Graphical Models 73 (2011) 97–110

Contents lists available at ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

Sketch-based warping of RGBN images

Thiago Pereira [a,*], Emilio Vital Brazil [a], Ives Macêdo [a], Mario Costa Sousa [b], Luiz Henrique de Figueiredo [a], Luiz Velho [a]

[a] IMPA, Estrada Dona Castorina 110, Rio de Janeiro, RJ 22460-320, Brazil
[b] University of Calgary, 2500 University Dr. NW, Calgary, Alberta, Canada T2N 1N4

ARTICLE INFO

Article history:
Received 13 August 2010
Received in revised form 14 November 2010
Accepted 24 November 2010
Available online 13 December 2010

Keywords:
RGBN Images
Image deformations
Sketch-based Interfaces
Warping
Hermite–Birkhoff radial basis functions

ABSTRACT

While current image deformation methods are careful in making the new geometry seem right, little attention has been given to the photometric aspects. We introduce a deformation method that results in coherently illuminated objects. For this task, we use RGBN images to support a relighting step integrated in a sketch-based deformation method. We warp not only colors but also normals. Normal warping requires smooth warping fields. We use sketches to specify sparse warping samples and impose additional constraints for region of interest control. To satisfy these new constraints, we present a novel image warping method based on Hermite–Birkhoff interpolation with radial basis functions that results in a smooth warping field. We also use sketches to help the system identify both lighting conditions and material from single images. We present results with RGBN images from different sources, including photometric stereo, synthetic images, and photographs.

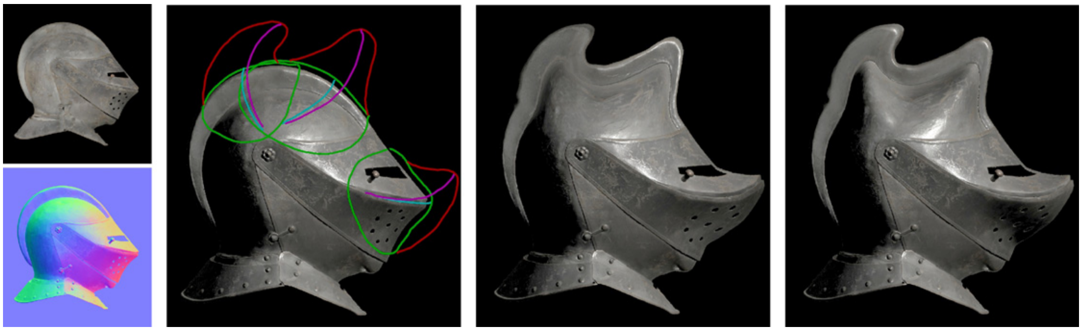
1. Introduction

Deformation is a well-studied problem in computer graphics. It started with warping and morphing methods for images [1] and evolved into methods for deforming 3D triangular meshes, and more recently volume data [2]. During image warping, it is important to contemplate the 3D world. In fact, most images consist of projections of 3D objects. Each pixel stores a color, that is the result of a complex illumination process overlooked in traditional image warping. Current image deformation methods only aim at transforming geometry consistently, paying no attention to photometric aspects. In some cases, especially for small deformations, lighting may be wrong but perceptually acceptable. In other cases, lighting is wrongly depicted and perceived. We propose the use of information from albedo and normals to design an illumination-aware deformation method. Our results are visually consistent regarding geometric and photometric aspects, even under extreme deformations (Fig. 1). Our method can be applied to change shape and light not only in static images but also in dynamic settings such as image-based animations and interactive images for electronic magazines.

Our method is entirely image-based, not relying on 3D models at any stage. This is an advantage since image artists may not have modeling skills. Our technique is only made possible due to advances in vision that introduced a new 2.5D image data structure: the RGBN image [3], a photo in which each pixel contains not only color but also a normal vector. RGBN images can be obtained in several ways. First, they can be the result of a 3D rendering process or simulation. A second approach is photometric stereo [4], which uses multiple input images from the same viewpoint, each image illuminated with different but known light positions. Recent user-assisted approaches [5–7] allow obtaining RGBN images even from single images. Finally, in Lumo [8] Johnston creates normals for cel animation.

* Corresponding author. Fax: +55 21 25295067.
E-mail addresses: tpereira@impa.br (T. Pereira), emilio@impa.br (E.V. Brazil), ijamj@impa.br (I. Macêdo), smcosta@ucalgary.ca (M.C. Sousa), lhf@impa.br (L.H. de Figueiredo), lvelho@impa.br (L. Velho).

1524-0703/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.gmod.2010.11.001



(a) RGBN image (b) Shaded input and sketches (c) Traditional image warping (d) RGBN image warping

Fig. 1. Our system takes as input albedo and normals (a), and a shaded image (b). In addition, the user makes two kinds of sketches (b): warping sketches specify a region of interest and its deformed shape; lighting sketches help the system estimate lighting and material. We relight the warped result (c) and obtaining objects illuminated coherently (d). Simple color warping stretches the original highlight and misses some shadows. With relighting, we recreate fine highlights and shadows in the right places. Three sequential deformations were used.

We chose to work with RGBN images because they enable relighting, a key component of our illumination-aware deformation method. We provide the user with a simple sketch-based interface to specify a deformation (Fig. 1b). Unlike previous methods, we transport not only the colors but also the normal vectors during the deformation. With the warped RGBN in hand, the system can relight the new image using the appropriate illumination. Sketch-based tools allow to the user to specify these lighting conditions intuitively. Additionally, we only relight the image in the deformed region; and therefore, only local illumination needs to be detected.

Our main contributions are:

- a quality hybrid (interactive and automatic) image deformation pipeline that consistently warps both geometry and illumination components (highlights, midtones, and shadows) of the source image;
- a sketch-based method that specifies displacement fields with region of interest control on images, under a variety of user-defined paths;
- a novel warping method that employs RBF-based Hermite–Birkhoff interpolation to implement generalized warping constraints, including Jacobian restrictions, to produce smooth fields.

In addition to RGBN images, these methods are also original for traditional color images.

2. Related work

2.1. Sketch-based deformation

Image deformation methods have attracted the attention of researchers and there are many different solutions regarding control. Some approaches use control points or straight lines [9,10]. However, these are usually inconvenient for specifying curve restrictions, requiring too many points.

Sketching from the user [11] is a more natural way to communicate deformations in images [12–14], because artists typically want to define complex shapes intuitively and quickly. Weng et al. [14] use only source sketches and adjust a B-spline for control point manipulation. Our approach is most similar to methods that use sketches for defining both source and destination points [12,13].

When warping normals as in our method, a $\mathcal{C}^1$ warping field is required to avoid creating seams. Igarashi et al. [10] calculate a deformation on a mesh and linearly interpolate in the interior of the triangles. This results in a discontinuous Jacobian, that is constant inside each triangle. Fang and Hart [12] interpolate the field from the sketches using a Laplace equation, resulting in discontinuous derivatives on the sketches themselves.

Approaches based on MLS [9,14] and the method of Eitz et al. [13] seem to result in smooth warping fields. However, they provide little control and smoothness when trying to define a region of interest (ROI) for deformation. In our sketch-based interface anything outside the ROI defined by the user is left unchanged. Our system builds a warping field which is $\mathcal{C}^1$ even on the desired part of the border of the ROI. Previous methods would be appropriate for warping normals only on global and full-object deformations, while our approach builds upon RBF-based generalized interpolation [15–17] to ensure both the prescribed sparse constraints and the smoothness in the ROI without the need of additional parameters nor singular kernels [9]. Using the Bilaplacian operator would allow imposing position and derivative constraints [18] as in our method. However, we are unaware of the possibility of restricting derivatives at some, but not all, points. This is possible with our method and very important for ROI deformation.

Working with a ROI paves the way for image-based object editing. Barret and Cheney [19] used local warpings for object editing. They provide a complete tool including segmentation methods to define objects and inpainting methods to fill disclosed background. Oh et al. [20] use image layers with per-pixel depth as objects. Depth is used for perspective transformations and texture corrections. Their methods support limited relighting, because they pay no attention to existing light sources.

2.2. Light detection and relighting

Instead of depth, we work with normals because they are easier to acquire. They can be accurately obtained for both diffuse [4] and specular [21] materials using multiple images. Additionally, normals can also be obtained from single images [5,7,8].

Though these methods are able to recover scene geometry, they often do not recover reflectance. Fortunately, there are techniques to factor images into a product of reflectance and illumination, together referred as intrinsic images. This separation has long been studied [22] and there have been many recent advances using multiple images [23], machine learning [24], bilateral filtering [20], or user-assisted approaches [6,7].

