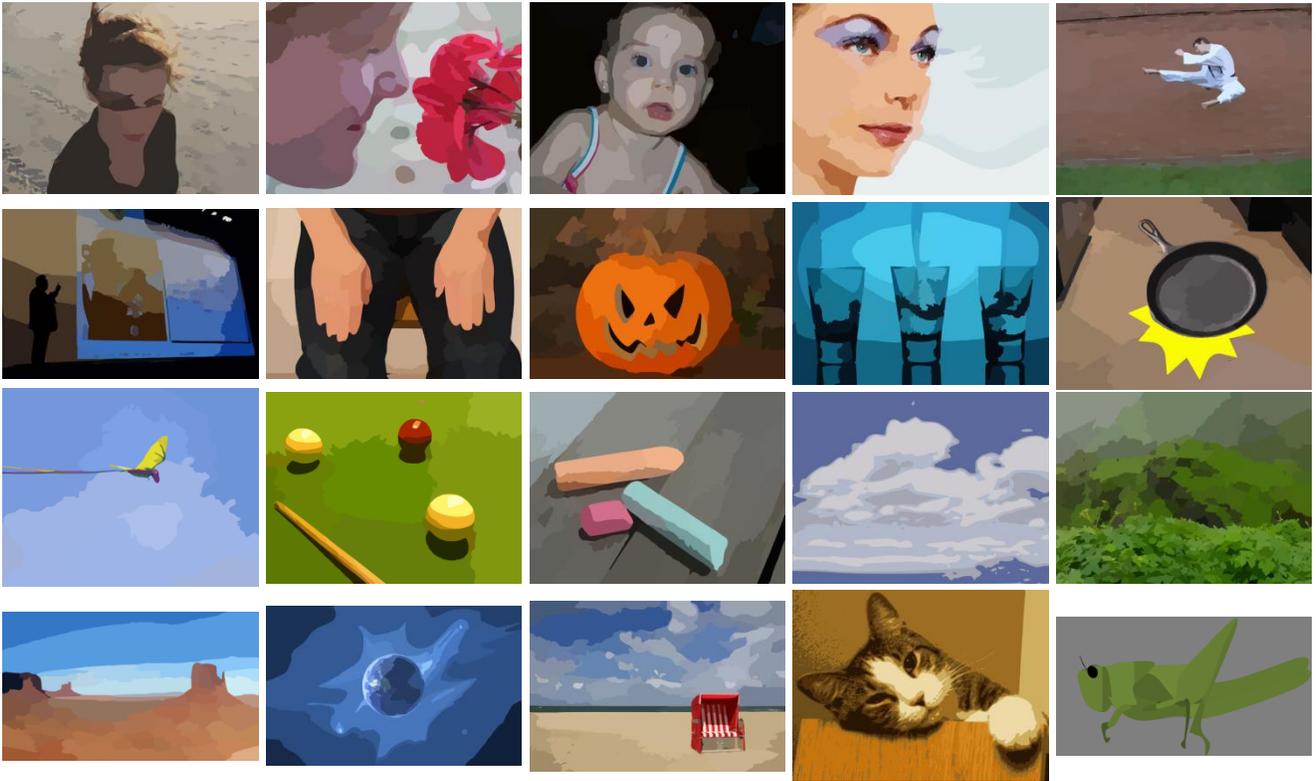


图 2. 我们在 NKS 数据集中使用了 20 个结构图像。



图 3. 我们在 NKS 数据集中使用了 10 个自然纹理图像。

表 I
我们 NKS 数据集的总结

种类	大小		数量	结构	纹理
	宽度	高度			
人类	512	340 ~ 384	70	向量	地毯, 木, ...
人工制品	512	343 ~ 397	50		
风景	512	298 ~ 384	60		
动物	512	277 ~ 384	20		

不同类别的数目如下，人为 70，人工制品为 50，风景为 60，动物为 20。

B. 在 NKS 数据集上对图像平滑进行基准测试

对比方法. 我们总共在 NKS 数据集上评估了 14 种图像平滑算法。这些算法包括 10 种传统滤波器: BF [31], WLS [14], EAW [15], GF [19], L0 [52], RTV [54], TF [2], FGS [33], RGF [57], fastABF [17] 和 4 种深层滤波器: LRNN [27], FIP [5], 在 [59] 中使用的两个 baseline ResNet 和 VDCNN。我们使用 PSNR, SSIM [42] 和 FSIM [56] 的常用指标来定量评估 NKS 数据集上比较方法的性能。这些度量标准衡量的是平滑图像与相应真实值之间的距离。

表 II

通过我们的 NKS 数据集上的 14 种最新图像平滑算法, 比较了平均 PSNR, SSIM [42] 和 FSIM [56]。这些方法 (在 CPU 或 GPU) 在 NKS 数据集中的 110 张大小为 512×384 图片上平均运行时间 (单位是秒) 如下。最好的和次好的结果被分别用红色和蓝色标记出来了。我们将会在 §IV 部分介绍 PNLs。“Pub.”表示“出版地点”。

	No.	Method	PSNR	SSIM	FSIM	Device	Time
Local Filters	1	BF [31] _{ICCV'98}	32.00	0.8478	0.8556	CPU	1.53
	2	EAW [15] _{TOG'09}	28.02	0.7953	0.8200	CPU	0.02
	3	GF [19] _{TPAMI'13}	32.09	0.8779	0.8672	CPU	0.03
	4	RGF [57] _{ECCV'14}	32.51	0.9135	0.9128	CPU	0.22
Global Filters	5	WLS [14] _{TOG'08}	28.59	0.9011	0.9107	CPU	0.91
	6	L0 [52] _{TOG'11}	33.01	0.9249	0.9374	CPU	0.48
	7	RTV [54] _{TOG'12}	31.81	0.9206	0.9234	CPU	0.68
	8	TF [2] _{TIP'13}	33.23	0.9186	0.9149	CPU	0.33
	9	FGS [33] _{TIP'14}	23.46	0.8368	0.7978	CPU	0.03
	10	fastABF [17] _{TIP'18}	31.44	0.8977	0.8917	CPU	0.37
Deep Filters	11	LRNN [27] _{ECCV'16}	30.61	0.8666	0.8600	CPU	0.43
	12	FIP [5] _{ICCV'17}	32.03	0.8946	0.9061	GPU	0.45
	13	VDCNN [59] _{TIP'19}	33.38	0.9349	0.9395	GPU	1.47
	14	ResNet [59] _{TIP'19}	33.13	0.9354	0.9434	GPU	3.76
Ours	15	PNLS (Fast)	33.45	0.9378	0.9397	CPU	5.10
	16	PNLS (Slow)	33.68	0.9420	0.9440	CPU	78.68

结果. 比较结果列于表中 II. 可以看到, 在 PSNR, SSIM [42] 和 FSIM [56] 的三个指标上, TF [2] 在传统 (局部和全局) 滤波器中表现最佳, 而 [59] 中的 baselines VDCNN 和 ResNet 的性能要优于其他两个深层滤波器。我们观察到, 与 WLS [14] 相比, BF [31] 可获得更高的 PSNR 结果, 但 SSIM [42] 和 FSIM [56] 结果却较低。这是合理的, 因为三个评估指标旨在从不同方面测量两个图像的相似性。具体来说, 峰值信噪比 (PSNR) 计算平滑图像的同位置处的像素与相应的真实值之间的距离。但是, 人眼通常对像素级差异具有鲁棒性, 但对结构差异敏感。即, PSNR 不能反映人眼的视觉特性。因此, PSNR 结果通常与主观观察不一致。为此, 我们求助于易于结构化的指标, 例如

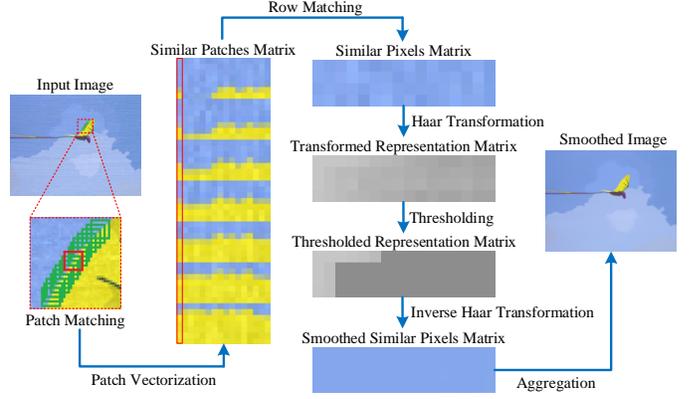


图 4. PNLs 平滑方法的流程图. 对于输入图像, 我们从中提取许多参考色块。对于每一个参考图像块 (红色的框), 我们在它周围搜索最相似的图像块 (绿色的框)。此过程称为图像块匹配, 其相似性通过欧几里得距离来衡量。然后, 我们将参考图像块及其相似 patches 展开为向量列, 并将它们放在一起以形成相似图像块矩阵。此步骤称为图像块矢量化。现在我们对相似图像块矩阵执行行匹配。对于这个矩阵, 每一行被称为参考行, 并且我们在相似图像块矩阵中搜索最相似的行来组成相似像素矩阵。通过 Haar 变换将相似像素矩阵的三个通道变换到表示域。我们执行阈值化以进行像素平滑, 并且通过逆 Haar 变换将每个通道的阈值表示矩阵转换回平滑的相似像素矩阵。最后, 我们将这些平滑的相似像素聚合回输入图像的正确位置。上面的步骤包括对参考图像块的平滑处理。为了执行一次图像平滑迭代, 我们对所有参考图像块执行上述过程。我们执行 PNLs 几次迭代以提高其性能。

