Error-Tolerant Image Compositing

Michael W. Tao · Micah K. Johnson · Sylvain Paris

Abstract Gradient-domain compositing is an essential tool in computer vision and its applications, e.g., seamless cloning, panorama stitching, shadow removal, scene completion and reshuffling. While easy to implement, these gradient-domain techniques often generate bleeding artifacts where the composited image regions do not match. One option is to modify the region boundary to minimize such mismatches. However, this option may not always be sufficient or applicable, e.g., the user or algorithm may not allow the selection to be altered. We propose a new approach to gradient-domain compositing that is robust to inaccuracies and prevents color bleeding without changing the boundary location. Our approach improves standard gradient-domain compositing in two ways. First, we define the boundary gradients such that the produced gradient field is nearly integrable. Second, we control the integration process to concentrate residuals where they are less conspicuous. We show that our approach can be formulated as a standard least-squares problem that can be solved with a sparse linear system akin to the classical Poisson equation. We demonstrate results on a variety of scenes. The visual quality and runtime complexity compares favorably to other approaches.

Keywords Gradient-domain compositing · Visual masking

1 Introduction

Gradient-domain compositing is an essential technique at the core of many computer vision applications such as seamless cloning (Prez et al. 2003; Agarwala et al. 2004; Georgiev 2006; Jia et al. 2006), panorama stitching (Levin et al. 2006; Agarwala 2007; Sivic et al. 2008), inpainting (Whyte et al. 2009), shadow removal (Finlayson et al. 2009), scene completion (Hays and Efros 2007), and reshuffling (Cho et al. 2010). These methods first delineate the composited regions, then compute a target gradient field and boundary conditions from these regions, and finally solve the Poisson equation to reconstruct an image. A major issue with gradient-domain compositing is that the combined gradient field may not be integrable; that is, an image with gradients that match the target field as well as the specified boundary conditions may not exist. Existing work mitigates this issue by moving the boundary to more carefully combine the merged regions. However, when the combined images are widely different, this strategy may not be sufficient. Or, if the user has specified the boundary by hand, he or she may not want it to be altered. For instance in Fig. 1, the selection cannot be modified because the tree trunks have to abut the pyramids. Even with boundary refinement, the target gradient fields may be far from integrable, yielding color leaks and halos typical of Poisson-based methods.

In this paper, we present an algorithm for minimizing artifacts in gradient-domain image compositing. We characterize the origin of typical bleeding artifacts and analyze
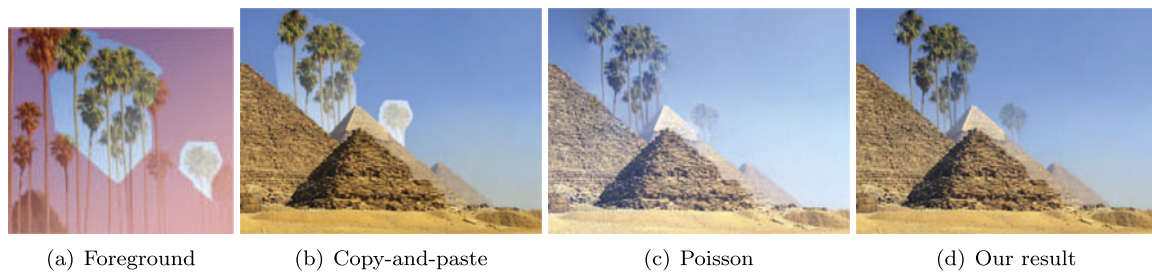
M.W. Tao (✉)
University of California, Berkeley, CA, USA
e-mail: mtao@berkeley.edu

M.K. Johnson
Massachusetts Institute of Technology, Cambridge, MA, USA
e-mail: kimo@csail.mit.edu

S. Paris
Adobe Systems, Inc., Cambridge, MA, USA
e-mail: sparis@adobe.com

(a) Foreground   (b) Copy-and-paste   (c) Poisson   (d) Our result

Fig. 1 We present an image-compositing technique tolerant to selection inaccuracies. In this example, a user wishes to add trees to an image of the Egyptian pyramids, but it is not possible to select the trees without cutting through the foliage (**a**). Moreover, to ensure a good insertion behind the pyramids, it is not possible to modify the selection boundary. A direct copy of the pixels yields a undesirable visible seam (**b**). Standard gradient-domain compositing minimizes the seam, but leads to bleeding artifacts where the foliage is cut (**c**). Our method characterizes where color leakage should be avoided, producing a seamless composite without bleeding artifacts (**d**)

the image to locate the areas where they would be most and least conspicuous. Based on this analysis, we propose a two-step algorithm. First, we process the gradient values on the boundary to minimize artifacts in regions where bleeding would be visible. Second, we describe a weighted integration scheme that reconstructs the image from its gradient field so that residuals are located in textured regions where they are less visible. Our results show that the combination of these two steps yields significantly better composites. Moreover, our method is formulated as a least-squares optimization that can be solved using a sparse linear system, which makes our approach computationally efficient. We demonstrate our approach on scenarios in which boundary mismatches are likely to occur: user-driven seamless cloning (Prez et al. 2003), heterogeneous panorama stitching (Sivic et al. 2008), and scene reshuffling (Cho et al. 2010).

1.1 Related Work

Gradient-domain techniques are useful to a variety of problems in computer vision, including image stitching, intrinsic images, shadow removal, and shape-from-shading (Levin et al. 2006; Tappen et al. 2005; Finlayson et al. 2006; Agrawal et al. 2006; Bhat et al. 2009). In most of these problems, the gradient field contains non-integrable regions and many authors have noted that reconstruction artifacts are often due to boundary conditions. As a result, a variety of methods have been introduced to minimize artifacts by refining the boundary location (Agarwala et al. 2004; Jia et al. 2006; Levin et al. 2006; Lalonde et al. 2007). Rather than moving the boundary, which may not always be possible, we focus on reconstructing the final image from the target gradient field once the boundary is specified. Our approach is complementary and orthogonal to boundary-refinement methods. We show that our image analysis combined with a careful study of the numerical scheme reduces visible artifacts. Our approach could benefit many computer vision algorithms that rely on gradient-domain reconstruction as a subroutine.

The general formulation of the gradient-domain reconstruction problem is to seek an image I that approximates the target field $\mathbf{v}$ in a least-squares sense (with ∇, the gradient operator):

$$\operatorname*{argmin}_{I} \int \|\nabla I - \mathbf{v}\|^2 \tag{1}$$

which can be minimized by solving the Poisson equation:

$$\Delta I - \operatorname{div}(\mathbf{v}) = 0 \tag{2}$$

where Δ is the Laplacian operator $\partial^2/\partial x^2 + \partial^2/\partial y^2$ and div is the divergence operator $\partial/\partial x + \partial/\partial y$. To solve this equation, one also needs boundary conditions that depend on the application. After solving for ΔI, I and $I + k$, where k is a constant, can satify the equation. Usually one or more data terms are used to bind the k to a certain value. We illustrate how to compute the target gradient $\mathbf{v}$ in the context of seamless compositing using three inputs: the background image, B; the foreground image, F; and a selection, $\mathcal{S}$ with a boundary β (Levin et al. 2006).

$$\mathbf{v}(x, y) = \begin{cases} \nabla F & \text{if } (x, y) \in \mathcal{S}, (x, y) \notin \beta \\ \nabla B & \text{if } (x, y) \notin \mathcal{S} \\ \frac{1}{2}(\nabla F + \nabla B) & \text{if } (x, y) \in \beta \end{cases} \tag{3}$$

Other cases such as panorama stitching are similar except that the images are not named "foreground" and "background." For the sake of simplicity, we will name the images foreground and background.