While intrinsic image separation together with shape-from-shading methods are very useful, they are limited in two aspects we must take into account for re-illumination: lighting and reflectance. We present a semi-automatic method for detecting light and reflectance because it is hard to achieve nice results with the fully-automatic previous approaches. Schoeneman et al. [25] designed a method to obtain lighting. They start with a full 3D scene and find intensities for a fixed set of lights taking into account any lighting phenomena. Ikeuchi and Sato [26] find a directional light starting with a range scan and a single image. Our method is similar to the latter, since only normals are required under the directional light hypothesis.

The second limitation is reflectance. The detected color provides a reasonable diffuse albedo, but it fails to provide more complex material properties, including specular properties. Under uncontrolled lighting, there are methods that acquire parametric BRDF from images [27]. They can work with multiple [28] or single [29,30,26] images, usually iteratively updating parameters according to observed rendered differences. Most single images use many iterations and thresholds to separate Lambertian from specular regions. Our method is non-iterative, since the user marks these different regions with sketches. Estimation methods also vary according to the material model used, from higher dimensional BRDFs [29] to the Torrance–Sparrow [26] or Phong models [31]. We follow approaches based on the Phong parametric model.

The first works with RGBN images focused on changing color only. In Textureshop [32] the authors recovered normals from photographs, interactively tuning them and then developed texture synthesis on the resulting RGBN images. The normals are used to guide local distortions in the synthesized texture. All their results were produced under a single light hypothesis, as in our method. In addition to color synthesis, normal synthesis methods have also been presented [33]. Toler-Franklin et al. [3] have shown that many NPR rendering algorithms work with RGBN images. To accomplish this task, they developed signal processing techniques for normals, like filtering and derivatives estimation. Pereira and Velho [34] extended the filtering methods and developed methods to design features on RGBN images. They do not use warping on RGBN images, only warping new features along a curve. Loviscach [35] presented methods for combining and warping normals, but only showed results with user-provided warping equations.In contrast to these works, we apply free-form warping to the RGBN image itself.

Finally, Okabe et al. [7] present a user-assisted system for relighting from single images. They use user input to estimate an RGBN image and a light source. While in their method the lighting changes while geometry stays the same, in our method it is the geometry that changes.

3. Overview

We propose a sketch-based pipeline to deform images using more than pixel color information. In our system, we take as input three different buffers (Fig. 3a): the shaded buffer, which is an input image containing colors that are influenced by scene lighting conditions; and diffuse albedo and normal channels, which specify the object's texture and geometry. Afterwards, assuming a constant material, we use light sketches and all three buffers to obtain a local estimate of specular properties, a single directional light and an ambient illumination term (Fig. 3b) which are used to relight the final result. Additionally, warping sketches specify a region of interest (ROI) and a deformation. These are used to build the warping field (Fig. 3c) which is used to locally deform the albedo and normal buffers (Fig. 3d). To obtain the final image we relight the warped albedo and normals

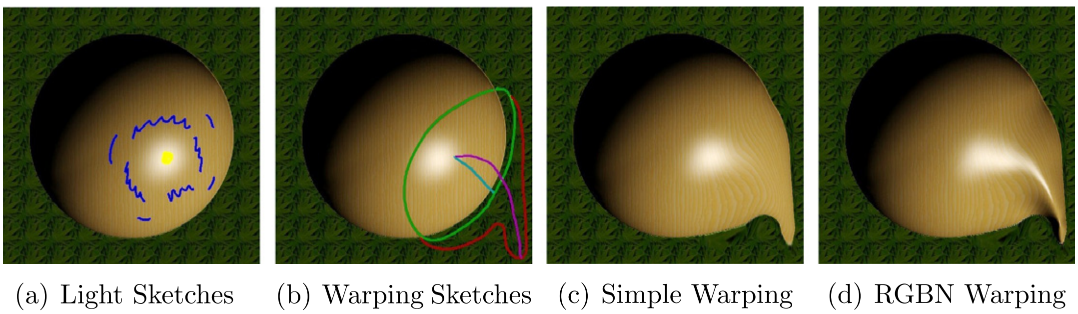


Fig. 2. Light sketches (a) mark regions of highlights and midtones. Warping sketches (b) define the free-form deformation. Relighting with warped normals reveals the shape of the object (d).

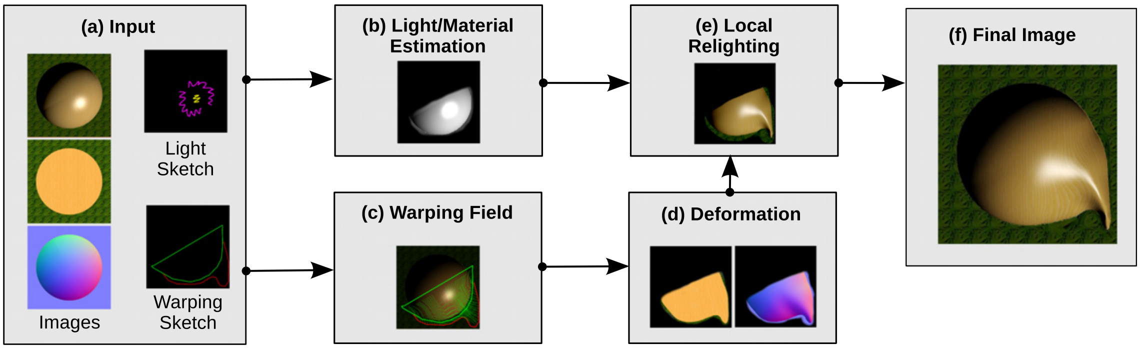


Fig. 3. Our pipeline starts with three buffers and two types of sketches (a). We use the albedo, normal and shaded buffers together with the light sketches to obtain a local estimate of material and original lighting (b). Meanwhile, we use the warping sketches (c) to build the warping field that is used to deform normals and albedo (d). Using the light and material information, we relight the deformed buffers (e). We compose the original and synthesized shaded images to obtain the final image (f).

in the ROI using the estimated light (Fig. 3e) and compose it with the rest of the original shaded image (Fig. 3f). The warping field is built using Hermite–Birkhoff radial basis functions (HBRBF) resulting in a smooth field necessary for normal manipulation.

Our method can produce local or global changes according to the input it receives. If it receives a full RGBN image, arbitrary relighting and unrestricted deformations are possible. Nevertheless, our method was designed to work under more harsh conditions. If albedo and normals are only available in the ROI, our method can deform them locally, keeping the rest of the image unchanged. In this case, we are restricted to the original lighting conditions to avoid dissonances with the unchanged part. Since working locally only brings additional complications, we focus our presentation on this scenario. In the next sections, we provide details of each step of our pipeline.

4. Light and material estimation

Even though ambiguities are present in most vision problems, humans have little difficulty in inferring what effects are caused by material or light variations over an object [36]. Therefore, in our method, the user helps the system estimate the local light and material. The light sketches are used to label different regions (Fig. 2). The first stroke marks a highlight region, a region where specular reflection dominates. The second stroke marks a midtone region, a region that faces the light source but, where specular reflection is very low. While these sketches may not be in the ROI, it is important that they represent the local lighting and material for the deformation. The exact pixels under the strokes are used as constraints for estimation.

We begin by detecting lighting and material specularity. Our two-step solution is similar to that of Ikeuchi and Sato [26]. First, our system detects lighting. Second, it detects material specularity parameters. The method consists of a series of linear equations, one for each pixel in the regions previously specified by the user. We calculate least square solutions to these two over-constrained systems. All equations are derived from the following simplified version of the Blinn–Phong shading model using a single light source: $\mathbf{i} = \rho\mathbf{a} + \rho i(\mathbf{d} \cdot \mathbf{n}) + \mathbf{k}_s \; i(\mathbf{h} \cdot \mathbf{n})^{\alpha}$. (Regarding notation, bold letters denote vectors and the $\rho\mathbf{a}$ product denotes a component wise product.)

The input images already provide the diffuse albedo ρ, normal vectors $\mathbf{n}$ and the shaded image $\mathbf{i}$. While the half-vector $\mathbf{h}$ can be computed from the normals, other parameters remain to be found. First, we determine the single light direction $\mathbf{d}$ and intensity i. We also assume that global illumination can be modeled well by a constant ambient light term $\mathbf{a}$. These hypotheses are plausible because we are only editing a small region or a single object.

The above equation should hold for each pixel and each color channel. However, fitting it over an image is already a non-linear problem that is made harder due to shadows. For this reason, we ask the user to provide two sketches marking highlights and midtones. This way, the problem can be made linear in each individual region.