SSIM 和 FSIM。与 PSNR 不同, SSIM 和 FSIM 反映了感知质量的相似性, 并将相似性与人类视觉神经生物学和感知联系在一起。由于这些原因, 我们使用这三个指标为平滑图像提供补充评估。我们还将 NKS 数据集中的 110 张大小为 512×384 的图像上报告不同方法的平均运行时间 (以秒为单位)。具体而言, 在 NVIDIA GTX 1080 GPU 上测试了 FIP [5], 在 [59] 中使用的两个 baselines ResNet 和 VDCNN, 而其他滤波器在 Intel Core i7-6700K CPU 上进行了测试。可以看到 EAW [15], GF [19] 和 FGS [33] 平均花费 0.02, 0.03 和 0.03 秒来处理 512×384 的图像, 这比另一个要快得多滤波器。

IV. 提出的像素级非局部平滑

在本节中, 我们介绍提出的像素级非局部平滑 (PNLS) 方法, 该方法包括三个步骤: 1) 搜索非局部相似像素 (§IV-A); 2) 估计平滑阈值 (§IV-B); 3) 通过基于 Haar 变换的阈值进行平滑处理 (§IV-C)。PNLS 的流程图如图所示 4. 请注意, 我们首先将 RGB 图像转换为亮度-色度空间 [8], 然后获得相应的 YCbCr 图像。我们搜索相似的像素并估计 Y 通道中的平滑阈值。Cb 和 Cr 通道的相似像素根据 Y 通道中的结果进行分

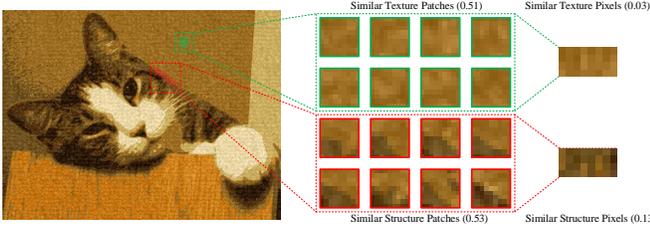


图 5. 像素级平滑的重要性。属于纹理 (0.51) 的相似贴片和属于结构 (0.53) 的相似贴片的标准偏差 (std) 非常接近。但是纹理上相似像素的标准差 (0.03) 比结构区域相似像素的标准差 (0.13) 小得多。因此, 像素级平滑可以很好地区分混合图像中的纹理和结构区域。

组。然后, 我们在每个通道上通过基于阈值的 Haar 变换执行图像平滑。最后, 我们将 YCbCr 空间中的平滑图像转换回 RGB 空间。

A. 搜索非局部相似像素

对于输入图像 $I \in \mathbb{R}^{h \times w}$, 我们将其转换为 YChCr 空间, 并且从它的 Y 通道中以步长为 s (横向和纵向) 提取大小为 $m \times m$ 的参考图像块。对于每个参考图像块, 我们首先在参考图像块周围的大小为 $R \times R$ 的窗口中搜索其相似图像块。正如 [20], [51] 所建议的, 相似性是通过欧几里得距离来度量的。然后我们重整每一个相似图像块 (包括参考图像块自己) 变成一个向量 $\mathbf{v}_r \in \mathbb{R}^{m^2}$ ($r = 1, \dots, R^2$, \mathbf{v}_1 是参考图像块)。我们通过选择与其最相似的 q 个图像块 (包括 \mathbf{v}_1 本身) 来为参考图像块 \mathbf{v}_1 执行图像块匹配。通过逐列堆叠 q 向量, 我们得到了类似的图像块矩阵 $\mathbf{P} = [\mathbf{v}_1, \dots, \mathbf{v}_q] \in \mathbb{R}^{m^2 \times q}$ 。但是, 仅图像块匹配无法很好地将纹理与结构区分开, 因为纹理通常处于提取图像块的范围内 [54]。我们在图 5 中更清楚地说明了这一点。具体来说, 我们分别从纹理区域 (绿色框) 和结构区域 (红色框) 中提取大小为 9×9 的相似纹理图像块。通过简单的计算, 我们可以看到纹理区域中相似图像块和结构区域中相似图像块的标准偏差 (std) 非常接近 (0.51 和 0.53)。为了更好地区分结构和纹理区域, 我们执行像素级行匹配以提取相似像素。也就是说, 对于相似图像块矩阵, 我们将每一行称为参考行, 并搜索相似图像块矩阵中最相似的行, 以形成相似像素矩阵。如图所示 4, 我们通过行匹配提取 4 个相似的像素组, 分别从相似的纹理和结构图像块中形成大小为 4×8 的相似像素矩阵。图 5 表明纹理区域中相似像素组的 std 为 0.03, 远小于结构区域相似像素组的 std 为 0.13。这表明像素级相似度比图像块级相似度更好地将纹理区域与结构区分开。我们将设置一

个平滑阈值, 以帮助准确地平滑图像, 这将在 §IV-B 中引入。对于相似像素组 (像素行) 的行匹配, 我们将相似图像块矩阵 \mathbf{P} 的第 i -th 行 $\mathbf{v}^i \in \mathbb{R}^q$ ($i = 1, \dots, m^2$) 作为参考行, 并且计算参考行 \mathbf{v}^i 和其它每一行 $\{\mathbf{v}^j \in \mathbb{R}^q, j = 1, \dots, m^2\}$ 之间的欧几里得距离, 如下所示:

$$d^{ij} = \|\mathbf{v}^i - \mathbf{v}^j\|_2. \quad (1)$$

然后我们选取距离参考行 \mathbf{v}^i 最近 (具有最小距离) 的 p 行像素, 组成相似像素矩阵 $\mathbf{S} = [\mathbf{v}^{i_1}, \dots, \mathbf{v}^{i_q}] \in \mathbb{R}^{p \times q}$ 。注意我们有 $\mathbf{v}^{i_1} = \mathbf{v}^i$ 和 $d^{i i_1} = 0$ 。对应于在 Y 通道中执行的图像块和像素匹配, 提取相似图像块矩阵 \mathbf{P} 的 Cb 和 Cr 通道中的相似像素矩阵。

B. 平滑阈值的估计

如图 5 所示, 与纹理区域中的像素相比, 图像中结构相似的图像块中的像素受到更大 stds 的影响。然后, 阈值对于确定是否平滑每个相似像素矩阵 (以及平滑程度) 以区分结构和纹理至关重要。由于 \mathbf{S} 的 p 行中的像素非常接近, 因此我们可以将 std 视为由于纹理变化而产生的能量估计, 并用如下式子计算。

$$\sigma = \frac{1}{m^2(p-1)\sqrt{q}} \sum_{t=2}^p \sum_{i=1}^{m^2} d^{i i_t}. \quad (2)$$

为了执行一致的图像平滑, 我们将全局阈值设置为所有相似像素矩阵的平均 σ 。

C. 通过基于 Haar 变换的阈值进行平滑

在 §IV-A 中, 我们获得了一组相似的像素矩阵 $\mathbf{S} \in \mathbb{R}^{p \times q}$ 和阈值 σ 。然后, 我们使用 Haar 变换 [18] 对相似像素矩阵进行阈值处理。具体来说, 我们采用带提升方案的 Haar 变换 [9], [38], 包括垂直变换矩阵 $\mathbf{H}_l \in \mathbb{R}^{p \times p}$ 和水平变换矩阵 $\mathbf{H}_r \in \mathbb{R}^{q \times q}$ 。为了执行 Haar 变换, 我们将 p, q 设置为 2 的幂。详细的转换过程在补充文件。这样, 我们得到了变换后的表示矩阵 $\mathbf{T} \in \mathbb{R}^{p \times q}$:

$$\mathbf{T} = \mathbf{H}_l \mathbf{S} \mathbf{H}_r. \quad (3)$$

通过使用平滑阈值, 我们能够恢复变换表示矩阵 \mathbf{T} 的第 i -th ($i = 1, \dots, p$) 行, 第 j -th ($j = 1, \dots, q$) 列的元素通过

$$\hat{\mathbf{T}} = \mathbf{T} \odot \mathbb{I}_{\{|\mathbf{T}| \geq \lambda \sigma^2\}}, \quad (4)$$