The gradients from the foreground image F and background image B are integrable since they are computed directly from images. But the gradients along the boundary between the two images may not be integrable, creating a source of errors that the integration routine must manage. Farbman et al. (2009) address this issue by relying on users to identify the leaks. The gradients of marked regions are ignored, which removes the leaks. In comparison, our method

analyzes the image to automatically adapt the integration process. Our approach shares similarities with the method of Lalonde et al. (2007) who propose to take the image gradient magnitude into account during the reconstruction process. However, color leaks may still appear with this technique when boundaries are not accurate.

Besides image compositing, gradient-domain methods have also been used in computer vision for surface reconstruction problems, such as shape-from-shading and photometric stereo. In these problems, an algorithm estimates the gradient of a surface at every pixel and then a robust Poisson solver is used to find the surface that best fits the estimated gradients. We refer to the recent work of Agrawal et al. (2006), Reddy et al. (2009), and the references therein for detail. Although image compositing and robust integration techniques both reconstruct a 2D signal from its gradients, the two problems are fundamentally different. The gradients from surface-reconstruction methods are noisy everywhere, whereas image-compositing gradients are problematic only at the boundary between foreground and background. In this paper, we exploit this specificity to improve the quality of the results. We also rely on visual masking to locate integration residuals where they are less conspicuous.

1.2 Contributions

In this paper, we introduce several contributions:

– *Low-curl boundaries*. We describe a method that limits the artifacts by minimizing the curl of the target gradients on the foreground-background boundary.
– *Weighted Poisson equation*. We show how to add weights to the Poisson equation so that integration residuals lie in textured regions where they are less visible due to visual masking.
– *Efficient non-bleeding compositing*. We combine the two previous contributions to obtain a compositing algorithm that prevents bleeding artifacts while remaining linear akin to the original Poisson equation.

1.3 Overview

Our algorithm consists of two steps. First, we focus on the boundary between the foreground and background regions. We characterize the origin of the bleeding artifacts and we show how to modify the gradient field **v** to minimize them. The second step focuses on the location of the integration residuals. We show that artifacts are less visible in textured regions due to visual masking. We describe an algorithm that controls the integration residuals such that they are located in textured areas. In the results section, we show that the combination of these two steps yields visually superior results.

(a) Within foreground   (b) On boundary
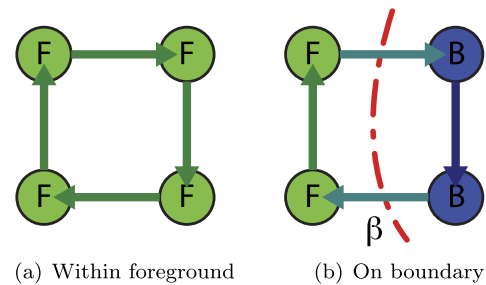
Fig. 2 *Estimating the curl on a discrete grid. Circles* denote pixels and *arrows* denote differences between pixels. If the curl is computed within the foreground region (**a**), all the derivatives come from *F* and the curl is null. The background case is equivalent (not shown). On the boundary (**b**), derivatives from diverse sources are used and in general the curl is not zero

2 Low-Curl Boundary

A necessary condition for a gradient field **u** to be integrable is to have a zero curl.[1] That is, if there exists an image I such that $\nabla I = \mathbf{u}$, then $\text{curl}(\mathbf{u}) = \partial \mathbf{u}_y / \partial x - \partial \mathbf{u}_x / \partial y = 0$. For example, consider the configuration illustrated in Fig. 2(a). When all pixels come from one image, in this case the foreground image, the derivatives are consistent and the curl is zero. Therefore, this region is integrable, i.e., the image can be reconstructed from its gradients. The same observation holds for regions from the background image.

In the image compositing problem, a non-integrable gradient field only occurs on the boundary, as illustrated in Fig. 2(b). On the boundary, the gradient field is a mixture of two fields and may have non-zero curl since gradients come from mixed sources. When the gradient field has a non-zero curl, we cannot minimize the Poisson equation (2) exactly and residuals remain. These residuals are often visible in composited images as halos, bleeding, or other artifacts.

2.1 Reducing the Curl on the Boundary

Since the non-integrability of regions along boundary is the source of artifacts, we seek to alter the desired gradient field to minimize the bleeding artifacts. Let **v** represent the desired gradient field of the composited image. To preserve the visual information of the two images, we do not modify the foreground or background gradients in **v**. We only modify **v** values on the boundary such that the curl is as small as possible.

A naive solution would be to seek $\text{curl}(\mathbf{v}) = 0$ everywhere. But the following counterexample shows that this approach would not achieve our goal. Consider the standard copy-and-paste operation that directly combines the pixel

---

[1]Note that for a 2D vector field $\mathbf{u} = (\mathbf{u}_x, \mathbf{u}_y)$, the curl is a scalar value that corresponds to the z component of the 3D curl applied to the 3D vector field $(\mathbf{u}_x, \mathbf{u}_y, 0)$.

Fig. 3 *Influence of the curl term*. With high weights W_β, the composite is seamless but suffers from bleeding (**a**). With low W_β, bleeding disappears but seams become visible (**b**). Our adaptive approach locally adjusts the weights to achieve seamless results with no leaks (**c**)

(a) High weight   (b) Low weight   (c) Adaptive
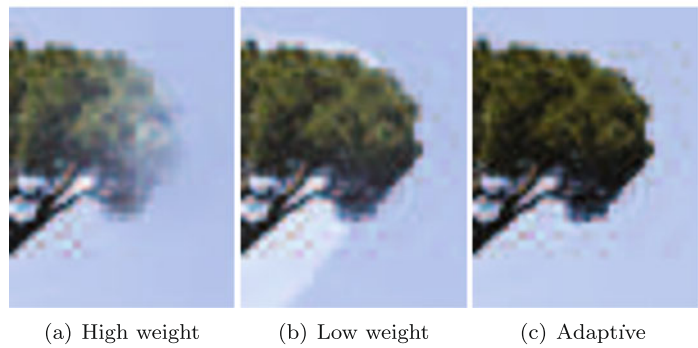
values and produces an image I_{seam} with visible seams. The curl of the gradient field of I_{seam} is null since it is computed from an actual image. And, inside the selection, gradients are equal to the foreground values since pixels have been copied. The same holds outside the selection with the background values. However, on the boundary, gradients are different from either the foreground or the background, which generates the seams. We address the shortcomings of this naive solution by seeking gradient values that minimize the curl and are close to the gradients of the input images.

A Least-Squares Approach We formulate our goal using a least-squares energy where the desired gradients **v** on the boundary are the unknowns. The first term minimizes the curl: $\int_\beta [\text{curl}(\mathbf{v})]^2$ and the second term keeps the values close to the input gradients $\int_\beta (\mathbf{v} - \nabla F)^2 + \int_\beta (\mathbf{v} - \nabla B)^2$. This last term has the same effect as keeping **v** close to the average gradient. We combine the two terms to obtain:

$$\operatorname*{argmin}_{\mathbf{v}} \int_\beta \left([\text{curl}(\mathbf{v})]^2 + W_\beta \left[\mathbf{v} - \frac{1}{2}(\nabla B + \nabla F) \right]^2 \right) \qquad (4)$$

where W_β controls the importance of the second term.

Adaptive Weights Figure 3 shows results for several values of W_β. For large W_β, we only minimize the proximity to the input gradients, which is the standard gradient compositing with seamless boundaries but leaking artifacts. For a small W_β, we have the naive solution described above where we only minimize the curl. There are no bleeding artifacts but the boundary is visible. We combine these two behaviors by varying the weights according to the local image structure.