We start by estimating a single light direction and intensity. To this purpose, we work on the midtone region, where specular contributions are null. We estimate the ambient term, in addition to the light, extending the approach of Ikeuchi and Sato [26]. Briefly, we use least squares to solve the following linear system in six variables $(\mathbf{a}, \mathbf{l} = i\mathbf{d})$: $\rho_{k,c}a_c + \rho_{k,c}i\mathbf{d} \cdot \mathbf{n}_k = I_{k,c}$, where c indexes the color channel and k indexes all pixels in the midtones region.

A restriction to the above equation is that the components of the ambient term must be kept positive. While we could have used a constrained least squares method, we did not find that necessary since in most of our tests the optimal ambient term already comes positive or is only slightly negative.

Having found the light, we now turn to detecting the specular coefficient k_s and the specular exponent α. We use a least squares solution to the log-linearized equations: $\log k_s + \alpha \log(\mathbf{h}_k \cdot \mathbf{n}_k) = \log S_{k,c}$, where $S_{k,c} = (I_{k,c} - \rho_{k,c}a_c - \rho_{k,c}i(\mathbf{d} \cdot \mathbf{n}_k))/i$.

Our approach is similar to that of Ukida et al. [31], but we only have a single image. Hence, we assume the material is spatially constant obtaining enough equations. In addition, we assume that the specular albedo k_s does not depend on c. Detecting the highlight color would also be possible with a slightly different linear system because there are many more equations than variables.Although we have not experimented with this approach, this change would be important for some materials.

Since highlights are usually regions of high luminance, they tend to saturate pixel values. We ignore the pixels that are possibly saturated, i.e., pixels with the maximum value of 255.

5. Warping field

Since we warp normals, our warping field must be $\mathcal{C}^1$, i.e., its Jacobian must be continuous. In this section, we describe a sketch-based method with ROI control. Our method builds an inverse warping field, i.e., a field **W** that maps a target region Λ to a source region Ω. Everything outside $\Omega \cup \Lambda$ is unchanged (a discussion of the uncovered region $\Omega - \Lambda$ occurs in Section 7). We introduce derivative constraints to produce a smooth field not only in the interior but also in the border of the region of interest. Any other method that produces $\mathcal{C}^1$ warping fields can be used instead.

5.1. User sketches

The user begins by drawing the boundary Ω^b of the region of interest (Fig. 4). Next, he sketches a curve L specifying the deformation. Most often, the curve L is open, defining a new object silhouette. However, L can also be closed; this results in regions that may be disjoint. In the former case, L must start and end on Ω^b (Fig. 4a), inducing a partition of Ω^b in two curve segments Ω_1^b and Ω_2^b (Fig. 4b). The border of the destination region Λ is defined by joining L and Ω_1^b (Fig. 4c).

To increase control, the user can provide skeleton curves Ω^s and Λ^s to constrain the deformation (Fig. 5a). If the skeleton curves touch the boundaries (Fig. 5c), they induce a partition of Ω^b (analogously Λ^b) in two parts Ω_+^b and Ω_-^b (analogously Λ_+^b and Λ_-^b). Alternatively, if skeletons are not provided, the user can directly partition Ω^b by marking splitting points on the boundary. This partition helps the user influence the sketch correspondence.

To create warping samples, matching proceeds by pairing uniform samplings of corresponding curve segments, e.g., Ω_+^b with Λ_+^b. We found that this simple scheme provides a good compromise between the number of curves and field control; however, more automatic sketch-correspondence methods could be used [13,37].

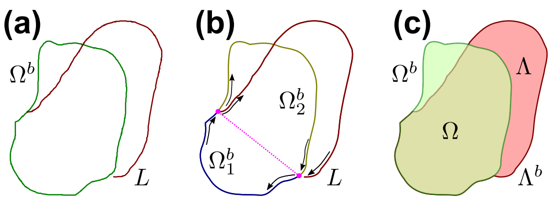


Fig. 4. Processing of the input curves when L is open: (a) user input, (b) We project the ends of L on Ω^b and break it in two parts Ω_1^b and Ω_2^f and (c) we connect L with Ω_1^f resulting in the region Λ.

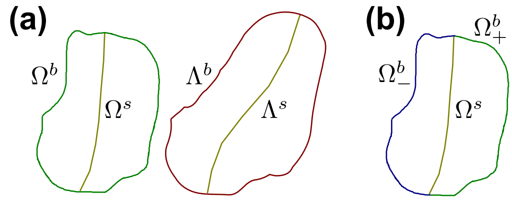


Fig. 5. In addition to contours the user can sketch skeleton lines (a). If these curves touch the boundary they induce a partition (b) of Ω^b (the same thing holds for Λ^b) in two parts Ω_+^b and Ω_-^b.

In Fig. 6, we show the effect of using the skeleton lines and partitioning the original sketches. Boundary and skeleton sketches provide good control. For example, the user can select between rotation-like and shear-like fields (Fig. 7).

When connecting L to the correct Ω_i^b, care must be taken to guarantee proper orientation. We induce an orientation on the curve L by connecting its end points and using clockwise orientation on the resulting polygon. Next, L is connected to the Ω_i^b resulting in an oriented boundary Λ^b of the target region Λ, say Ω_1^b.

All warping sketch curves are preprocessed by supersampling such that the maximum distance between consecutive samples is half a pixel. Afterwards, we apply resolution reduction four times [38]. This filter is used to get the overall shape of the curve and reduce noise from the input method.

We build a displacement field by sparse interpolation of sketch samples. Our samples are vectors starting in Λ and ending in Ω (Fig. 4c). In particular, we start with samples that map boundary to boundary and skeleton to skeleton. Conceptually, we define a displacement field in the entire image, by restricting the interpolated field to Λ and setting zero displacements outside of Λ. Any $\mathcal{C}^1$ warping field in Λ that interpolates these constraints will result in a continuous warp in the entire image, except on the new silhouette L. Notice that continuity on Ω_1^b is a consequence of the constraints that fix it in place.

However, a continuous field is not enough, because a non-smooth warp will result in discontinuous normals (Fig. 8). While many interpolation methods can be used to define $\mathcal{C}^1$ fields inside Λ, we impose derivative constraints on the samples of Ω_1^b that result in continuity of the derivative there. These more advanced constraints are satisfied using Hermite–Birkhoff RBF interpolation. In theory, since derivatives are only imposed on the samples, our field may not be smooth between the samples of Ω_1^b. Nevertheless, in practice, we have observed that the resulting field is sufficiently smooth for warping normals.

5.2. Hermite–Birkhoff RBF warping field

In our system, we used a warping field computed by Hermite–Birkhoff interpolation based on radial basis functions [16]. Our constraints prescribe displacements at given points and, at some of them, they also enforce $\mathcal{C}^1$-continuity of a restricted warping field.

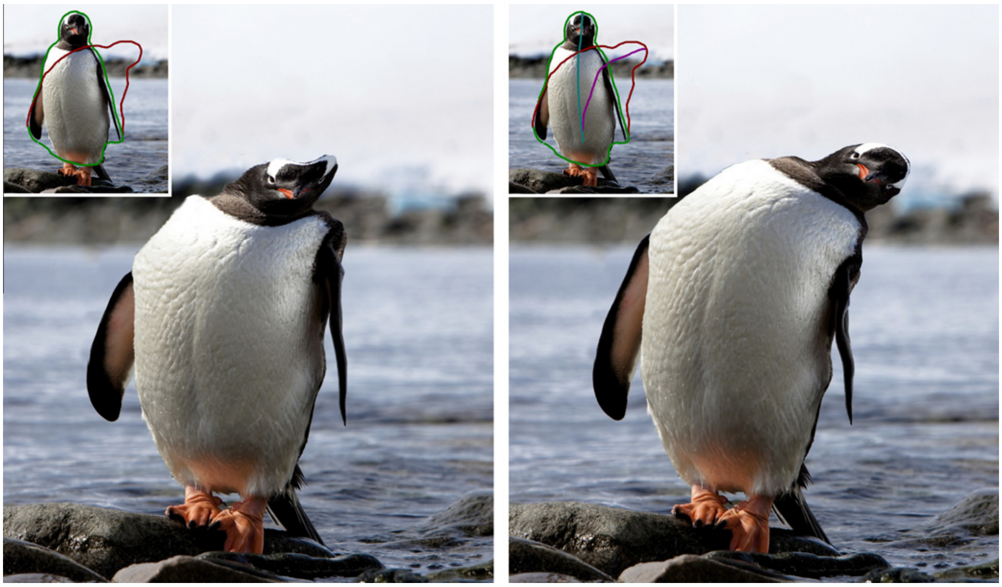
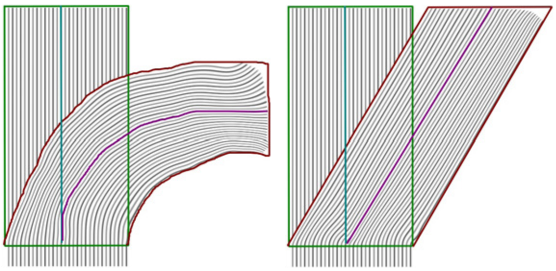


Fig. 6. On the left, arc-length correspondence failed, resulting in extreme compression in the head; on the right, the partition induced by the skeleton curves led to a proper correspondence and warping.



