AppWarp: Retargeting Measured Materials by Appearance-Space Warping

Xiaobo An[*] Xin Tong[†] Jonathan D. Denning[*] Fabio Pellacini[*‡]
[*]Dartmouth College [†]Microsoft Research Asia [‡]Sapienza University of Rome

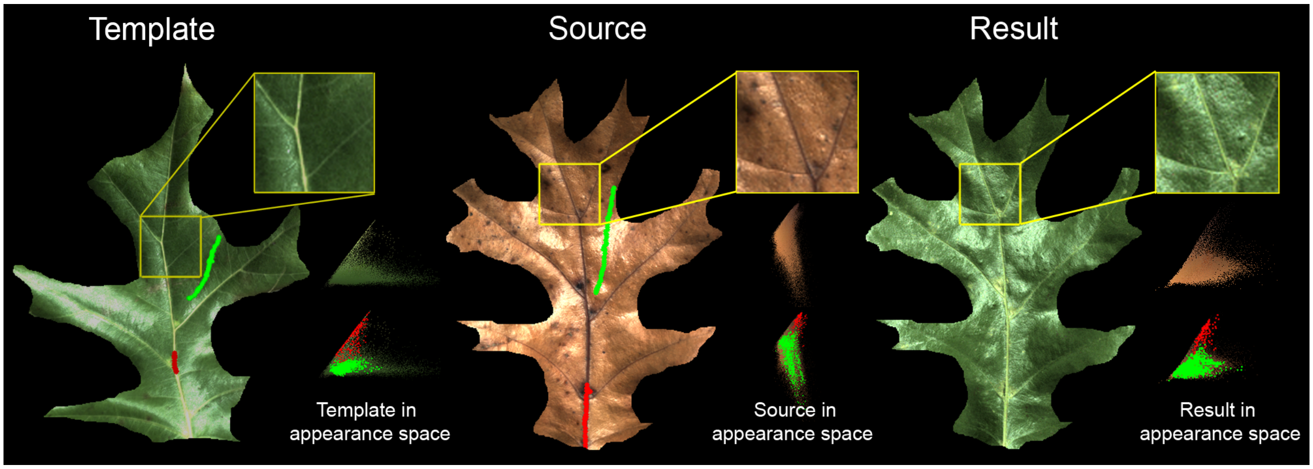
Template
Source
Result
Template in appearance space
Source in appearance space
Result in appearance space

Figure 1: *A material retargeting result, generated by our* AppWarp *algorithm, where a* source *measured material (brown leaf) is edited by applying the reflectance functions found on a* template *measured material (green leaf). To guide the retargeting process, the user marks pairs of strokes to specify corresponding regions on source and template. In this case, the user marks the stem (red strokes) and body (green strokes) on the two leaves. From these correspondences,* AppWarp *computes a retargeted material that maintains the spatial patterns of the source while exhibiting reflectance behaviors of the template. Below the zoom-in, the reflectance functions are shown projected into two dimensions with corresponding user's strokes overlaid. Data from [Lawrence 2009].*

Abstract

We propose a method for retargeting measured materials, where a source measured material is edited by applying the reflectance functions of a template measured dataset. The resulting dataset is a material that maintains the spatial patterns of the source dataset, while exhibiting the reflectance behaviors of the template. Compared to editing materials by subsequent selections and modifications, retargeting shortens the time required to achieve a desired look by directly using template data, just as color transfer does for editing images. With our method, users have to just mark corresponding regions of source and template with rough strokes, with no need for further input.

This paper introduces *AppWarp*, an algorithm that achieves retargeting as a user-constrained, appearance-space warping operation, that executes in tens of seconds. Our algorithm is independent of the measured material representation and supports retargeting of analytic and tabulated BRDFs as well as BSSRDFs. In addition, our method makes no assumption of the data distribution in appearance-space nor on the underlying correspondence between source and target. These characteristics make *AppWarp* the first general formulation for appearance retargeting. We validate our method on several types of materials, including leaves, metals, waxes, woods and greeting cards. Furthermore, we demonstrate how retargeting can be used to enhance diffuse texture with high quality reflectance.