Intuitively, a seamless boundary is desirable when both sides of the boundary are smooth. This is the case for instance when we stitch a sky region with another sky region. A seamless boundary is also acceptable when both sides are textured because leaking is a low-frequency phenomenon that will be hidden by visual masking. Figure 5 illustrates this effect that has also been used in the rendering literature (Drettakis et al. 2007; Ramanarayanan et al. 2007, 2008; Vangorp et al. 2007). In these two cases, we seek high

values for W_β. But when a textured region is composited adjacent to a smooth region, we want to prevent bleeding because such regions would generate unpleasing artifacts on the smooth side, e.g. in the sky. In this case, we want low values of W_β. The following paragraph explains how we compute W_β based on the local amount of texture.

Estimating the Local Amount of Texture Our strategy relies on the presence or absence of texture in a given neighborhood. In this paragraph, we describe a simple and computationally efficient texture estimator although one could use other models (Su et al. 2005; Bae et al. 2006). Formally, our scheme is:

$$T_{\sigma_1,\sigma_2}(\mathbf{g}) = \frac{G_{\sigma_1} \otimes \|\mathbf{g}\|}{G_{\sigma_2} \otimes \|\mathbf{g}\|} n(\|\mathbf{g}\|) \qquad (5)$$

where **g** is a gradient field, G_σ is a Gaussian of width σ, σ_1 and σ_2 are two parameters such that $\sigma_1 < \sigma_2$, $\otimes$ is the convolution operator, and $n(\cdot)$ a noise-controlling function. Our scheme relies on image gradients, for instance $T(\nabla I)$ is the texture map of the image I as shown in Fig. 4. We compare the average amplitude of the gradients in two neighborhoods defined by σ_1 and σ_2, effectively comparing a small gradient neighborhood over a larger gradient neighborhood. If the image is locally textured, then the average in the small neighborhood will be higher than in the large neighborhood, corresponding to $T > 1$. Conversely, $T < 1$ corresponds to regions with locally less texture than in the larger neighborhood. This scheme would be sensitive to noise in smooth regions where gradients are mostly due to noise. We address this issue with the function n that is equal to 0 for very small gradient and 1 otherwise. In practice, we use a smooth step equal to 0 for the bottom 2 % of the intensity scale and 1 for 4 % and above. In our context, the goal is to differentiate textureless areas from textured regions; the relative amount of texture in textured regions does not matter. Consequently, we use the estimator $\bar{T} = \min(1, T)$ that considers all textured regions to be equal.

Computing the Boundary Weights Recall that we want W_β to be large when both foreground and background regions
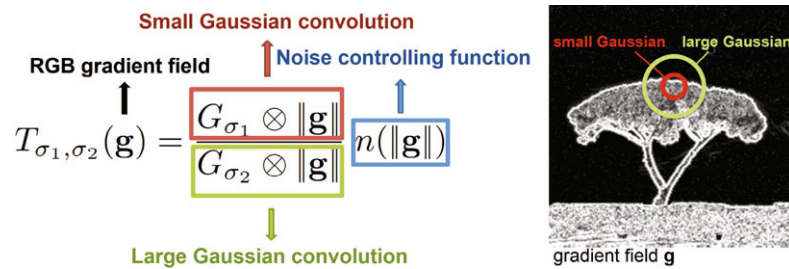
Fig. 4 We illustrate our equation for estimating the local amount of texture. We use our computed gradient field g, to compute the local texture. A small Gaussian shown in *red* and large Gaussian shown in *green* are used. To avoid noisy results, we also use a noise controlling function (Color figure online)

Fig. 5 We show the same dots on a uniform background (**a**) and on a photograph (**b**) but the two dots on the tree are not visible because of the texture of the foliage

(a) dots on a gray background   (b) same dots on a photograph
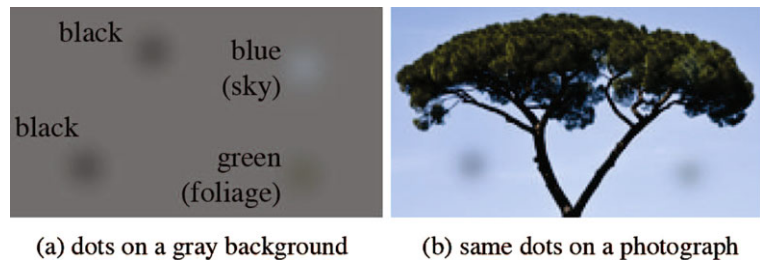
have the same degree of texture, either both smooth or both textured. If one of them is smooth and the other textured, we want W_β to be small. We consider the difference $D = |\bar{T}(\nabla F) - \bar{T}(\nabla B)|$ and define W_β using a function that assigns a small value w when D is large, 1 when D is small, and linearly transitions between both values. Formally, we use:

$$W_\beta = \begin{cases} w & \text{if } D > \tau \\ \min(1, w + \lambda(1 - D/\tau)) & \text{otherwise} \end{cases} \quad (6)$$

where λ and τ control the linear transition. We found that $\lambda = 4$, $w = 0.05$, $\tau = 0.005$, $\sigma_1 = 0.5$, and $\sigma_2 = 2$ work well in practice. All results are computed with these values unless otherwise specified.

Discussion Figure 6 illustrates the effect of our approach that reduces the curl on the compositing boundary. Bleeding artifacts are significantly reduced. In next section, we describe how to remove the remaining leaks. For color images, we use RGB gradients in Eq. (5) so that we can distinguish textures of different colors effectively. Our texture estimation also assumes some sort of irregularity in the textured areas. One advantage to this is that in regularly dashed regions or half-tones, the estimator will register the areas as less textured, which prevents color bleeding from entering these regions. As described in our results, we used a simple scheme to compute textures, but a more complex and accurate texture estimator can also be used. From an efficiency standpoint, an important characteristic of our approach is that it can be solved with a sparse linear system since our least-squares energy (Eq. (4)) involves only sparse linear operators and W_β depends only on the input data.
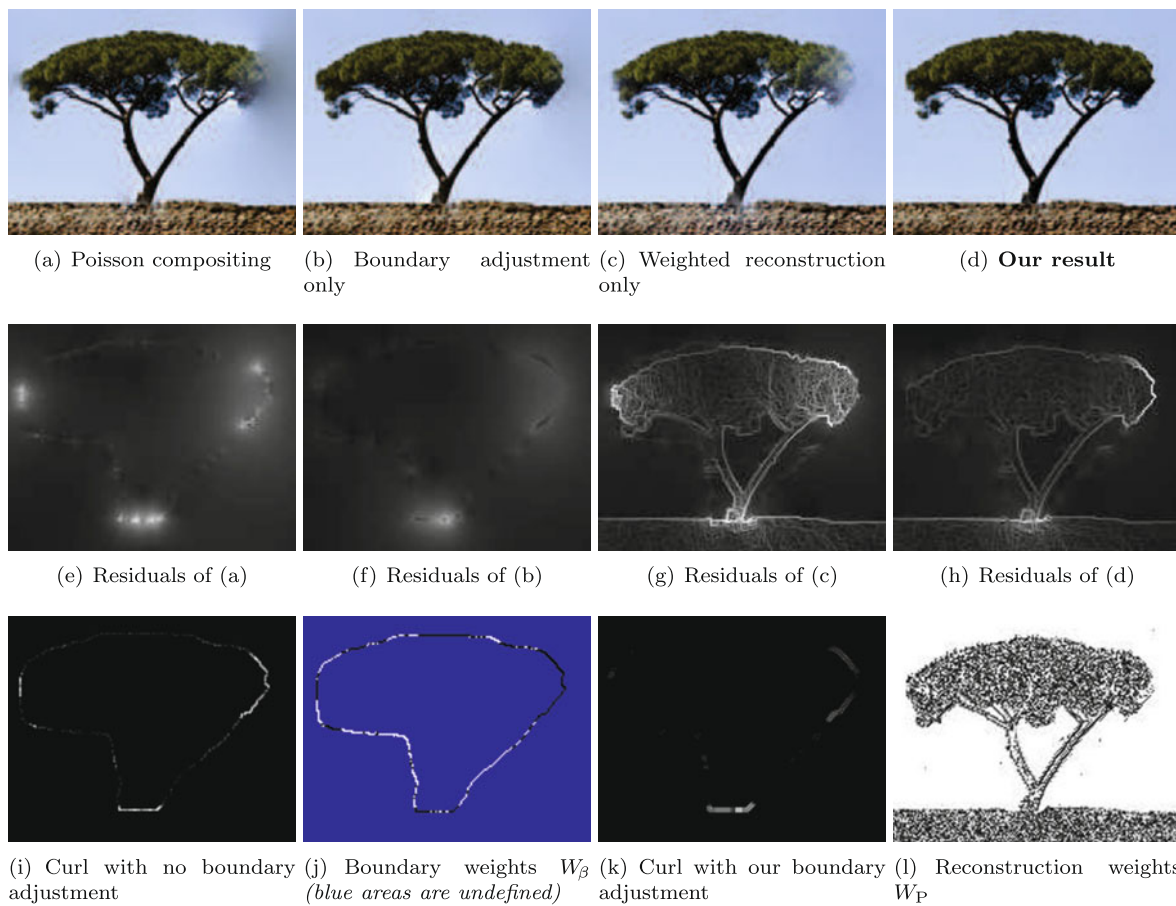
3 Controlling the Location of the Residuals