(a) Rotation (b) Shear

Fig. 7. We can sketch rotation-like or shear-like fields.

Formally, we are looking for a displacement field $\mathbf{F}:\mathbb{R}^2\to\mathbb{R}^2$ such that $\mathbf{F}(\mathbf{y}^j)=\mathbf{c}^j$, $\mathbf{F}(\mathbf{x}^i)=\mathbf{0}$, and $D\mathbf{F}(\mathbf{x}^i)=\mathbf{0}$, where $\mathbf{y}^j\in L\cup\Lambda^s$, $\mathbf{x}^i\in\Omega_1^b$, $\mathbf{c}^j\in\Omega^b\cup\Omega^s$. The restrictions on the Jacobian $D\mathbf{F}$ at the boundary points $\mathbf{x}^i$ enforce $\mathcal{C}^1$-continuity of the interpolated field at the boundary of the region of interest, outside of which the field is supposed to be zero. Notice that the sets $\mathbf{x}^i$ and $\mathbf{y}^j$ are disjoint.

These constraints arise naturally in our context and lead to an instance of (multivariate and unstructured) Hermite–Birkhoff interpolation, which, unlike Hermite interpolation, does not require all derivative information at every sample.

This generalized interpolation problem can be computationally solved by means of radial basis functions techniques. First, notice that these constraints do not require coupling between the component fields $F_1,\ F_2:\mathbb{R}^2\to\mathbb{R}^2$ which define $\mathbf{F}=(F_1,F_2)$. This reduces the vector problem to two scalar problems defined by $F_k(\mathbf{y}^j)=c_{j,k}$, $F_k(\mathbf{x}^i)=d_{i,k}$ and $\nabla F_k(\mathbf{x}^i)=\mathbf{0}$, $k=1,2$. Employing the generalized interpolation framework [15–17], we deduce appropriate forms of linearly-augmented RBF-based interpolants for both F_1 and F_2:

$$F_k(\mathbf{x})=\sum_i\{\alpha_i\psi(\mathbf{x}-\mathbf{x}^i)-\langle\boldsymbol{\beta}^i,\nabla\psi(\mathbf{x}-\mathbf{x}^i)\rangle\}+\sum_j\gamma_j\psi(\mathbf{x}-\mathbf{y}^j)+\langle\mathbf{a},\mathbf{x}\rangle+b$$

where $\psi(\mathbf{x})=\phi(\|\mathbf{x}\|)$ for a suitable radial basis function $\phi:\mathbb{R}_+\to\mathbb{R}$ and $\alpha_i,\ \gamma_j,\ b\in\mathbb{R}$ and $\boldsymbol{\beta}^i,\ \mathbf{a}\in\mathbb{R}^2$ are the fitting

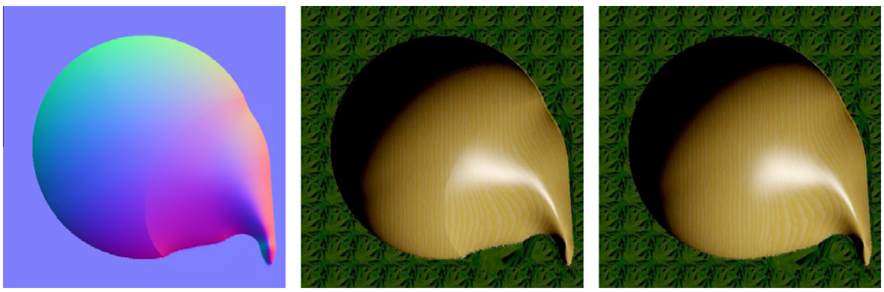


Fig. 8. If the Jacobian is not restricted on the border, the field is not smooth, resulting in discontinuous normals (left) and shading (middle). On the right, a continuous derivative was used.

coefficients uniquely determined by the aforementioned interpolation constraints along with the additional side-conditions:

$$\sum_i \{\alpha_i \mathbf{x}^i + \boldsymbol{\beta}^i\} + \sum_j \gamma_j \mathbf{y}^j = \mathbf{0}$$
$$\sum_i \alpha_i + \sum_j \gamma_j = 0.$$

Thus, after choosing a suitable ϕ, the fitting coefficients can be computed by solving two symmetric indefinite linear systems which only differ in their right-hand sides. Our implementation employs an LDL^T factorization of the system matrix which is used to fit both F_k. We found that LAPACK's `xSYSV` LDL^T routines provide a better balance in performance/memory/stability requirements than the alternative `xGESV` and `xSPSV` (also from LAPACK), which implement a general *LU* decomposition and a packed LDL^T-factorization respectively.

There are several alternatives for choosing a suitable basis function ϕ. Since we are interested in warping fields at least $\mathcal{C}^1$ to correctly propagate normals and to approximate the smoothness condition on the boundary of the region of interest, we seek a ϕ inducing a ψ at least $\mathcal{C}^2$, since the interpolant contains first-order differentials of ψ. We employed the globally-supported basis function $\phi(r) = r^3$, which has interesting variational properties, as studied by Duchon in his seminal paper [39]. Consequently, our method has no additional interpolation parameters.

It is noteworthy that, due to the use of Duchon's basis function, we need the polynomial part in the interpolant to ensure solvability of the resulting linear system. Although we could drop this affine term if we used Wendland's functions [16], we found that Duchon's provides better warping fields without noticeable degradation in performance due to the additional polynomial term. Moreover, although Wendland's functions are compactly-supported, the resulting system would still be dense because the associated radius parameter would need to be large due to the sparsity of our samples.

Our vector field interpolation reconstructs a displacement field $\mathbf{F}$. From it, we obtain a warping $\mathbf{W}:\Lambda \to \Omega$, $\mathbf{W}(\mathbf{x}) = \mathbf{x} + \mathbf{F}(\mathbf{x})$ which is used for deformation in the next section.

6. Deformation

Having built the warping field $\mathbf{W}$, we can use it to transport colors and normals. First, we reconstruct these attributes with bilinear interpolation. Transporting color amounts to using the field $\mathbf{W}$ to reference a point in Ω. This simple solution fails for warping normals. If we relight the final image with copied normals, we will obtain traditional color warping results assuming there are no estimation errors. Normals have to rotate, twist and stretch according to the warping field. If we had a 3D warping function, we could transform the normals by the transpose of the 3D Jacobian. However, all we have is a 2D warp. Our solution is to look at normal vectors as 2D gradients and to warp gradients following the work of Pereira and Velho [34].

Given a unit normal $\mathbf{n} = (n_1, n_2, n_3)$, we first convert it to a gradient using $-n_1/n_3 = z_x$ and $-n_2/n_3 = z_y$. Next, we warp the gradients. Given a gradient field $\mathbf{g}$, define $\mathbf{g}^* : \Lambda \to \mathbb{R}^2$, $\mathbf{g}^* = \mathbf{g}(\mathbf{W}(\mathbf{x})) \cdot D\mathbf{W}(\mathbf{x})$. We say $\mathbf{g}^*$ is the field $\mathbf{g}$ warped by $\mathbf{W}$. In other words, transfer the gradient vector with the warping field and multiply it by the 2D warping Jacobian. This gradient warping rule is equivalent to transferring the heights directly because of the chain rule: $\mathbf{g}^* = \mathbf{g} \cdot D\mathbf{W} = \nabla z \cdot D\mathbf{W} = \nabla(z \circ \mathbf{W})$. Finally, we obtain the normal back using:

$$\mathbf{n}(x,y) = \frac{(-z_x, -z_y, 1)}{\sqrt{z_x^2 + z_y^2 + 1}}$$

Computationally, we approximate the Jacobian by finite differences in Λ. This was done for simplicity only, as we could also have calculated it analytically from the field expression [40]. In short, to warp normals we first convert them to gradients, warp the gradients, and finally convert them back to normals.