其中 \odot 表示按元素生产, \mathbb{I} 是指标函数, λ 是控制阈值范围的参数。

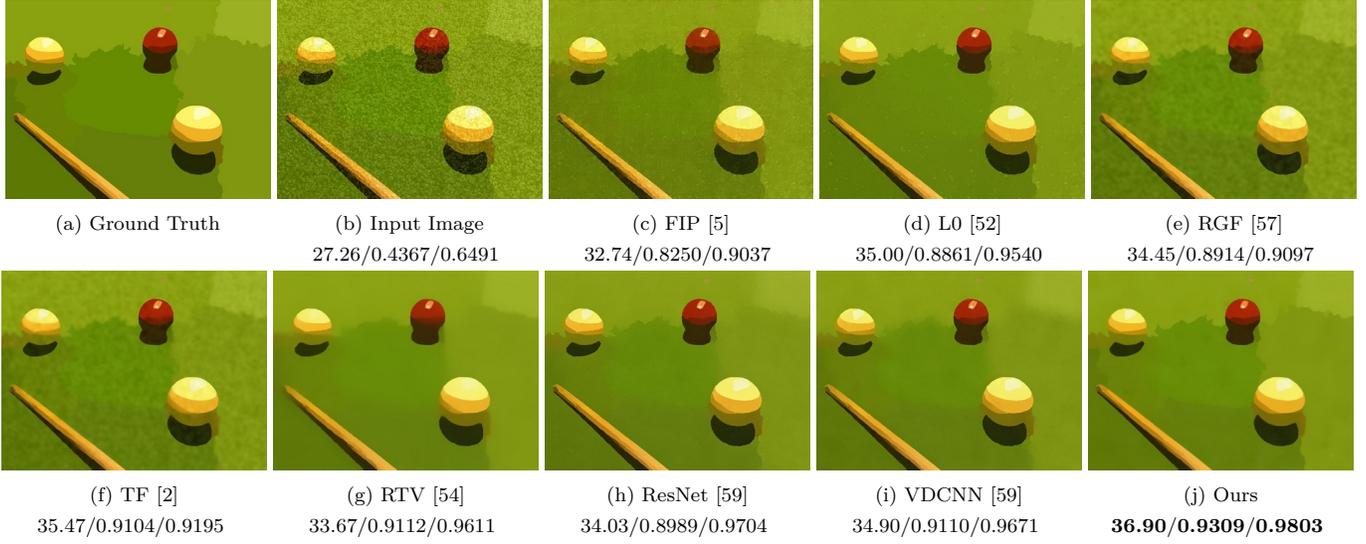


图 6. 在 NKS 数据集中的图像“S15_T01”上, 不同方法的平滑后图像和 PSNR(dB)/SSIM/FSIM 结果的比较。最好的结果被加粗了。

根据小波理论 [38], \mathbf{T} 的最后两行中的元素 (1-st 列除外) 属于 Haar 变换的高频带, 这些元素是纹理信息。我们直接在 $\hat{\mathbf{T}}$ 中将元素设置为零:

$$\tilde{\mathbf{T}}(i, j) = \hat{\mathbf{T}}(i, j) \odot \mathbb{I}_{\{\text{if } i=1, \dots, p-2 \text{ or } j=1\}}, \quad (5)$$

这里, $\tilde{\mathbf{T}}(i, j)$ 和 $\hat{\mathbf{T}}(i, j)$ 分别是矩阵 $\tilde{\mathbf{T}}$ 和矩阵 $\hat{\mathbf{T}}$ 的第 i -th ($i = 1, \dots, p$) 行, 第 j -th ($j = 1, \dots, q$) 列。然后, 我们在阈值表示矩阵上 $\tilde{\mathbf{T}}$ 使用垂直逆 Haar 变换矩阵 $\mathbf{H}_{il} \in \mathbb{R}^{p \times p}$ 和水平逆 Haar 变换矩阵 $\mathbf{H}_{ir} \in \mathbb{R}^{q \times q}$ 。在 S 补充文件中提供了详细的逆变换过程。然后我们可以得到没有纹理的平滑相似像素矩阵 $\tilde{\mathbf{S}}$, 通过

$$\tilde{\mathbf{S}} = \mathbf{H}_{il} \tilde{\mathbf{T}} \mathbf{H}_{ir}. \quad (6)$$

最后, 将平滑的相似像素矩阵聚合到原始图像中的相应位置。具体地, 将平滑的相似像素矩阵中的像素放置在原始图像的相应位置上。由于一个相似像素矩阵中的每个像素通常会与同一矩阵或其他矩阵中的其他像素重叠, 因此该像素的最终值应为该位置出现的像素的平均值。将原始图像中所有平滑的像素放置在具有最终值的正确位置后, 原始图像将进行迭代平滑。这是基于 Haar 变换技术的详细图像平滑处理 [18]。

D. 迭代平滑方案

为了获得更好的性能, 我们对 $N = 10$ 迭代应用上述平滑过程。实验表明, 我们的 $N = 10$ 的 PNLs 方法获得了令人满意的平滑效果 (请参考 §?? 了解更多细节)。

E. 复杂度分析

提出的 PNLs 方法包含三个部分: 1) 在 §IV-A 中, 图像块匹配的复杂度是 $\mathcal{O}(whR^2m^2/s^2)$, 然而行匹配的复杂度是 $\mathcal{O}(whqm^4/s^2)$; 2) 在 §IV-B 中, 平滑阈值估计的复杂度是 $\mathcal{O}(whm^2p/s^2)$; 3) 在 §IV-C 中, 基于 Haar 变换的阈值的复杂度是 $\mathcal{O}(whpqm^2/s^2)$ 。因为以上过程要迭代 N 次, 所以 PNLs 的复杂度是 $\mathcal{O}(whm^2N/s^2 \cdot \max\{R^2, qm^2, pq\})$ 。

V. 实验

在本节中, 我们首先在几种图像平滑基准数据集上比较提出的像素级非局部平滑 (PNLS) 方法和其它方法。我们还进行了全面的消融研究, 以深入了解提出的 PNLs 方法。附加文件中有更多视觉效果的对比较果。

A. 实现细节

参数设置. 如表 II 所示, 我们有两个版本的 PNLs: 快速 PNLs (No. 15) 和低速 PNLs (No. 16)。快速 PNLs 的参数包括: 搜索框的大小 $R = 15$, 提取相邻参考图像块的步长 $s = 4$, 图像块的大小 $m = 4$, 迭代次数 $N = 10$, 阈值是 $\lambda = 0.4$ 。低速 PNLs 的参数和高速 PNLs 的参数几乎差不多, 只是把步长修改为了 $s = 1$ (高速 PNLs 的步长为 $s = 4$)。对于两个版本的 PNLs 来说, 按照 [20] 的建议, 我们都设置了 $p = 4$ 和 $q = 16$ 。[20] 中的实验表明改变 p 或 q 的值只会影响 PNLs 在图像去噪方面的性能, 这与图像平滑类似。