Although our boundary treatment reduces the curl of the gradient field **v**, in general **v** is not integrable. As with other gradient-domain methods, our goal is to produce an image with a gradient field ∇I as close as possible to **v**. Our strategy is to modify the Poisson equation (Eq. (2)) in order to locate the residuals as much as possible in regions where they will be the least objectionable. Intuitively, we want to avoid errors in smooth regions such as the sky where they produce color leaks and halos, and put them in textured areas where visual masking will conceal the artifacts (Fig. 5).

3.1 Adapting the Poisson Equation

Let's assume that we have a scalar map W_P with high values in regions where errors would be visible and low values otherwise. We discuss later how to compute such a function using our texture estimator $\bar{T}$. Given W_P, we modulate the least-squares strength so that we penalize less the regions where we prefer the residuals to be, that is, regions with low W_P values:

$$\underset{I}{\arg\min} \int W_P \|\nabla I - \mathbf{v}\|^2 \quad (7)$$

Since we want to reduce the difference between ∇I and **v**, W_P has to be strictly positive everywhere. Moreover, to keep

(a) Poisson compositing (b) Boundary adjustment only (c) Weighted reconstruction only (d) **Our result**

(e) Residuals of (a) (f) Residuals of (b) (g) Residuals of (c) (h) Residuals of (d)

(i) Curl with no boundary adjustment (j) Boundary weights W_β *(blue areas are undefined)* (k) Curl with our boundary adjustment (l) Reconstruction weights W_P

Fig. 6 We illustrate the effect of the two parts in our approach: curl reduction at the boundary and weighted reconstruction. As a comparison, the Poisson reconstruction (**a**) exhibits high residuals, (**e**), and curl not equating zero, (**i**), at badly cut regions. Our boundary weights minimize the residuals at badly cut regions (**f**). The weights illustrated in *black* adhere to curl equating to zero (**j**). As a result, the edges are more well-defined as seen in (**b**). With the weighted reconstruction, the residuals account for the texture of the image as shown in (**g**). The weights are shown in (**k**), where white areas represent astringent control of bleeding (**g**). By combining the curl reduction and weighted reconstruction, the residuals of the edges are reduced while texture is respected in (**h**). As a result, areas with curl not equating to zero are minimized as shown in (**l**), producing a more desirable result (**d**) (Color figure online)

our approach computationally efficient, we will design W_P such that it does not depend on the unknown image I. In this case, Eq. (7) is a classical least-squares functional that can be minimized by solving a linear system. To obtain a formula similar to the Poisson equation (2), we apply the Euler-Lagrange formula (Aubert and Kornprobst 2002). Recall that W_P does not depend on I. Thus, we obtain the following linear system:

$$\mathrm{div}\big(W_P(\nabla I - \mathbf{v})\big) = 0 \qquad (8)$$
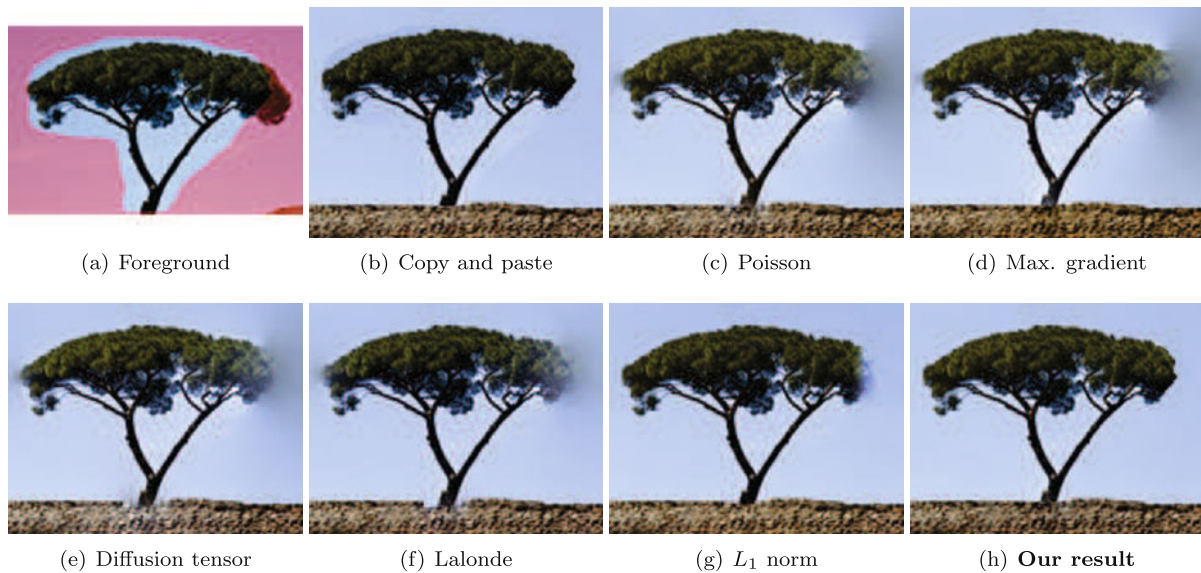
In Sect. 3.2, we show that although this equation is simple, it has favorable properties.

Computing the Weights To keep our scheme linear, we do not use any quantity related to the unknown I. We use the desired gradient field $\mathbf{v}$ to estimate the texture location in the image. Although $\mathbf{v}$ is not equal to the gradient of the final output, it is a good approximation that is sufficient to compute the weights W_P. Since we want high weights in smooth regions and low weights in textured areas, we use the following formula: $W_P = 1 - p\bar{T}(\mathbf{v})$ where p is a global parameter that indicates how much we control the residual location. For instance, $p = 0$ corresponds to no control, that is, to the standard Poisson equation, whereas larger values impose more control. p has to be strictly smaller than 1 to keep $W_P > 0$. We found that values close to 1 performs better in practice. We use $p = 0.999$ in all our results. We also found that it is useful to have a more local estimate of the texture, which we achieve using $\sigma_1 = 0$ to compute $\bar{T}$ while keeping the other parameters unchanged. Figure 6 shows a sample weight map.