7. Local relighting and composing

After estimating lighting and material, and obtaining the new albedo and geometry represented by normals, we have all we need to relight the final result. Global relighting is straightforward. However, it is important to discuss the case of local manipulation, which results in changes only in the region Λ while the rest of the image is kept unchanged. A trivial solution would be to calculate the new color of each pixel in Λ under the detected lighting. However, this solution is too dependent on accurate estimates and is not robust to errors. For instance, unless we detected precisely the original conditions, we would create noticeable seams.

During relighting, we calculate shading for two images: the warped and the original image. The reason we relight the original image is to calculate a residue. This residue consists of the colors that are not predicted by our shading model and estimates (Fig. 9). It is the difference between the original image and the image estimated by our model. We proceed with warping this residue simply by transporting colors. The warped residue image is added to the final result. In this way, we guarantee that no seams will be created due to lighting differences (Fig. 10).

The goal of all previous estimation steps is to produce a residue that is as close as possible to zero. Nevertheless, besides lighting and material estimate errors, reaching zero may not be possible because of global illumination effects. Therefore, this procedure is essential.

Regarding the final composition, two different cases must be considered to guarantee a seamless result in the border of Λ. In the first case, since points on Ω_1^b are not allowed to move, the color transition is already continuous. In the second case, points on Ω_2^b do move. However, these points usually lie on the border of an object, where the original image was already discontinuous. When the points are not on an object border, blending methods should be used. For these cases, we output an alpha mask with the known seams based on the user's sketch lines.This mask allows manual or automatic blending to be performed.In our

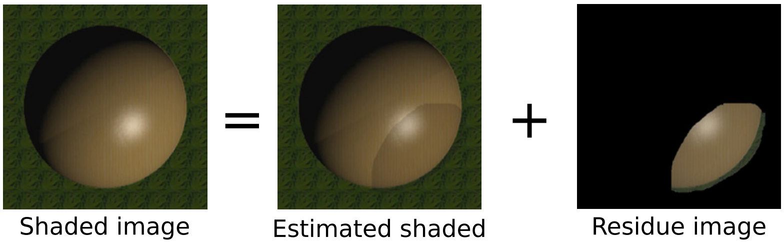
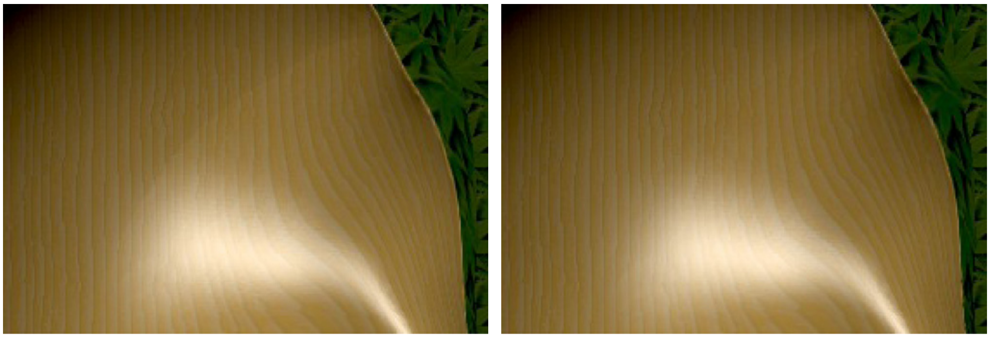


Fig. 9. The shaded image is decomposed into an estimated shaded and a residue image.

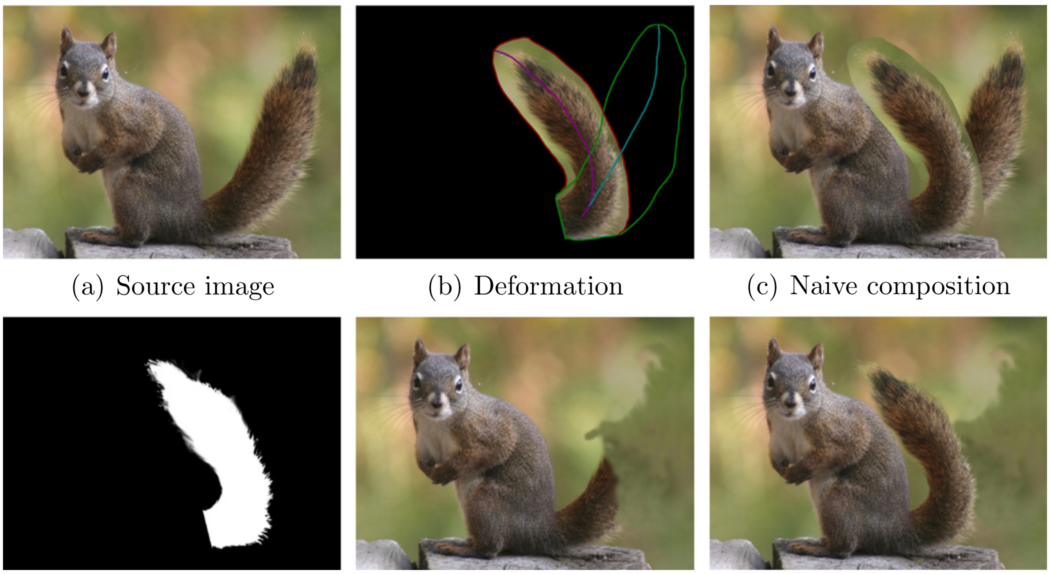


(a) No residue (b) Residue correction

Fig. 10. The residue is recombined after warping and relighting. It makes the method robust to light detection errors, avoiding artifacts (left).

examples, there are small artifacts as a result of using a mask derived from the sketches, as can be seen by zooming in. One exception is Fig. 11, where we show how a more refined alpha mask can be provided by the user.

The user may decide to move or shrink the object which results in disclosure of an unknown background. We find this region using set difference: $\Gamma = \Omega - \Lambda$. While in all results in this work we have filled Γ manually, automatic



(a) Source image (b) Deformation (c) Naive composition

(d) User-provided mask (e) Manual inpainting (f) Final

Fig. 11. Steps of the final composition. The original image (a) could be composed directly with a naive alpha channel (1 inside Λ, 0 outside) as in (b and c). For improved results the user can use a more sophisticated alpha channel (d) and inpainting (e) to obtain the final composition (f).

Table 1
The execution time of our system is dominated by the evaluation of the field. Column samples denotes the total samples used, while column Hermite denotes how many of these had derivative restrictions. All the results were generated on a single core of a 2.66 GHz Intel Xeon.

Image	# Samples	# Hermite	# Pixels	Time (s)
Box (Fig. 15)	46	24	31k	3
Reservoir (Fig. 15)	33	0	76k	4
Sphere (Fig. 8)	51	0	53k	4
Sphere (Fig. 2)	51	26	53k	7
Helmet (step 1) (Fig. 1)	68	40	101k	19
Helmet (step 2) (Fig. 1)	70	42	107k	21
Soldier (Fig. 14)	81	23	117k	24
Ruins (Fig. 20)	87	29	124k	25
Helmet (step 3) (Fig. 1)	87	57	147k	33
Zebra(Fig. 12)	91	0	141k	35
Hammer (Fig. 16)	130	19	169k	37
Squirrel(Fig. 11)	133	33	282k	58
Pine (Fig. 19)	155	0	340k	70
Zebra (Fig. 12)	91	74	141k	74
Penguin (Fig. 6)	110	70	314k	80
Penguin (Fig. 6)	136	70	314k	86
Shell (Fig. 13)	155	0	556k	112

inpainting techniques can be used instead [41]. In Fig. 11, we show an example, where composition was performed with manual inpainting and a commercial image manipulation program to generate the alpha mask.

8. Results and discussion

Having presented the full method, we now discuss some results. We made experiments with albedo and normals from different sources like photometric stereo datasets (hammer, shell, pine, helmet, and soldier), synthetic models (sphere), and single photographs (ruins and box). We have also made experiments with color only deformation (zebra, penguin, and reservoir).

Even though the fitting stage of the interpolation involves solving a linear system, it is very fast because the number of variables is of the order of the number of samples. For all examples, the execution time of the system is dominated by the HBRBF field evaluation, usually taking a few seconds (Table 1), because that the RBF field must be evaluated per-pixel. The resulting complexity is $O(MN)$, where N is the number of pixels and M is the number of samples. Better performance could be obtained by only evaluating the field in a coarser grid [9] and using cubic interpolation for preserving smoothness. In addition, warping time is larger when using derivative constraints (Table 1) because these constraints require a slightly more complex interpolant that takes longer to evaluate. Light and material estimation are very fast steps involving only the solution of linear systems of up to six variables. In addition, the normal deformation and subsequent relighting operations are also fast.