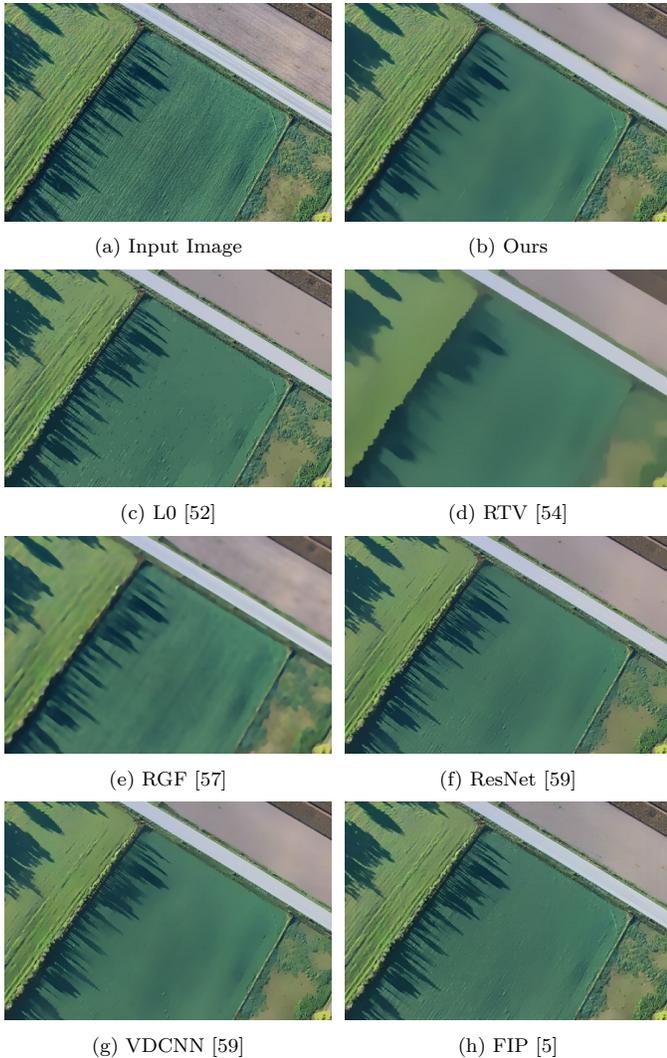


图 7. DIV2K [1] 数据集中的图像“0261”通过不同方法得到平滑图像的比较。

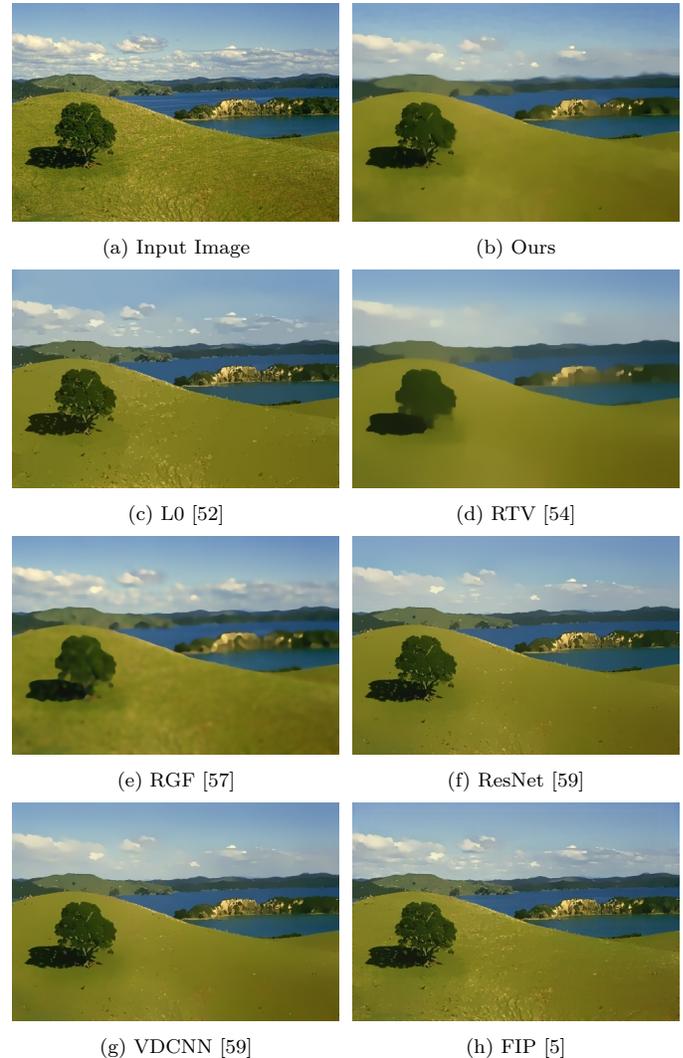


图 8. Comparison of smoothed images by different methods on the image “0251” from the EPS dataset [59].

比较方法. 我们将提出的 PNLs 方法与 14 种最新的图像平滑方法进行了比较: BF [31], WLS [14], EAW [15], GF [19], L0 [52], RTV [54], TF [2], FGS [33], RGF [57], fastABF [17], LRNN [27], FIP [5], 两个 baselines ResNet and VDCNN in [59]. 对于每种比较方法, 我们都从相应作者的网站下载其原始代码, 并使用其默认参数设置进行实验. 对于深层方法: LRNN [27], FIP [5], 和 ResNet/VDCNN in [59], 我们根据他们的论文重新训练了这些模型, 并根据它们在我们的 NKS 数据集上的性能对它们的超参数进行了微调. 对比是分别在 PSNR, SSIM [42], FSIM [56], 和视觉效果上进行评估的.

数据集. 我们在 DIV2K 数据集 [1] (1000 个具有不同内容的高分辨率 RGB 图像), RTV [54] 中使用的图像

以及边缘保留图像平滑 (EPS) 数据集 [59] (500 个图像) 上使用比较图像平滑方法, 以此评估 PNLs. 请注意, DIV2K [1] 和 RTV [54] 数据集中的图像没有相应的真实值图像, 而 EPS [59] 数据集的真实值是通过手动选择由七个现有的最新图像平滑算法生成的平滑图像来构造的.

评估标准. 在 [1], [54], [59] 3 个数据集上, 我们通过主观比较平滑图像的视觉质量来评估不同平滑方法的性能. 在我们的 NKS 数据集中, 以结构图像作为真实值. 因此, 我们从视觉质量上定性评估了各种平滑算法的性能, 同时在 PSNR, SSIM [42] and FSIM [56] 上定量的评估了它们的性能.

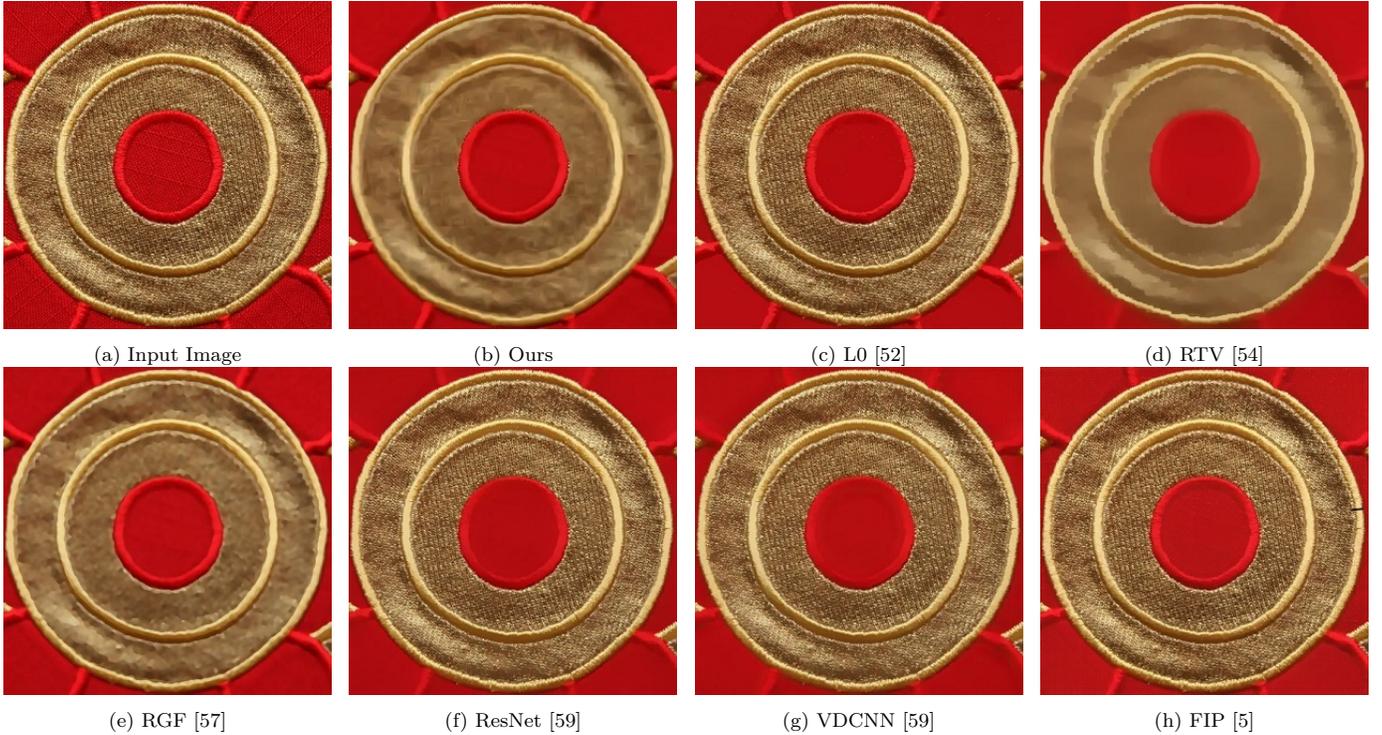


图 9. RTV [54] 数据集中的图像“11_11”通过不同方法得到平滑图像的比较。

B. 对比结果

如表II所示, 慢速 PNLs 在 NKS 数据集上获得 PSNR/SSIM/FSIM 的最佳结果。与慢速 PNLs 相比, 快速 PNLs 在客观指标上表现得很有竞争力。此外, 它大大改善了运行时间, 平均需要 5.10 秒来处理一张 512×384 图像。在图 6, 7, 8, 和 9, 我们将 PNLs 方法的视觉结果与其他最新的图像平滑方法进行了比较。我们观察到我们的 PNLs 方法在平滑图像纹理的同时保留了图像内容的结构。