3.2 Analysis of the Residual Structure

Independent of the actual definition of W_P, we can show that the residuals produced by our approach have structure that

(a) Foreground   (b) Copy and paste   (c) Poisson   (d) Max. gradient

(e) Diffusion tensor   (f) Lalonde   (g) L_1 norm   (h) **Our result**

Fig. 7 We compare several approaches on an example where we composite a tree on a sky background. To test the robustness against selection accuracies, we introduce three errors (**a**): a small error on the *left*, a large error on the *right*, and the trunk is inserted in the ground. A direct copy-and-paste produces an image with visible seams in the sky region (**b**). Poisson compositing (Prez et al. 2003) (**c**), maximum gradient (Prez et al. 2003) (**d**), diffusion (Agrawal et al. 2006) (**e**), Photo Clip Art (Lalonde et al. 2007) (**f**), and robust Poisson reconstruction using the L_1 norm (Reddy et al. 2009) (**g**) generate seamless boundaries but suffer from bleeding artifacts where the selection cuts through the foliage and also at the contact between the trunk and the ground. In comparison, our method (**h**) produces artifact-free results. We provide more comparisons in supplemental material

is aligned with the image content. Wang et al. (2004) have demonstrated that such structural similarity produces more acceptable results. To better understand the role of W_P, we distribute the divergence in Eq. (8): $W_P \operatorname{div}(\nabla I - \mathbf{v}) + \nabla W_P \cdot (\nabla I - \mathbf{v}) = 0$. With $W_P \neq 0$, the relation $\operatorname{div}(\nabla I) = \Delta I$, and the logarithmic gradient $\nabla W_P / W_P = \nabla \log W_P$, we obtain:

$$\underbrace{\Delta I - \operatorname{div}(\mathbf{v})}_{\text{Poisson term}} + \underbrace{\nabla \log W_P \cdot (\nabla I - \mathbf{v})}_{\text{new term}} = 0 \qquad (9)$$

The left term is the same as the standard Poisson equation (2) while the right term is new. In regions where W_P is constant, the new term is null and our scheme behaves as the Poisson equation, that is, it spreads the residuals uniformly. In other regions where W_P varies, our scheme differs from the Poisson equation and allows for discontinuities in the residual. Since W_P measures the amount of texture, it means that residual variations are aligned with texture edges, which ensures the structural similarity that has been shown desirable by Wang et al. (2004). We provide illustrations of this property in Fig. 6.

3.3 Relationship with Existing Methods

For this section, we make explicit the variable W_P, that is, Eq. (7) becomes $\int W_P(\mathbf{v}) \| \nabla I - \mathbf{v} \|^2$, and Eq. (8), $\operatorname{div}(W_P(\mathbf{v}) (\nabla I - \mathbf{v})) = 0$. We discuss the relationships among our work and related methods independently of the actual definition of W_P.

The Poisson Equation and its Variants Rewriting the Poisson equation (2) as $\operatorname{div}(\nabla I - \mathbf{v}) = 0$, we see that our linear system has the same complexity since we do not introduce new unknowns nor new coefficients in the system; we only reweight the coefficients. Agrawal et al. (2006) also describe an anisotropic variant that is linear. However, while this method performs well in shape-from-shading, it does not prevent bleeding when applied to image compositing (Fig. 7). The L_1 reconstruction method that Reddy et al. (2009) propose in the context of shape-from-shading has the same difficulty with image compositing (Fig. 7).

Edge-preserving Filtering Our method is also related to Farbman's edge-preserving filter (Farbman et al. 2008) that minimizes an attachment term plus $\int W_P(I_0) \| \nabla I \|^2$ where I_0 is the input image. Farbman projects the formula on the x and y axes but we believe that it does not have a major impact on the results. More importantly, Farbman's method and ours share the idea of using a modulation W_P that depends on fixed quantities and preserves the least-squares nature of the problem; Farbman uses the input image I_0 and we use the target gradient field $\mathbf{v}$. Finally, our work has common points with Perona and Malik's nonlinear anisotropic diffusion filter (Perona and Malik 1990): $\partial I / \partial t = \operatorname{div}(W_P(\nabla I) \nabla I)$. The difference is that our modulation term W_P is not a function of the image I which makes our equation linear, and we have a term $\nabla I - \mathbf{v}$ instead of ∇I, which can be interpreted as Perona and Malik
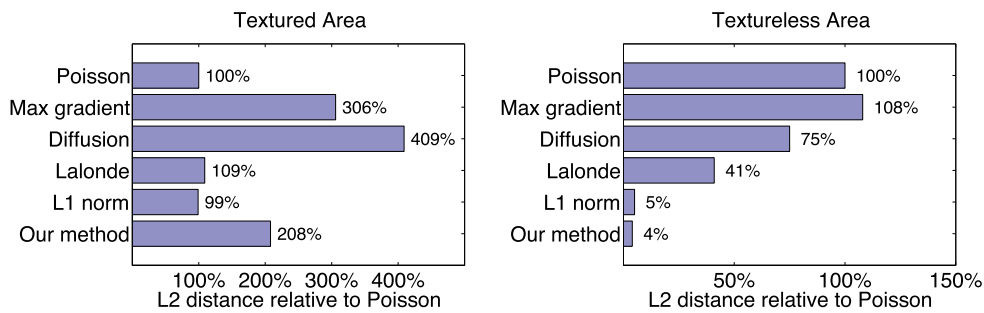
Fig. 8 We numerically evaluate bleeding introduced by different methods. We selected two 11×11 regions in the tree example (Fig. 7), one within the textured area below the trunk and one in the sky on right of the foliage. We compute the L_2 RGB difference between the image before and after compositing, normalized relative to Poisson compositing; that is, 100 % indicates as much bleeding as Poisson and 0 % indicates no bleeding. In the textured region (*left*), all methods bleed but the bleeding is masked by the high frequency texture. In the textureless area (*right*), most methods cause visible bleeding, which is particularly visible in this smooth region. The L_1-norm and our method achieve similarly low values which confirm minimal bleeding. But in a number of cases, the L_1-norm method introduces an undesirable color cast shown in the tree example, whereas our method yields a satisfying output

"diffuse gradients" whereas we "diffuse integration residuals."

4 Results

We demonstrate our approach on a typical hand-made compositing scenario which may generate inaccurate selections (Figs. 7, 13, and 15). We also show that our approach applies to heterogeneous panorama stitching (Sivic et al. 2008) (Fig. 12) and image reshuffling (Cho et al. 2010) (Fig. 14). More results are in our supplemental material. All the results are computed using the same parameters unless otherwise specified. These settings performed well in all of our experiments. Parameter variations are also shown in the supplemental material.

Quantitative Evaluation We use direct compositing and Poisson compositing as baselines to estimate how much bleeding occurs. For direct compositing, we directly copy pixel values and the result I_d exhibits visible seams but not bleeding. For Poisson compositing, we copy gradient values and solve the Poisson equation. The result I_P is seamless but colors leak where the selection is inaccurate. Then we consider an image I, pick a pixel $\mathbf{p}$ in the potential leaking area, and compute: $\|I(\mathbf{p}) - I_d(\mathbf{p})\| / \|I_P(\mathbf{p}) - I_d(\mathbf{p})\|$. Expressed in percentages, 0 % indicates no bleeding at all and 100 % indicates as much bleeding as Poisson compositing. Figure 8 compares the results of several methods on the tree example (Fig. 7). In texture areas, all methods including ours introduce significant bleeding. But because of visual masking, it is mostly invisible and is not an issue. In smooth regions, only the L_1-norm method and ours produce nearly bleeding-free images—however, the L_1-norm method also generates visible artifacts inside the pasted regions whereas our method does not (Fig. 7). We repeated this experiments
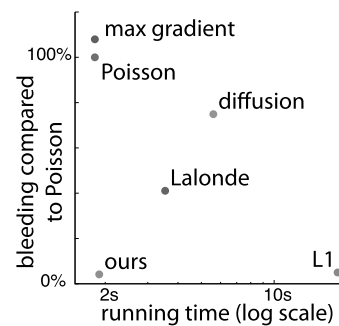

Fig. 9 This plot locates each method according to its speed and how much bleeding it introduces in the sky region on Fig. 7 as reported in Fig. 8. Our method is as fast as the standard Poisson solver while introducing almost no bleeding. In comparison, the other methods are slower and generate color leaks. Note that the L_1 method does not produce bleeding artifact on this example but it creates a severe color cast (Fig. 7)

on smooth regions in several images and obtained equivalent results. Existing methods perform well on average but also fail in some cases. In comparison, our approach always improves over standard Poisson compositing (Fig. 10). These quantitative results follow closely with human perception of the residual errors as described in Fig. 5.