The skeleton curves offer the user additional information about the geometry of the warping, as illustrated in Fig. 6. When these curves are used, the internal samples guide the warping towards local rotations, and also the induced partition improves sketch correspondence. In Fig. 15, we can observe the same technique used on a reservoir engineering illustration. In this case, content within a ROI is exaggerated by distortion as guided by the

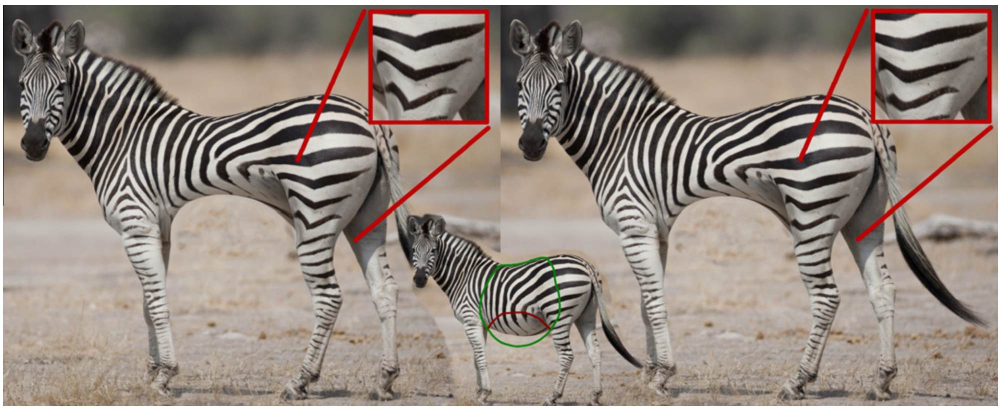


Fig. 12. Smoothness in the border of the ROI is not useful only for normal warping. Without derivative constraints the stripes bend abruptly (left). We generate smooth color transitions (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

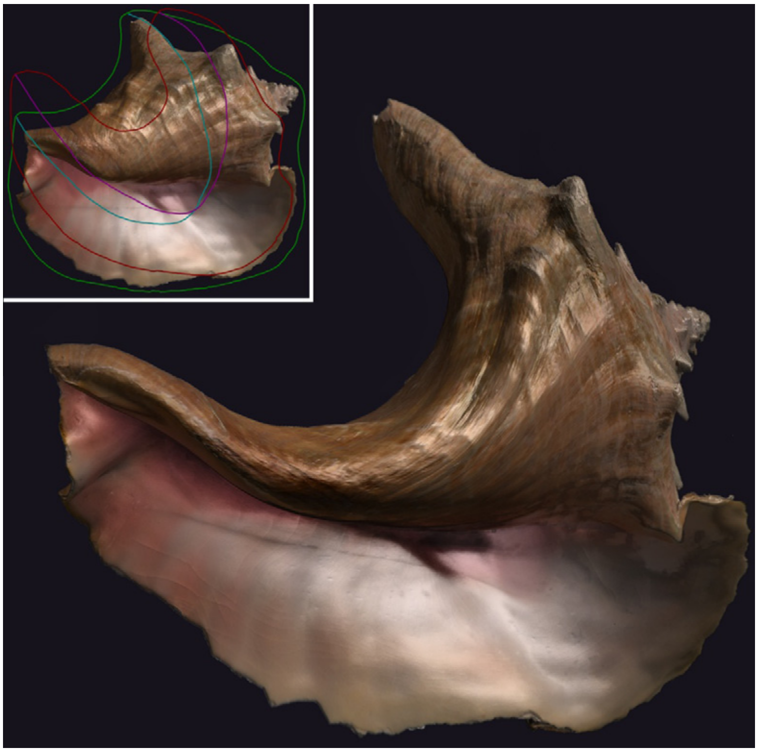


Fig. 13. Full object deformations can also be achieved by specifying a new silhouette which is a closed curve.

skeleton curves. This creates emphasis for improving the visualization and it is inspired by traditional illustration techniques [42]. Fig. 12 shows how smoothness in the ROI is essential for color warping. The zebra's stripes break abruptly if derivative constraints are not used.

The perfect scenario for our pipeline is when we have complete and precise information about albedo and normals of the scene. As expected, the photometric stereo datasets give our more expressive results. Comparing Fig. 16a and b, we observe the correctness of shadows and highlights. The same comparison can be done in Figs. 1, 2, and 14, where we can see the importance of highlights and shadows to communicate volume. Our system correctly generates shadows in regions that are not facing

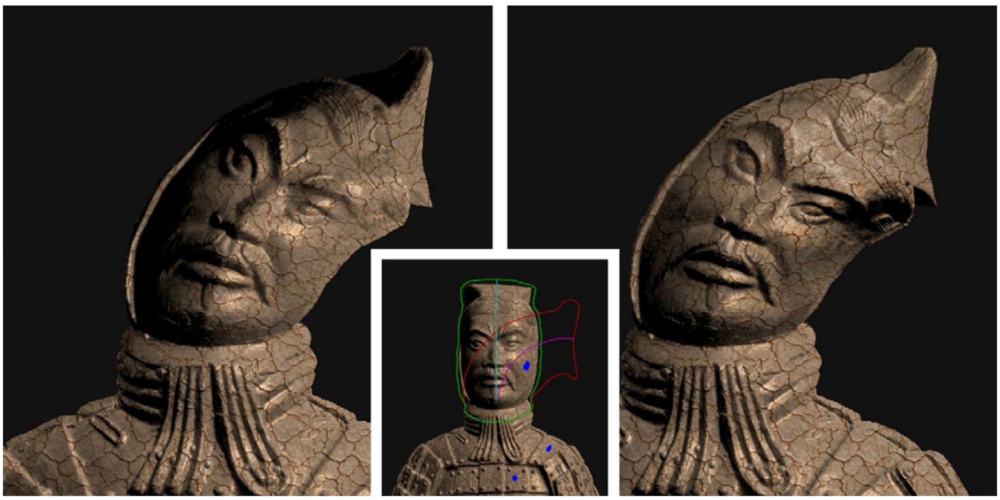


Fig. 14. Results of deformation using our system in a Chinese statue: the result without relighting (left), with relighting and residue (right), the original shaded image together with user sketches (bottom).

the light. However, since it only uses local information, it cannot generate cast shadows. In Figs. 13 and 19, we show full object deformation using closed silhouette curves.

In Fig. 20, our input consists of a single photograph of ruins. Neither albedo nor normals are available. However, to warp the tower the user only had to inform these locally. The albedo was found by adjusting the illumination manually in an image editing tool. This example shows a scenario, where it is easy for the user to directly inform the RGBN image. While the original image only had two levels

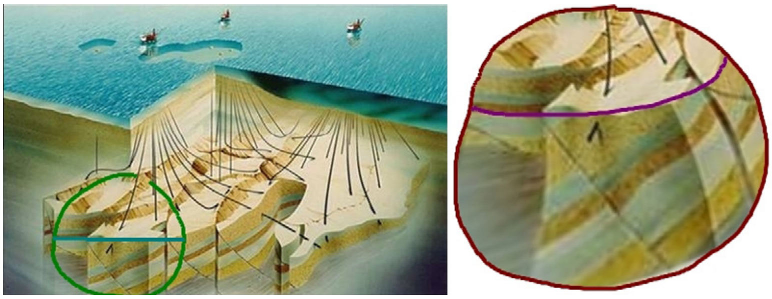
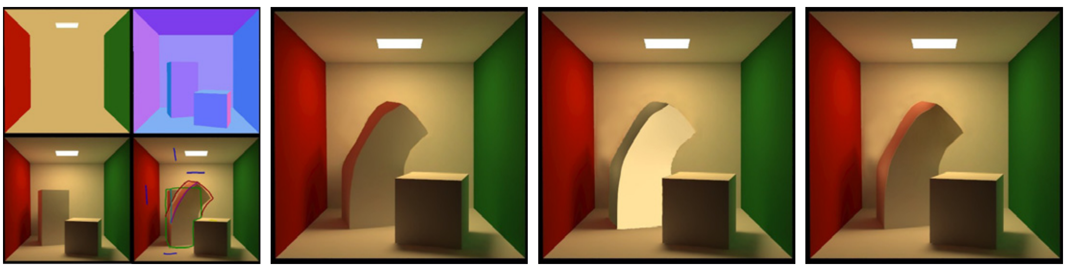


Fig. 15. Curvature of reservoir layers is greatly exaggerated to bring emphasis in the visualization (photo: Statoil).