在图 9 中, RTV [54] 可以很好地消除中间圆圈之间的纹理, 但会破坏红色背景图案的结构, 使其非常模糊。其他方法甚至没有完全去除中间圆之间的纹理。我们的 PNLs 在消除纹理的同时, 确保了背景图案的完整性和清晰度。请注意, 我们的 PNLs 表现好在于 [59] 中的基于深度学习的平滑网络, 即使这些网络用了很多带“真实值”的图片训练。

在图 6 中, 我们在图像“S15_T01”上比较了不同图像平滑方法的视觉质量 (主观度量) 和 PSNR/SSIM/FSIM (客观度量) 结果 (“15”是结构图像的索引, 而“01”是纹理索引 1)。

C. NKS 数据集上用户调研

尽管我们的 NKS 数据集具有真实值, 但是像以前的工作 [59] 一样, 通过高度受控的用户研究来主观地

评估比较方法更具说服力。为了与我们的算法一起进行用户研究, 我们选择了 7 种 sota 的方法: L0 [52], RTV [54], RGF [57], FIP [5], TF [2] 和两个 baselines ResNet and VDCNN in [59]。我们借助南开大学随机抽取的 80 名本科生和研究生进行用户研究, 并统一使用我们的 NKS 数据集进行评估。如图 10 所示, 在每一页中, 我们通过比较方法提供输入图像和平滑图像。用户需要做的只是选择他或她认为对图像平滑效果最好的平滑图像。图 11 是投票结果, 表明 PNLs 赢得了最多的票 (182 票), 比 ResNet [59] (175 票), 和 L0 [52] (153 票) 要好。

D. 消融研究

在这里, 我们对参数如何影响 PNLs 进行了更深入的研究。所有实验均在我们的 NKS 平滑数据集上进行。我们 PNLs 有 5 个主要参数, 搜索域大小 $R = 15$, 提取参考图像块步长 $s = 4$, 图像块的大小 $m = 4$, 迭代次数 $N = 10$, 阈值 $\lambda = 0.4$ 。对于每一个参数, 我们固定其它参数, 研究单独一个参数对 PNLs 的影响。PSNR, SSIM [42], 和 FSIM [56] 的结果总结在了表 III 中。我们观察到, 通过增加搜索域大小 R , 我们的 PNLs 的性能会提高, 因为一次可以匹配更多的图像块。在我们的 PNLs 中增加步长 s 将减少要处理的参考图像块的数量, 因此自然会降低我们 PNLs 的定量性能 (但

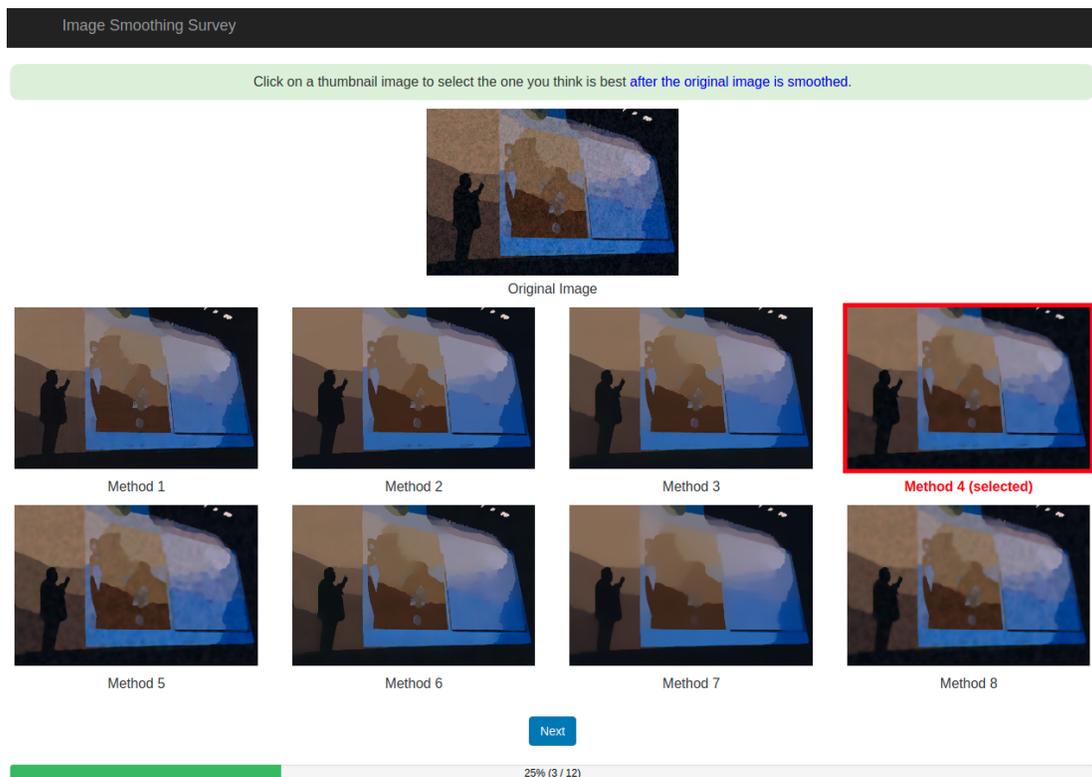


图 10. 用户研究实验中使用的系统接口。每个用户有 12 轮投票。在每一轮投票期间，用户选择通过他/她认为是图像平滑最佳结果的八种不同方法平滑的图像之一。用户只需要单击选定的图像，然后单击下一步按钮即可进行下一轮投票。

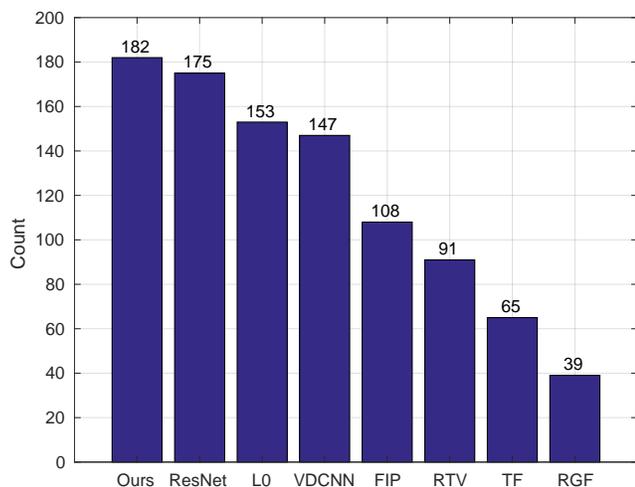


图 11. 80 位用户针对不同方法的投票直方图。

也会加快运行时间)。PNLS 的平滑强度随迭代次数 N 的增加而增加。我们的 PNLS 对于参数 m 或 λ 的更改非常鲁棒。另外，Table III 显示迭代次数 N 是对我们的 PNLS 性能最敏感的参数。为了研究不同的 N 对我们的 PNLS 的影响，在图 12 中，我们将 PNLS 与不同的 N 的视觉质量和 PSNR/SSIM/FSIM 结果进行比较，在图像 “S12_T06”。可以看到，我们的 PNLS 在

$N = 5, 10, 15, 20$ 的情况下始终表现良好，证明了我们的 PNLS 在不同迭代次数下的鲁棒性。

VI. 应用领域

我们将提出的像素级非局部平滑 (PNLS) 方法应用于 4 个图像处理任务：语义区域平滑，图像细节增强，图像边缘增强和图像抽象。请注意，我们在这里执行了四个图像处理应用程序，以显示 PNLS 在图像平滑方面的广泛实用性，而不是将 PNLS 与在相应任务上的特定方法进行比较。

语义区域平滑 主要用于仅平滑图像的前景或背景区域，而保留其他区域不变。在此任务中，我们首先预测将图像的前景和背景分开的前景掩膜，然后对我们感兴趣的区域进行平滑处理。如图 13 所示，对前景（上一行）或背景（下一行）进行了平滑处理，而背景区域则保持不变。在这里，我们使用显著的地面真理作为相应的前景掩膜。在实践中，我们采用著名的 [6] 方法来预测显著区域。