Parameter Changes In this section, we will refer to Eqs. (5) and (6) and explore the impact parameters make on the results.

Changing λ affects the adaptivity of using seamless or hard egdes around the selection. With a low value of λ ($\lambda \ll 1$), the boundaries of the selection will have seams. However, with a high λ value ($\lambda \gg 1$), we get a larger adaptation ability of around the boundaries. This disadvantage of high values is that results can be unpredictable as boundaries may vary greatly in different areas. The value used balances these two properties.
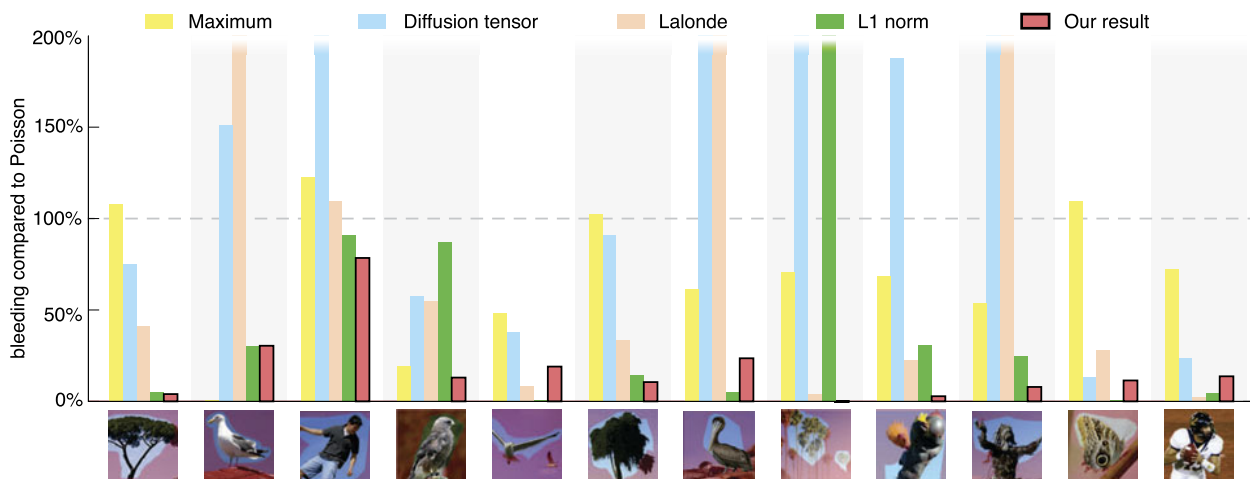
Fig. 10 Estimation of the amount of bleeding in a textureless area over several images. All existing methods have performances that largely vary depending on the image. In particular, all of them sometimes introduce more bleeding than the original Poisson method. In compari- son, our approach performs consistently well. The images used for this test are provided in supplemental material and the code to compute these values is also included. We have truncated bars larger than 200 %

Fig. 11 In some cases, when compared to the input (**a**), Poisson compositing (**b**) and our approach (**c**) discolor the pasted region. See text for details

(a) Foreground   (b) Poisson   (c) **Our result**

The parameters w and τ effectively control the amount of leakage coming from the background to the foreground image. With small values of w or τ ($\ll 1$), seams will start to appear to prevent the leakage. With large values of w or τ ($\gg 1$), leakage from the background to foreground is easily visible. The value used balances these two properties.

The parameters σ_1 and σ_2 control the texture estimation. Too low of a value for σ_1 will make the texture-estimation less discriminative because the operation will become too local. A high value will also pose the same problem of be- ing less discriminative because the operation becomes too global, losing local information. Ideally, σ_2 should be a sig- nificantly larger than σ_1, but with too large of σ_2, perfor- mance becomes an issue because we have to use a blur with a larger kernel.

With the balanced values we chose, the results are robust throughout many different images. All of our evaluations and examples are done with the same parameters. Again, we found that $\lambda = 4$, $w = 0.05$, $\tau = 0.005$, $\sigma_1 = 0.5$, and $\sigma_2 = 2$ work well in practice. All results are computed with these values unless otherwise specified.

Complexity   We compute the final result in two linear steps. This is equivalent to a single linear system because I is

a linear function of $\mathbf{v}$ (Eq. (8)) and $\mathbf{v}$ is a linear function of B and F (Eq. (4)). Further, only sparse operators are involved: divergence, curl, and weights that correspond to diagonal matrices. Compared to the Poisson equation, we solve for the same number of unknowns, that is, the number of pixels in I. The only overhead is the computation of $\mathbf{v}$, for which the number of unknowns is orders of magnitude smaller, since only pixels on the boundary are concerned. To summarize, our method has the same complexity as the Poisson equation. In comparison, nonlinear methods (Reddy et al. 2009) require more complex iterative solvers. Figure 9 shows that our implementation achieves timings similar to the Poisson reconstruction, resulting in a run-time faster than most other implementations while introducing almost no bleeding.

Discussion   Although our method produces high quality outputs, a close examination reveals that the boundary can be sometimes overly sharp. This minor issue is difficult to spot at first and less conspicuous than color leaks. Nonethe- less, matching the sharpness of other edges in the im- age would be an interesting extension to this work. An- other promising extension is to combine our method with optimized boundary selection techniques to further reduce