(a) No relighting (b) Relighting

Fig. 16. Simple color warping results in wrong shadows. This lighting pattern would require a light in the middle of the image, which is not the original lighting. With relighting, we recreate the proper shadow. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Input (b) No correction (c) Simple relighting (d) Relighting with residue

Fig. 17. The detected light was stronger than it should have been. Recombining the residue reduces detection errors and even captures some color bleeding effects. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

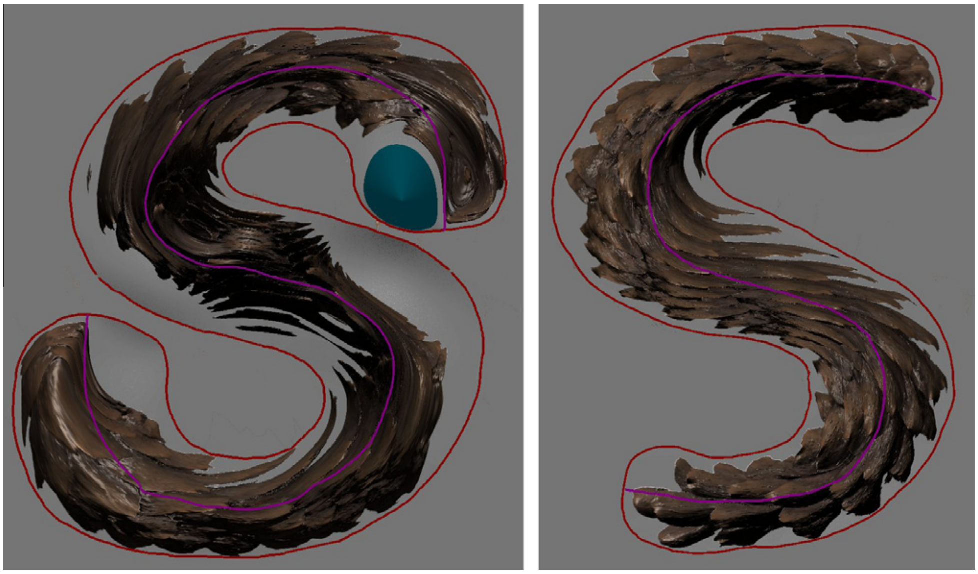


Fig. 18. The first sketch has close sheets which results in close samples with very different displacements. To satisfy these conflicting constraints, the interpolation introduces distortion. The second sketch presents an improved result but still with too much shear. The dark cyan pixels represent points from outside the original image. Original image and sketches are shown in Fig. 19. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of shading, after warping, there are many different possible normals and shading values. Our method generates a result that presents all these variations. In all our examples, lighting and specular albedo were detected automatically from the sketches, except for this image because of the lack of normal variation.

Fig. 19. Full object deformation.

If scene geometry is simple enough, the user can manually specify normals and albedo in the whole scene (Fig. 17). In addition, this experiment shows the importance of the residue for correcting light detection errors and it even includes global illumination effects in our final composition, such as color bleeding.

In short, the results show many different scenarios concerning the effect of relighting after an image deformation. There are scenarios of large contrasts, where relighting makes a big difference. This is the case for the hammer (Fig. 16). Frequently, shading and highlights are the only shape cues present in the interior of an object. While contours may provide the overall scene and object information, they may not inform the actual shape (Fig. 2). In addition, there are cases, where the changes made possible by relighting are subtle. Even so, we believe the correct details produced by our method are important (Fig. 1). Finally, we understand relighting is not always necessary after an image deformation. This is specially true for less dramatic lighting conditions (Fig. 12).

9. Conclusion

Our contribution in this work is a sketch-based method to deform images that does more than simply transfer colors. Our method generates results that appear to be photometrically consistent. By relighting the warped image, we can generate new highlights, shading and shadows, where necessary. To support relighting, we have developed a method to seamlessly deform RGBN images. We have shown examples of our method working both locally in

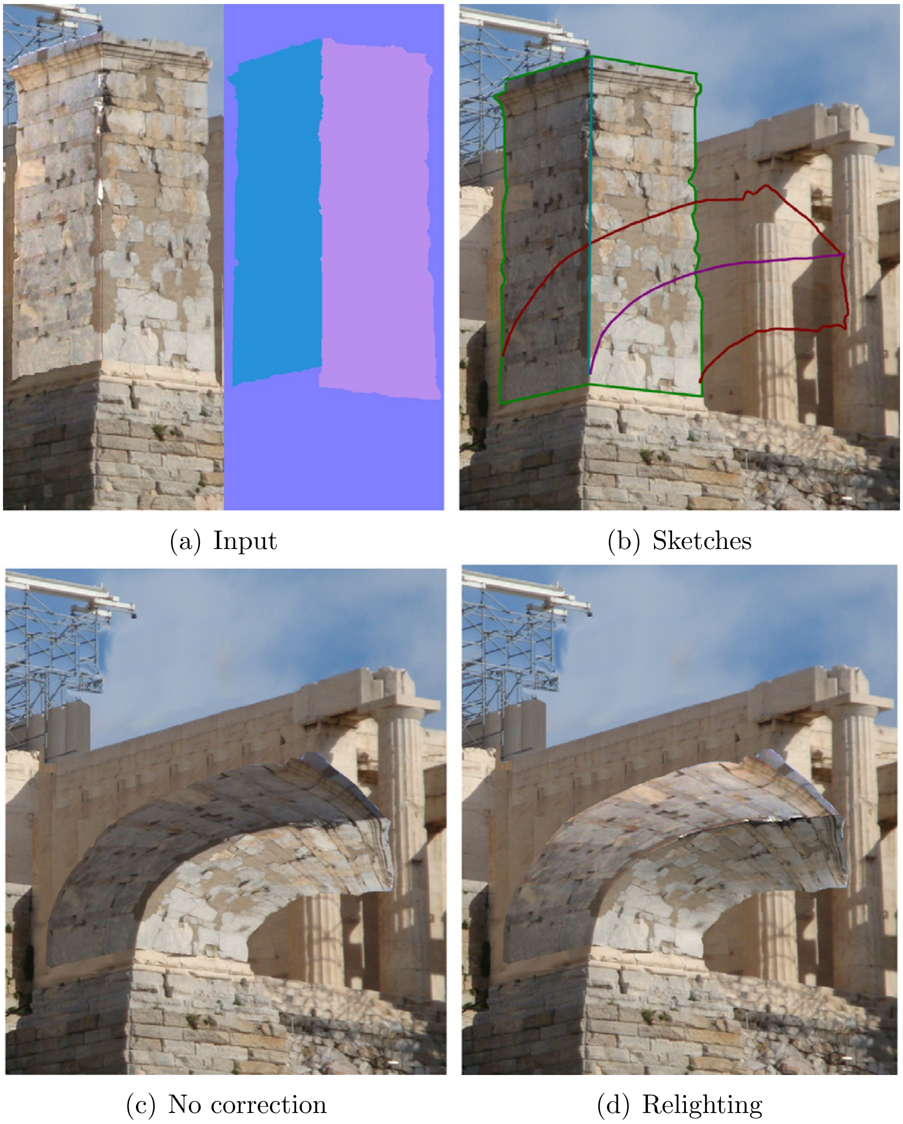


Fig. 20. Only deformations on the tower are possible, since the user did not provide background normals.

single photographs and globally with photometric stereo RGBN images.

The main limitation of our method is its assumption of a local shading model. For this reason, it cannot warp global illumination effects like shadows and inter-reflections correctly (Fig. 17). Secondly, since we use the Jacobian of the warping, it is desirable to build warping fields that have derivatives of guaranteed full rank. However, ensuring rank is related to avoiding fold-overs, known to be a hard problem. In addition, the field interpolation result is very dependent on the field samples. If the arc-length correspondence fails to capture user intent, it will result in a distorted warping field. Additionally, since we first build a global interpolant and later restrict it to the ROI, our method suffers with close sheets (Fig. 18). This problem could be avoided by the user if he creates the final deformation using more than one step, as we show in Fig. 1. Finally, RGBN images have less information than a full 3D model, as a result, our method only handles planar deformations, depth changes are not possible.

In future work, we intend to use our method in other applications, including warping of vector fields other than gradients. Moreover, by further exploiting the Hermite–Birkhoff interpolation theory with RBFs, more elaborate derivative restrictions can be used for much more than boundary restrictions. For instance, they could be used to specify local rotations and stretches, thus coupling the scalar fields which define the warping, especially if integrated in a friendly user interface. Finally, we would like to extend our method to image-based animation systems having as input either photographs [43] or drawings from cel animation pipelines as in Lumo [8]. The changes in shading that our method allows would provide strong cues for shape and motion.