图像细节增强 旨在增强图像的细节，同时避免产生伪影（渐变反转或光晕）。对于输入图像，我们的 PNLS 将其

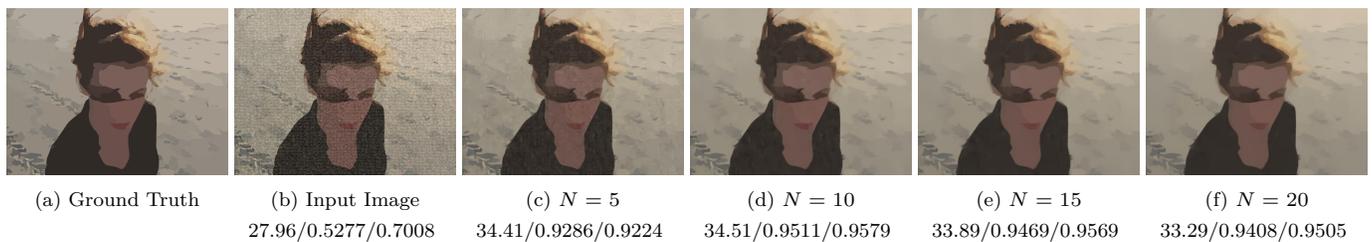


图 12. 在我们的 NKS 数据集集中的图像 “S12_T06” 上，我们的 PNLs 使用参数 N 以不同的值比较平滑图像和 PSNR(dB)/SSIM/FSIM 结果。

表 III

NKS 数据集上具有不同参数的 PNLs 的平均 PSNR (dB), SSIM 和 FSIM。我们一次更改一个参数，以评估其对 PNLs 的个别影响。“ \uparrow ” 表示越高越好。

R	Value	11	13	15	17	Margin
	PSNR \uparrow	33.30	33.39	33.45	33.48	0.18
	SSIM \uparrow	0.9358	0.9371	0.9378	0.9379	0.0021
	FSIM \uparrow	0.9378	0.9392	0.9397	0.9396	0.0019
s	Value	1	2	3	4	Margin
	PSNR \uparrow	33.68	33.65	33.59	33.45	0.23
	SSIM \uparrow	0.9420	0.9415	0.9403	0.9378	0.0042
	FSIM \uparrow	0.9440	0.9436	0.9424	0.9397	0.0043
m	Value	4	5	6	7	Margin
	PSNR \uparrow	33.45	33.41	33.32	33.27	0.18
	SSIM \uparrow	0.9378	0.9391	0.9388	0.9382	0.0013
	FSIM \uparrow	0.9397	0.9416	0.9413	0.9406	0.0019
N	Value	5	10	15	20	Margin
	PSNR \uparrow	33.31	33.45	33.08	32.68	0.77
	SSIM \uparrow	0.9206	0.9378	0.9377	0.9355	0.0172
	FSIM \uparrow	0.9166	0.9397	0.9409	0.9387	0.0243
λ	Value	0.3	0.4	0.5	0.6	Margin
	PSNR \uparrow	33.49	33.45	33.31	33.13	0.36
	SSIM \uparrow	0.9361	0.9378	0.9378	0.9370	0.0017
	FSIM \uparrow	0.9365	0.9397	0.9402	0.9396	0.0037

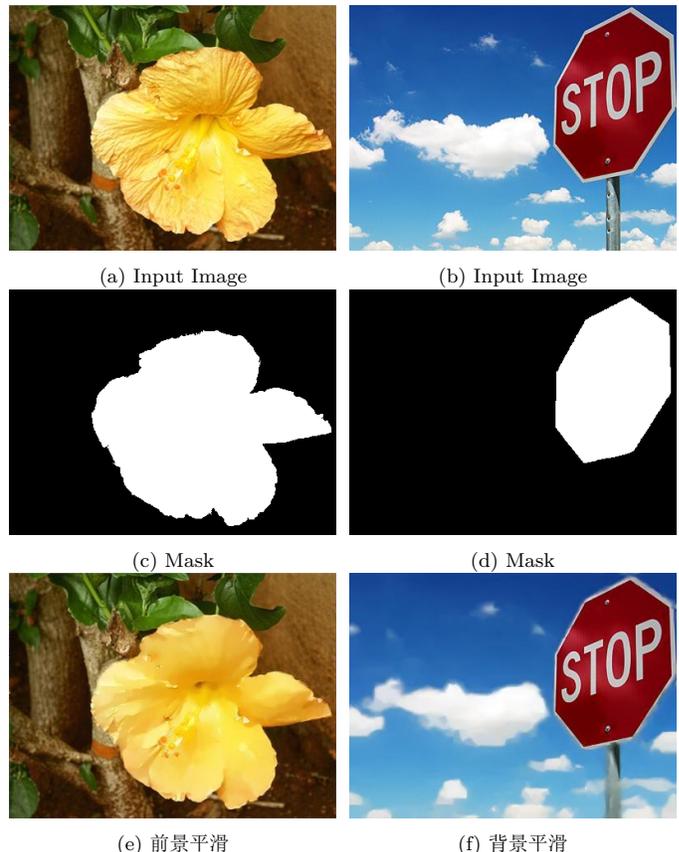


图 13. 语义区域平滑 PNLs 方法对来自 MSRA-B SOD 数据集 [41] 的两幅图像。

分解为基础层（平滑的图像）和细节层（移除的纹理）。然后我们将详细信息层放大 3 倍，同时保持基础层不变。通过将放大的细节层添加回基础层来获得增强的图像。在图 14 中，我们展示输入图片，PNLs 平滑后的图片，和 [59] 增强后的图片关于 “0347” 和 “0484”。我们观察到，与输入图像相比，具有增强细节的图像看起来非常自然。这表明我们的 PNLs 在删除细节的同时很好地保留了输入图像的结构。

图像边缘增强. 如图 15 (b) 所示，与输入图像 (图 15 (a)) 相比，马和草坪的边界仍然非常清晰。在这里，我们使用拉普拉斯算子和 Canny 边缘检测器 [4] 分别计算图 15 (a) 和 (b) 的渐变图和边缘图。由于周围区域

中纹理的干扰，很难区分输入图像 (a) 的渐变图 (c) 中的边界。由 Canny 边缘检测器 [4] 提取的边缘图 (e) 也受到 (a) 中纹理的严重影响。如图 15 (d) 和 (f) 所示，我们的 PNLs 方法可以平滑不重要的纹理（请参考图 15 (b) 中马和草坪的纹理）。因此，拉普拉斯算子和 Canny 边缘检测器 [4] 能够分别提取平滑图像 (Figure 15 (b)) 的清晰梯度图 (Figure 15 (d)) 和可靠边缘图 (Figure 15 (f))。

图像抽象. 我们的 PNLs 方法也可以应用于图像抽象任务。视觉结果显示在图 16 中。正如 [43] 所建议的，我们通过用 PNLs 替换双边滤波 [31] 来进行图像抽象，并分别提取输入图像 (a) 和 (b) 的卡通风格抽象 (c) 和

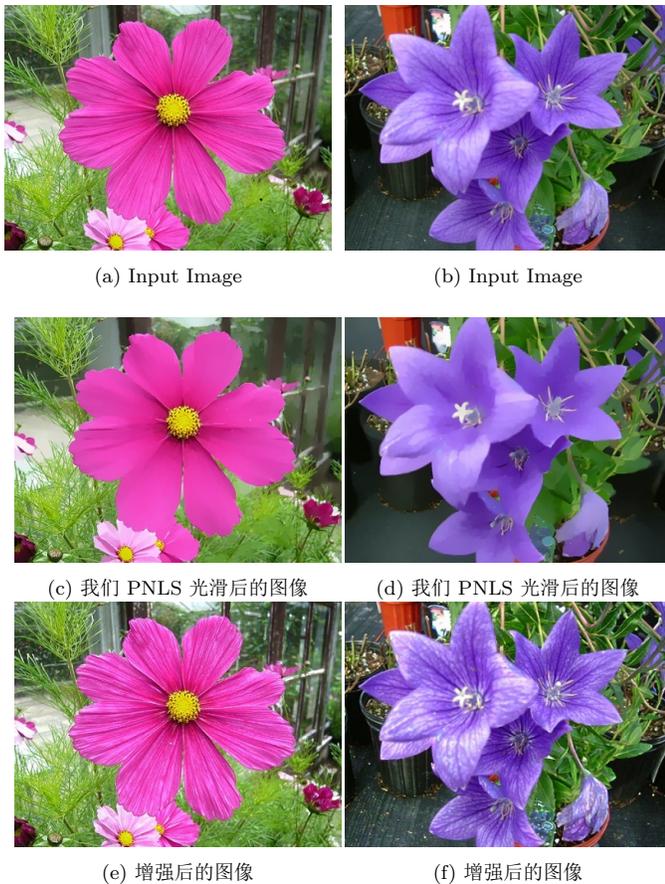


图 14. 图像细节增强图 PNLS 平滑方法实现。

(d)。此外，我们还使用 [29] 方法分别生成了抽象图像 (c) 和 (d) 的铅笔素描结果 (e) 和 (f)。可以看到，由于我们的 PNLS 能够捕获准确的图像结构，因此可以帮助获得有希望的图像抽象和铅笔素描结果。