(a) Photoshop Auto Blend

(b) close-up

(c) **Our result**

(d) close-up
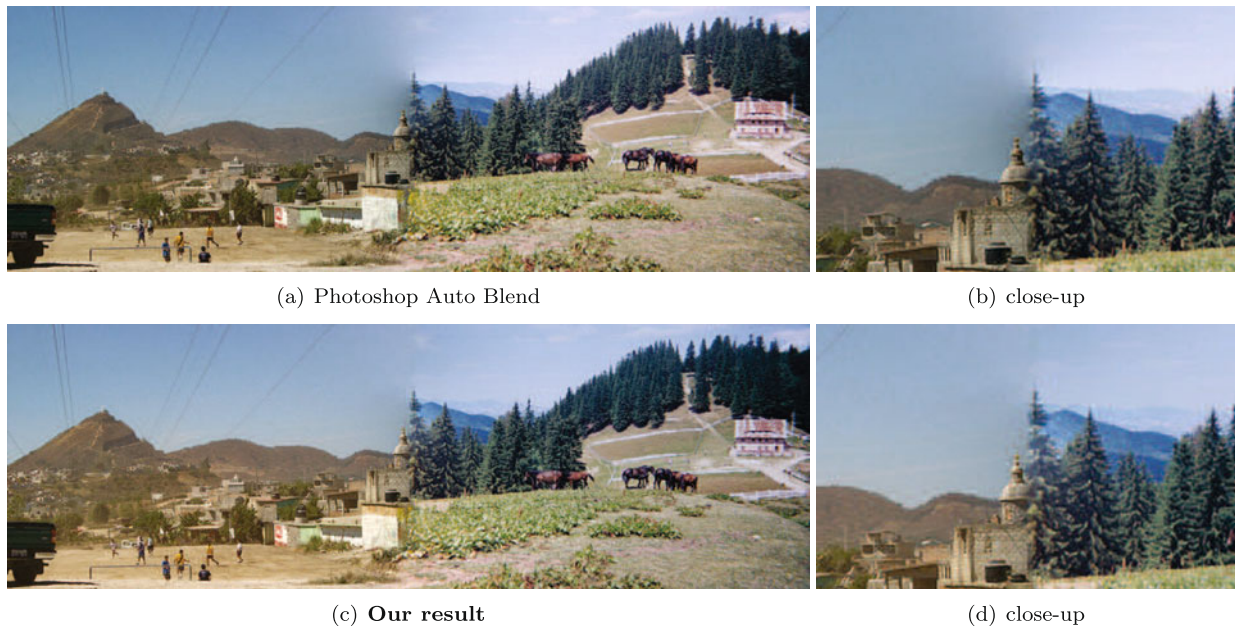
Fig. 12 For a heterogeneous panorama, Photoshop Auto Blend (Agarwala 2007) produces strong bleeding near the cut. In comparison, our method significantly improves the result. Our approach also performs better than to other methods on this challenging case as shown in supplemental material


(a) pixel cut-and-paste   (b) our result   (c) pixel cut-and-paste   (d) our result
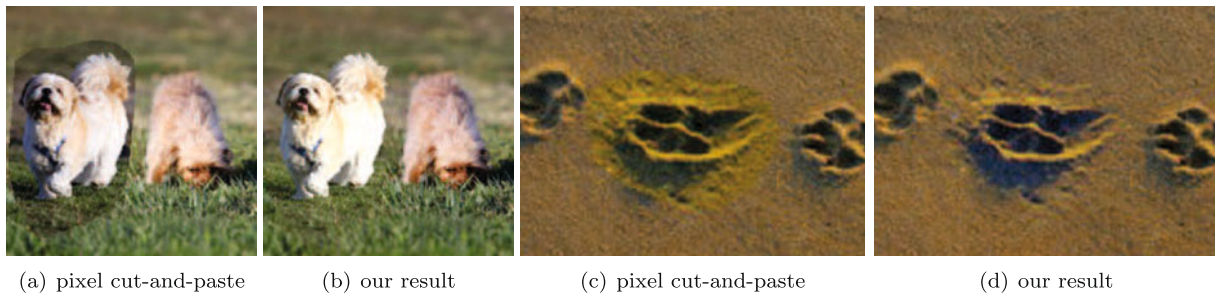
Fig. 13 Our method also achieves satisfying composite when the entire image is textured

(a) Blending used by Cho et al.   (b) close-up   (c) **Our result**   (d) close-up
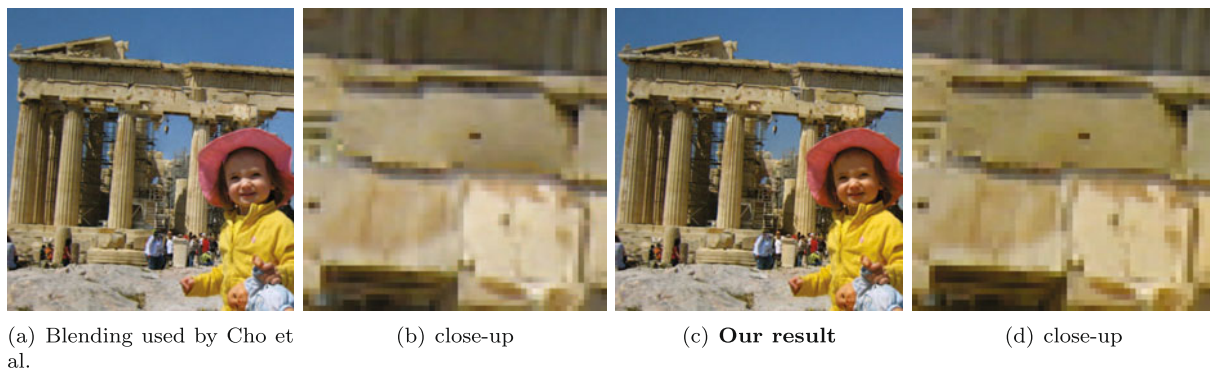
Fig. 14 Compared to the blending approach proposed by Cho et al. (2010) (**a**, **b**), our approach (**c**, **d**) improves the result of image reshuffling. We used the same patch locations and boundaries as Cho et al. but applied our method which yields better results than the Poisson-based blending proposed in the original article (Cho et al. 2010). In particular, our result produces more faithful colors but does have local color leaks as can be seen on the close-up (zoom of a region above the girl's hat). This result may be better seen in the supplemental material. Data courtesy of Tim Cho

(a) Diffusion tensor   (b) Lalonde   (c) L_1 norm   (d) Our result
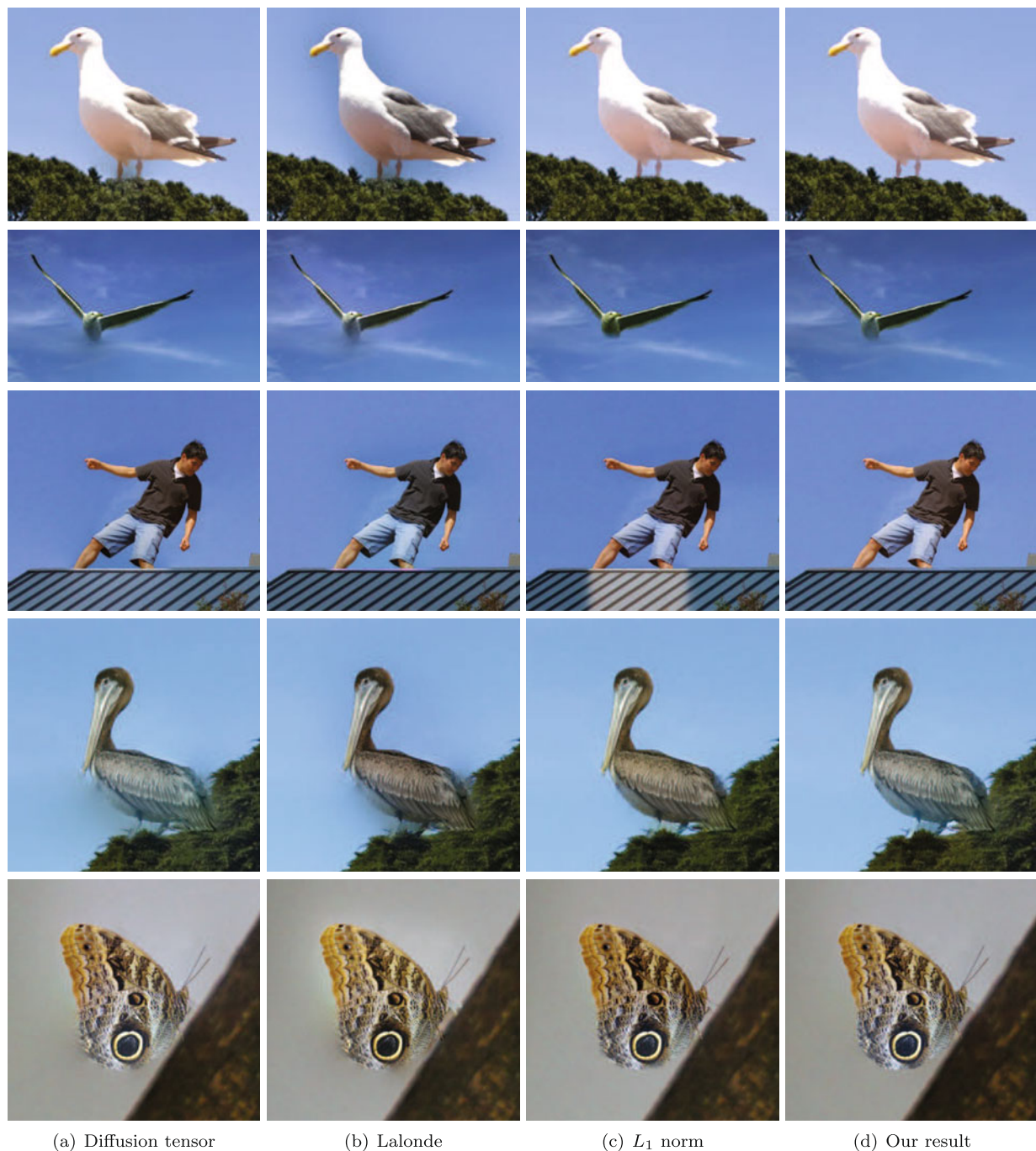
Fig. 15 Sample composites using different methods. We show the 3 existing techniques that usually perform best and our method. The method based on diffusion tenors and the one by Lalonde et al. reduce the bleeding artifacts but they nonetheless remain visible. The L_1- norm approach regularly achieves artifact-free results but sometimes completely fails, e.g. "the man on the roof". Our method consistently produces satisfying results