Acknowledgments

We thank Szymon Rusinkiewicz and the Princeton Graphics Group for the RGBN dataset used in this paper. We also thank Paul Debevec and ICT's Graphics Lab for making the helmet dataset available. We also thank the anonymous reviewers for their careful and valuable comments and suggestions. This research was supported in part by the iCORE/Foundation CMG Industrial Research Chair in Scalable Reservoir Visualization, by Discovery Grants Program from the Natural Sciences and Engineering Research Council of Canada, and grants from the Brazilian funding agencies CNPq and CAPES/PDEE.

References

[1] J. Gomes, L. Darsa, B. Costa, L. Velho, Warping and Morphing of Graphical Objects, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.
[2] C. Correa, D. Silver, M. Chen, Feature aligned volume manipulation for illustration and visualization, IEEE Trans. Visual. Comput. Graph. 12 (5) (2006) 1069–1076.
[3] C. Toler-Franklin, A. Finkelstein, S. Rusinkiewicz, Illustration of complex real-world objects using images with normals, in: NPAR, 2007, pp. 111–119.
[4] R.J. Woodham, Photometric method for determining surface orientation from multiple images, in: Shape Recovery, Jones and Bartlett Publishers, Inc., USA, 1992, pp. 115–120.
[5] T.-P. Wu, J. Sun, C.-K. Tang, H.-Y. Shum, Interactive normal reconstruction from a single image, ACM Trans. Graph. 27 (5) (2008) 1–9.
[6] A. Bousseau, S. Paris, F. Durand, User-assisted intrinsic images, in: SIGGRAPH Asia '09, ACM, 2009, pp. 1–10.
[7] M. Okabe, G. Zeng, Y. Matsushita, T. Igarashi, L. Quan, H. yeung Shum, Single-view relighting with normal map painting, in: Proceedings of Pacific Graphics 2006, pp. 27–34.
[8] S.F. Johnston, Lumo: illumination for cel animation, in: Proceedings of the 2nd International Symposium on Non-Photorealistic Animation and Rendering, NPAR '02, 2002, pp. 45–52.
[9] S. Schaefer, T. McPhail, J. Warren, Image deformation using moving least squares, ACM Trans. Graph. 25 (3) (2006) 533–540.
[10] T. Igarashi, T. Moscovich, J.F. Hughes, As-rigid-as-possible shape manipulation, ACM Trans. Graph. 24 (3) (2005) 1134–1141.
[11] R.C. Zeleznik, K.P. Herndon, J.F. Hughes, Sketch: an interface for sketching 3d scenes, in: SIGGRAPH '96: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, ACM, New York, NY, USA, 1996, pp. 163–170.
[12] H. Fang, J.C. Hart, Detail preserving shape deformation in image editing, ACM Trans. Graph. 26 (3) (2007) 12.
[13] M. Eitz, O. Sorkine, M. Alexa, Sketch-based image deformation, in: Proceedings of Vision, Modeling and Visualization (VMV), 2007, pp. 135–142.
[14] Y. Weng, X. Shi, H. Bao, J. Zhang, Sketching MLS image deformations on the GPU, Comput. Graph. Forum 27 (7) (2008) 1789–1796.
[15] G. Fasshauer, Solving partial differential equations by collocation with radial basis functions, in: Surface Fitting and Multiresolution Methods, Vanderbilt University Press, 1997, pp. 131–138.
[16] H. Wendland, Scattered Data Approximation, Cambridge University Press, Cambridge, 2005.
[17] I. Macêdo, J.P. Gois, L. Velho, Hermite interpolation of implicit surfaces with radial basis functions, in: XXII SIBGRAPI, IEEE Computer Society, 2009, pp. 1–8.
[18] Y. Gingold, D. Zorin, Shading-based surface editing, in: ACM SIGGRAPH 2008 Papers, ACM, New York, NY, USA, 2008.
[19] W.A. Barrett, A.S. Cheney, Object-based image editing, in: SIGGRAPH '02, ACM, 2002, pp. 777–784.
[20] B.M. Oh, M. Chen, J. Dorsey, F. Durand, Image-based modeling and photo editing, in: SIGGRAPH '01, ACM, 2001, pp. 433–442.
[21] T. Chen, M. Goesele, H.-P. Seidel, Mesostructure from specularity, in: CVPR '06: Proceedings, IEEE Computer Society, 2006, pp. 1825–1832.
[22] H. Barrow, J. Tenenbaum, Recovering intrinsic scene characteristics from images, Comput. Vis. Syst. (1978) 3–26.
[23] Y. Weiss, Deriving intrinsic images from image sequences, IEEE Int. Conf. Comput. Vis. 2 (2001) 68–75.
[24] M.F. Tappen, W.T. Freeman, E.H. Adelson, Recovering intrinsic images from a single image, IEEE Trans. Pattern Anal. Mach. Intell. 27 (9) (2005) 1459–1472.
[25] C. Schoeneman, J. Dorsey, B. Smits, J. Arvo, D. Greenberg, Painting with light, in: SIGGRAPH '93, ACM, 1993, pp. 143–146.
[26] K. Ikeuchi, K. Sato, Determining reflectance properties of an object using range and brightness images, IEEE Trans. Pattern Anal. Mach. Intell. 13 (11) (1991) 1139–1153.
[27] T. Weyrich, J. Lawrence, H. Lensch, S. Rusinkiewicz, T. Zickler, Principles of appearance acquisition and representation, in: SIGGRAPH '08: Courses, ACM, 2008, pp. 1–119.
[28] C. Tchou, J. Stumpfel, P. Einarsson, M. Fajardo, P. Debevec, Unlighting the parthenon, in: SIGGRAPH '04: Sketches, ACM, 2004, p. 80.
[29] S. Boivin, A. Gagalowicz, Image-based rendering of diffuse, specular and glossy surfaces from a single image, in: SIGGRAPH '01, ACM, 2001, pp. 107–116.
[30] K. Hara, K. Nishino, K. Ikeuchi, Mixture of spherical distributions for single-view relighting, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2007) 25–35.
[31] H. Ukida, Y. Tanimoto, T. Sano, H. Yamamoto, 3D Object shape and reflectance property reconstruction using image scanner, in IEEE International Workshop on Imaging Systems & Techniques Proceedings (IST 2009), Shenzhen, 2009, pp. 94–99.
[32] H. Fang, J.C. Hart, Textureshop: texture synthesis as a photograph editing tool, in: SIGGRAPH '04, ACM, 2004, pp. 354–359.
[33] T. Pereira, L. Velho, Normal synthesis on RGBN images, in: GRAPP'10: Proceedings, 2010.
[34] T. Pereira, L. Velho, RGBN image editing, in: XXII SIBGRAPI: Proceedings, IEEE Computer Society, 2009, pp. 24–31.
[35] J. Loviscach, Care and feeding of normal vectors, in: Shader X6: Advanced Rendering Techniques, Charles River, 2008.
[36] W.M. Andrews, Introduction to perceptual principles in medical illustration: lines and illusions, Tutorial Notes, Illustrative Visualization for Medicine and Science (Eurographics '06), 2006.
[37] J. Zimmermann, A. Nealen, M. Alexa, Silsketch: automated sketch-based editing of surface meshes, in: SBIM '07: Proceedings of the 4th Eurographics Workshop on Sketch-Based Interfaces and Modeling, ACM, New York, NY, USA, 2007, pp. 23–30.
[38] F. Samavati, R. Bartels, Local filters of B-spline wavelets, in: Proceedings of International Workshop on Biometric Technologies, University of Calgary, 2004, pp. 105–110.
[39] J. Duchon, Splines minimizing rotation-invariant semi-norms in Sobolev spaces, Lect. Notes Math., vol. 571, Springer, Berlin, Heidelberg, 1977. <http://www.springerlink.com/content/g27671q701166031>.
[40] E. Vital Brazil, I. Macêdo, M. Costa Sousa, L. Velho, L.H. de Figueiredo, Shape and tone depiction for implicit surfaces, Comput. Graph. 35 (1) (2011) 43–53.
[41] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: SIGGRAPH '00, ACM Press, 2000, pp. 417–424.
[42] E. Lane, Karst in Florida. Florida Geological Survey, Florida Bureau of Geology, Tallahassee, 1986, Special report no. 29. <http://www.dep.state.fl.us/geology/>.
[43] A. Hornung, E. Dekkers, L. Kobbelt, Character animation from 2d pictures and 3d motion data, ACM Trans. Graph. 26 (1) (2007) 1.