VII. 结论

尽管算法取得了进步，但在图像平滑社区中仍然需要适当的基准数据集。在本文中，我们构建了一个 Nankai Smoothing (NKS) 数据集来服务于图像平滑算法的比较。通过我们的 NKS 数据集，我们对 14 种流行的图像平滑算法进行了基准测试。此外，我们还利用自然图像先验的像素级非局部自相似性，提出了一种像素级非局部平滑法 (PNLS)。在几个基准数据集 (包括我们的 NKS 数据集) 上，我们的 PNLS 在质量和 (或) 定量性能上均优于其他竞争方法。广泛的参数分析验证了我们的 PNLS 在图像平滑上的鲁棒性。我们进一步验证了所提出的 PNLS 方法在语义区域平滑，细节/边缘增强和图像抽象等任务上的广泛实用性。

这项工作可以至少在两个方向上扩展。首先，我们可以进一步加快 PNLS 方法的速度，以满足实际应用。

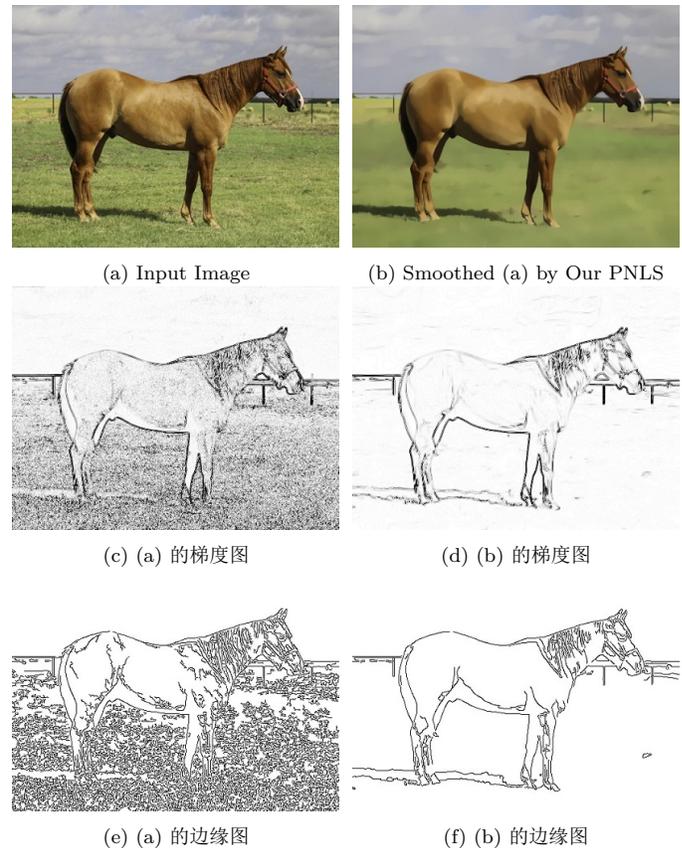


图 15. 边缘增强和提取结果的图示。我们的 PNLS 平滑方法可抑制纹理细节，并增强输入图像的结构边缘 (a)。

其次，我们可以构建一个更大的图像平滑数据集，其中包含更合理，更现实的真实值，包括训练图像和测试图像，以更好地基准基于深度学习的平滑网络。

参考文献

- [1] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, July 2017. 3, 8
- [2] L. Bao, Y. Song, Q. Yang, H. Yuan, and G. Wang. Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree. *IEEE Trans. Image Process.*, 23(2):555–569, 2013. 4, 5, 7, 8, 9
- [3] V. Bychkovsky, S. Paris, E. Chan, and F. Durand. Learning photographic global tonal adjustment with a database of input/output image pairs. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011. 3
- [4] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986. 11
- [5] Q. Chen, J. Xu, and V. Koltun. Fast image processing with fully-convolutional networks. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 2497–2506, 2017. 1, 2, 4, 5, 7, 8, 9
- [6] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):569–582, 2015. 10
- [7] Clker-Free-Vector-Images. Pixabay. <https://pixabay.com/users/clker-free-vector-images-3736/>. Accessed: 2019-11-02. 3

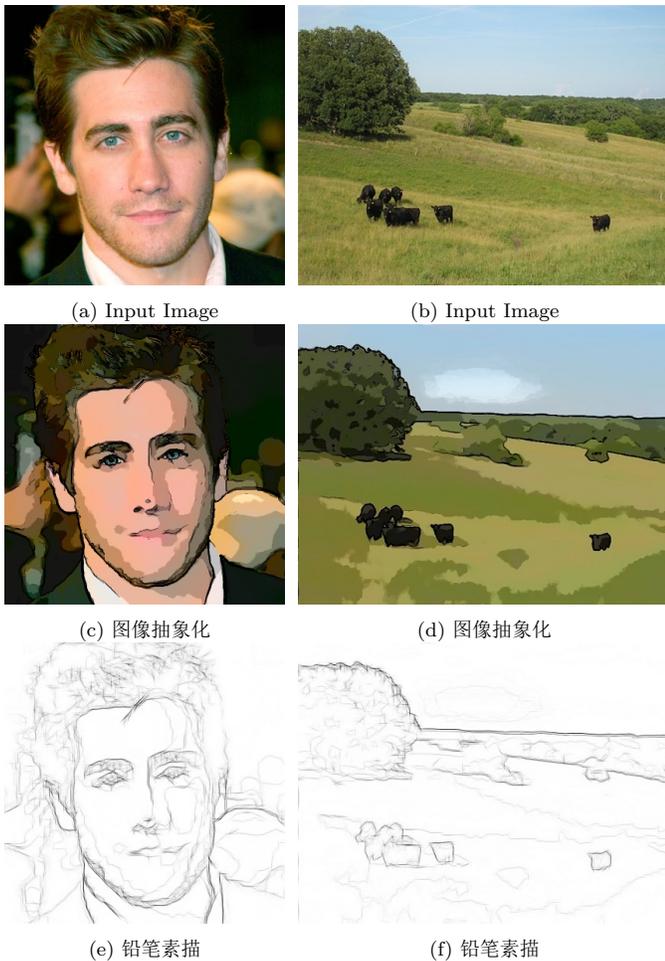
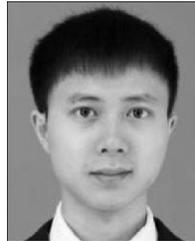


图 16. 图像抽象和铅笔素描结果插图, 其中我们的 PNLs 方法删除了纹理细节。

- [8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16(8):2080–2095, 2007. 5
- [9] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4(3):247–269, 1998. 6
- [10] Y. Du, J. Xu, Q. Qiu, X. Zhen, and L. Zhang. Variational image deraining. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 2406–2415, 2020. 1
- [11] Y. Du, J. Xu, X. Zhen, M.-M. Cheng, and L. Shao. Conditional variational image deraining. *IEEE Transactions on Image Processing*, 29:6288–6301, 2020. 1, 3
- [12] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 3238–3247, 2017. 2
- [13] Q. Fan, J. Yang, D. Wipf, B. Chen, and X. Tong. Image smoothing via unsupervised learning. *ACM Trans. Graph.*, 37(6), 2018. 2
- [14] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans. Graph.*, 27(3):67, 2008. 1, 2, 4, 5, 8
- [15] R. Fattal. Edge-avoiding wavelets and their applications. *ACM Trans. Graph.*, 28(3):22, 2009. 4, 5, 8
- [16] R. Fattal, M. Agrawala, and S. Rusinkiewicz. Multiscale shape and detail enhancement from multi-light image collections. *ACM Trans. Graph.*, 26(3), July 2007. 2
- [17] R. G. Gavaskar and K. N. Chaudhury. Fast adaptive bilateral filtering. *IEEE Trans. Image Process.*, 28(2):779–790, 2018. 4, 5, 8
- [18] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, Sep 1910. 6, 7
- [19] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(6):1397–1409, 2013. 1, 2, 4, 5, 8
- [20] Y. Hou, J. Xu, M. Liu, G. Liu, L. Liu, F. Zhu, and L. Shao. Nlh: A blind pixel-level non-local method for real-world image denoising. *IEEE Trans. Image Process.*, 29:5121–5135, 2020. 2, 6, 7
- [21] Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep joint image filtering. In *European Conference on Computer Vision.*, pages 154–169, 2016. 1, 2
- [22] Z. Liang, J. Xu, D. Zhang, Z. Cao, and L. Zhang. A hybrid l1-l0 layer decomposition model for tone mapping. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, June 2018. 1
- [23] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. Worksh.*, pages 136–144, 2017. 3
- [24] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang. Noise estimation from a single image. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, volume 1, pages 901–908, 2006. 2
- [25] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang. Non-local recurrent network for image restoration. In *Proc. Adv. Neural Inform. Process. Syst.*, pages 1673–1682, 2018. 3
- [26] D. Liu, B. Wen, J. Jiao, X. Liu, Z. Wang, and T. S. Huang. Connecting image denoising and high-level vision tasks via deep learning. *IEEE Trans. Image Process.*, 29:3695–3706, 2020. 3
- [27] S. Liu, J. Pan, and M.-H. Yang. Learning recursive filters for low-level vision via a hybrid neural network. In *European Conference on Computer Vision.*, pages 560–576, 2016. 1, 2, 4, 5, 8
- [28] W. Liu, X. Chen, C. Shen, Z. Liu, and J. Yang. Semi-global weighted least squares in image filtering. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 5861–5869, 2017. 2
- [29] C. Lu, L. Xu, and J. Jia. Combining sketch and tone for pencil drawing production. In *Proceedings of the Symposium on Non-Photorealistic Animation and Rendering*, pages 65–73, 2012. 12
- [30] K. Lu, S. You, and N. Barnes. Deep texture and structure aware filtering network for image smoothing. In *European Conference on Computer Vision.*, September 2018. 1, 2, 3
- [31] R. Manduchi and C. Tomasi. Bilateral filtering for gray and color images. In *Proc. IEEE Int. Conf. Comput. Vis.*, 1998. 1, 2, 4, 5, 8, 11
- [32] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. IEEE Int. Conf. Comput. Vis.*, volume 2, pages 416–423, July 2001. 3
- [33] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do. Fast global image smoothing based on weighted least squares. *IEEE Trans. Image Process.*, 23(12):5638–5653, 2014. 1, 4, 5, 8
- [34] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.*, 23(3):664–672, 2004. 1, 2