bleeding artifacts. As other gradient-domain methods, our method can yield some discoloration (Fig. 11 and supplemental material). This effect is often desirable to achieve seamless blending. If one wishes to preserve the original colors, matting can be a solution but it often requires a more careful user input. We also found that our approach

is useful in challenging applications such as heterogeneous panorama stitching (Sivic et al. 2008) where mismatches are common place (Fig. 12). In this case, we found that our method performs better with a smoother transition from seamless and leak-free compositing, which is achieved by setting $\tau = 0.01$ in Eq. (6). We also tested our algorithm on

images where texture is present everywhere to confirm that it is also able to achieve seamless composites in this case (Fig. 13).

5 Conclusion

We have described an image-compositing method that is robust to selection inaccuracies. The combination of low-curl boundaries and a weighted reconstruction based on visual masking produces artifact-free results on a broad range of inputs, in particular where other methods have difficulties. In addition, the solution is linear and has similar complexity to the standard Poisson equation. With robust results and speed, our method is a suitable replacement for the standard Poisson equation in many computer vision applications.

References

Agarwala, A. (2007). Efficient gradient-domain compositing using quadtrees. In *ACM transactions on graphics: Vol. 26. Proceedings of the ACM SIGGRAPH conference*.

Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D. H., & Cohen, M. F. (2004). Interactive digital photomontage. In *ACM transactions on graphics: Vol. 23. Proceedings of the ACM SIGGRAPH conference* (pp. 294–302).

Agrawal, A., Raskar, R., & Chellappa, R. (2006). What is the range of surface reconstructions from a gradient field? In *Proceedings of the European conference on computer vision*.

Aubert, G., & Kornprobst, P. (2002). *Applied mathematical sciences: Vol. 147. Mathematical problems in image processing: partial differential equations and the calculus of variations*. Berlin: Springer.

Bae, S., Paris, S., & Durand, F. (2006). Two-scale tone management for photographic look. In *ACM transactions on graphics: Vol. 25. Proceedings of the ACM SIGGRAPH conference* (pp. 637–645).

Bhat, P., Zitnick, C. L., Cohen, M., & Curless, B. (2009). Gradientshop: a gradient-domain optimization framework for image and video filtering. In *ACM transactions on graphics*.

Cho, T. S., Avidan, S., & Freeman, W. T. (2010). The patch transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(8), 1489–1501.

Drettakis, G., Bonneel, N., Dachsbacher, C., Lefebvre, S., Schwarz, M., & Viaud-Delmon, I. (2007). An interactive perceptual rendering pipeline using contrast and spatial masking. *Rendering Techniques*.

Farbman, Z., Fattal, R., Lischinski, D., & Szeliski, R. (2008). Edge-preserving decompositions for multi-scale tone and detail manipulation. In *ACM transactions on graphics: Vol. 27. Proceedings of the ACM SIGGRAPH conference*.

Farbman, Z., Hoffer, G., Lipman, Y., Cohen-Or, D., Fattal, R., & Lischinski, D. (2009). Coordinates for instant image cloning. In *ACM transactions on graphics: Vol. 28. Proceedings of the ACM SIGGRAPH conference*.

Finlayson, G. D., Hordley, S. D., Lu, C., & Drew, M. S. (2006). On the removal of shadows from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*, 59–68.

Finlayson, G. D., Drew, M. S., & Lu, C. (2009). Entropy minimization for shadow removal. *International Journal of Computer Vision*, *85*(1), 35–57.

Georgiev, T. (2006). Covariant derivatives and vision. In *Proceedings of the European conference on computer vision*.

Hays, J., & Efros, A. A. (2007). Scene completion using millions of photographs. In *ACM transactions on graphics: Vol. 26. Proceedings of the ACM SIGGRAPH conference*.

Jia, J., Sun, J., Tang, C. K., & Shum, H. Y. (2006). Drag-and-drop pasting. In *ACM transactions on graphics: Vol. 25. Proceedings of the ACM SIGGRAPH conference*.

Lalonde, J. F., Hoiem, D., Efros, A., Rother, C., Winn, J., & Criminisi, A. (2007). Photo clip art. In *ACM transactions on graphics: Vol. 26. Proceedings of the ACM SIGGRAPH conference*.

Levin, A., Zomet, A., Peleg, S., & Weiss, Y. (2006). Seamless image stitching in the gradient domain. In *Proceedings of the European conference on computer vision*.

Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*, 629–639.

Prez, P., Gangnet, M., & Blake, A. (2003). Poisson image editing. In *ACM transactions on graphics: Vol. 22. Proceedings of the ACM SIGGRAPH conference*.

Ramanarayanan, G., Ferwerda, J., Walter, B., & Bala, K. (2007). Visual equivalence: towards a new standard for image fidelity. In *ACM transactions on graphics: Vol. 26. Proceedings of the ACM SIGGRAPH conference*.

Ramanarayanan, G., Bala, K., & Ferwerda, J. (2008). Perception of complex aggregates. In *ACM transactions on graphics: Vol. 27. Proceedings of the ACM SIGGRAPH conference*.

Reddy, D., Agrawal, A., & Chellappa, R. (2009). Enforcing integrability by error correction using L-1 minimization. In *Proceedings of the conference on computer vision and pattern recognition*.

Sivic, J., Kaneva, B., Torralba, A., Avidan, S., & Freeman, W. T. (2008). Creating and exploring a large photorealistic virtual space. In *Proceedings of the IEEE workshop on internet vision*.

Su, S., Durand, F., & Agrawala, M. (2005). De-emphasis of distracting image regions using texture power maps. In *Proceedings of the ICCV workshop on texture analysis and synthesis*.

Tappen, M. F., Adelson, E. H., & Freeman, W. T. (2005). Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 1459–1472.

Vangorp, P., Laurijssen, J., & Dutr, P. (2007). The influence of shape on the perception of material reflectance. In *ACM transactions on graphics: Vol. 26. Proceedings of the ACM SIGGRAPH conference*.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(14), 600–612.

Whyte, O., Sivic, J., & Zisserman, A. (2009). Get out of my picture! Internet-based inpainting. In *Proceedings of the British machine vision conference*.