- [35] D. Ren, W. Zuo, D. Zhang, J. Xu, and L. Zhang. Partial deconvolution with inaccurate blur kernel. *IEEE Trans. Image Process.*, 27(1):511–524, 2018. 3
- [36] X. Shen, Y.-C. Chen, X. Tao, and J. Jia. Convolutional neural pyramid for image processing. *arXiv preprint arXiv:1704.02071*, 2017. 2
- [37] Z. Su, X. Luo, Z. Deng, Y. Liang, and Z. Ji. Edge-preserving texture suppression filter based on joint filtering schemes. *IEEE Trans. Multimedia*, 15(3):535–548, 2013. 1, 2
- [38] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3(2):186 – 200, 1996. 6, 7
- [39] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin. Deep learning on image denoising: An overview. *arXiv preprint arXiv:1912.13171*, 2019. 3
- [40] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu. Attention-guided cnn for image denoising. *Neural Networks*, 2020. 3
- [41] J. Wang, H. Jiang, Z. Yuan, M.-M. Cheng, X. Hu, and N. Zheng. Salient object detection: A discriminative regional feature integration approach. *Int. J. Comput. Vis.*, 123(2):251–268, 2017. 11
- [42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. 3, 4, 5, 8, 9
- [43] H. Winnemöller, S. C. Olsen, and B. Gooch. Real-time video abstraction. *ACM Trans. Graph.*, 25(3):1221–1226, July 2006. 2, 11
- [44] J. Xu. *Nonlocal self-similarity based prior modeling for image denoising*. PhD thesis, The Hong Kong Polytechnic University, 2018. 3
- [45] J. Xu, Y. Hou, D. Ren, L. Liu, F. Zhu, M. Yu, H. Wang, and L. Shao. Star: A structure and texture aware retinex model. *IEEE Trans. Image Process.*, 2020. 1
- [46] J. Xu, Y. Huang, M.-M. Cheng, L. Liu, F. Zhu, Z. Xu, and L. Shao. Noisy-as-clean: Learning unsupervised denoising from the corrupted image. *IEEE Trans. Image Process.*, 29:9316–9329, 2020. 3
- [47] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*, 2018. 3
- [48] J. Xu, L. Zhang, and D. Zhang. External prior guided internal prior learning for real-world noisy image denoising. *IEEE Trans. Image Process.*, 27(6):2996–3010, June 2018. 3
- [49] J. Xu, L. Zhang, and D. Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *European Conference on Computer Vision*, 2018. 3
- [50] J. Xu, L. Zhang, D. Zhang, and X. Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *Proc. IEEE Int. Conf. Comput. Vis.*, 2017. 3
- [51] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng. Patch group based nonlocal self-similarity prior learning for image denoising. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 244–252, 2015. 3, 6
- [52] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via l_0 gradient minimization. *ACM Trans. Graph.*, 30(6):174, 2011. 1, 2, 4, 5, 7, 8, 9
- [53] L. Xu, J. Ren, Q. Yan, R. Liao, and J. Jia. Deep edge-aware filters. In *International Conference on Machine Learning*, pages 1669–1678, 2015. 1, 2
- [54] L. Xu, Q. Yan, Y. Xia, and J. Jia. Structure extraction from texture via relative total variation. *ACM Trans. Graph.*, 31(6):139:1–139:10, 2012. 1, 2, 3, 4, 5, 6, 7, 8, 9
- [55] F. Zhang, L. Dai, S. Xiang, and X. Zhang. Segment graph based image filtering: Fast structure-preserving smoothing. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 361–369, 2015. 1
- [56] L. Zhang, L. Zhang, X. Mou, and D. Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE Trans. Image Process.*, 20(8):2378–2386, 2011. 3, 4, 5, 8, 9
- [57] Q. Zhang, X. Shen, L. Xu, and J. Jia. Rolling guidance filter. In *European Conference on Computer Vision*, pages 815–830, 2014. 2, 4, 5, 7, 8, 9
- [58] Z. Zhou, B. Wang, and J. Ma. Scale-aware edge-preserving image filtering via iterative global optimization. *IEEE Trans. Multimedia*, 20(6):1392–1405, 2018. 1, 2
- [59] F. Zhu, Z. Liang, X. Jia, L. Zhang, and Y. Yu. A benchmark for edge-preserving image smoothing. *IEEE Trans. Image Process.*, 28(7):3556–3570, 2019. 1, 3, 4, 5, 7, 8, 9, 11



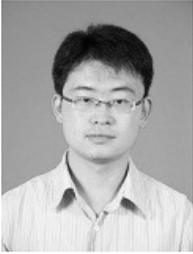
Jun Xu received the B.Sc. degree in pure mathematics and the M.Sc. degree from the School of Mathematics Science, Nankai University, in 2011 and 2014, respectively, and the Ph.D. degree from the Hong Kong Polytechnic University, in 2018. He worked as a Research Scientist at the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates. He is currently an Assistant Professor with the College of Computer Science, Nankai University. More information can be found on his homepage <https://csjunxu.github.io/>.



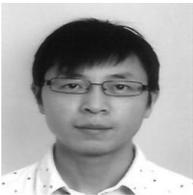
Zhi-Ang Liu received his B.S. degree from the School of Electrical Engineering and Automation, Harbin Institute of Technology in 2019. Currently he is a master student in the College of Computer Science, Nankai University, supervised by Prof. Ming-Ming Cheng. His research interests include machine learning and computer vision.



Ying-Kun Hou received his Ph. D degree from the School of Computer Science and Technology, Nanjing University of Science and Technology in 2012. He is currently an associate professor with the School of Information Science and Technology, Taishan University, Taian, China. His current research interests are in the areas of image processing, pattern recognition, and artificial intelligence.



Xian-Tong Zhen received the B.S. and M.E. degrees from Lanzhou University, Lanzhou, China in 2007 and 2010 and the Ph.D. degree from the Department of Electronic and Electrical Engineering, the University of Sheffield, UK in 2013. He worked as postdoctoral fellows with the University of Western Ontario, London, Canada and the University of Texas at Arlington, Texas, U.S.A. from 2013 to 2017. He was an associate professor with the School of Electronics and Information Engineering, Beihang University, Beijing, China. He is now an assistant professor in AIM Lab, University of Amsterdam, the Netherlands. His research interests include machine learning and computer vision.



Ling Shao is the Executive Vice President and Provost of the Mohamed bin Zayed University of Artificial Intelligence. He is also the CEO and Chief Scientist of the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates. His research interests include computer vision, machine learning, and medical imaging. He is a Fellow of IAPR, IET, and BCS.



Ming-Ming Cheng received his Ph.D. degree from Tsinghua University in 2012. Then he did two years research fellow with Prof. Philip Torr in Oxford. He is now a professor at Nankai University, leading the Media Computing Lab. His research interests include computer graphics, computer vision, and image processing. He received research awards, including ACM China Rising Star Award, IBM Global SUR Award, and CCF-Intel Young Faculty Researcher Program. He is on the editorial boards of IEEE TIP.