Links: DL PDF WEB

1 Introduction

Editing Measured Materials. In the past years, the use of measured materials in Computer Graphics has grown since these datasets capture the nuances of real-world surface appearance. For many applications, editing these datasets is desired to permit artistic control. The editing process is typically comprised of performing soft selections on the data and applying edits to each selected region. For example, one might want to increase the roughness of the body of a leaf without changing its stem. Prior work has focused on simplifying selection [Pellacini and Lawrence 2007; An and Pellacini 2008], but does not address the issue of finding the proper editing parameters. For anything but the simplest cases, the latter remains remarkably cumbersome since editing parameters are not intuitively related to appearance changes. This implies that significant trial-and-error is needed after selection, e.g., minutes for simplest cases as reported in [Kerr and Pellacini 2010]. This is further complicated by the fact that different material datasets use different representations: from analytic (e.g. Cook-Torrance [1982] BRDFs) to sampled tables (e.g., [Ashikhmin et al. 2000]). These different representations require entirely different editing operations and potentially hundreds of parameters for the sampled representations, making the process even harder.

Retargeting Measured Materials. In this paper, we focus on editing spatially-varying measured materials that can be described by spatially-varying BRDFs (bidirectional reflectance distribution function) for opaque surfaces or heterogenous BSSRDFs (bidirectional subsurface reflectance distribution function) for optically-thick translucent materials. Inspired by color transfer methods, we propose to edit a *source* measured material by applying the reflectance functions defined in a *template* dataset. For example, as seen in Fig. 1, one might want to apply the reflectance functions of a young leaf to rejuvenate an old one. The resulting dataset is a spatially-varying material that maintains the spatial patterns of the source datasets, while exhibiting the reflectance behaviors of the template. Compared to select-and-modify editing workflows, the main benefits of appearance retargeting is a drastic reduction in the user interaction needed to obtain the edited results, just like color transfer. In our work, the user only has to mark sparse correspondences in the two datasets, using rough strokes, to indicate the desired edits.

Retargeting Principles. In generating the results of this paper, we found necessary to enforce the following principles. We share the first two principles with appearance-space edit propagation [Pellacini and Lawrence 2007].

1. To maintain the spatial patterns of the source dataset, we should enforce the policy that similar source appearance is retargeted to similar appearance in the template. Note that while our goal is to maintain patterns in the spatial domain, the corresponding policy is formulated entirely in the appearance domain.
2. To remain intuitive to users and support artistic exploration, the specified user correspondences should be satisfied after retargeting.

In addition, we introduce three new principles that are specific to retargeting.

3. The retargeting algorithm should be independent of the appearance representations, supporting all models from analytic to sampled ones.
4. The result of retargeting should maximally reproduce the richness of variations of the template data by ensuring that the variance of the dataset is present in the result.
5. To maintain the realistic look of the template data, the resulting datasets should contain reflectance functions either present in the template dataset or carefully interpolated from template functions that are neighbors in the appearance domain. In generating the results in this paper, we found that copying template functions directly worked well.

AppWarp. This paper presents *AppWarp*, an algorithm for appearance retargeting that enforces the requirements stated above. Our algorithm works on spatially-varying BRDFs and heterogenous BSSRDFs, regardless of the chosen representation. For the remainder of this section, we will describe our proposed method for the case of BRDFs to simplify the presentation. We formulate retargeting as the problem of assigning a BRDF in the template dataset for each BRDF in the source, under the constraints that user correspondences are respected, the richness of variations in the original dataset is well reproduced and that similar template BRDFs are assigned to similar source BRDFs. Our algorithm works entirely in the appearance domain.

Our key observation is that we can make the algorithm drastically more efficient and numerically stable by working on low-dimensional projections of each dataset, since we only need to compute this retargeting map between source and template BRDFs, rather than all possible BRDFs. We use the term *appearance space* to indicate these low dimensional projections. We use multidimensional scaling to compute these projections since it will maintain distances between all pairs of points in the space, thus allowing us to efficiently measure appearance similarity as Euclidean distance. As in prior work [Matusik et al. 2003; Wang et al. 2006], we found that the intrinsic dimensionality of the each individual datasets is low (two to three dimensions for the datasets in this paper). Figure 1 shows the appearance spaces of source, template, and result, where each point corresponds to a BRDF in the dataset, and distance between two points corresponds to the distance between the corresponding BRDFs.

We retarget the source by nonlinearly warping source points in the appearance space onto template points in such a way that the regions stroked on source and template maximally overlap after the warping to respect user's intentions, and that the global shapes of the two point sets maximally overlap after the warping to reproduce the rich variance of the template. The warping function is nonlinear and locally smooth to adapt to the complex shapes of source and template data and to ensure that similar source points are retargeted to similar template ones. We compute this warping function by solving a soft-constrained nonlinear optimization using an iterative method that converges in a few seconds. After the warping is computed, source and template are aligned, and we can assign the closest template point to each source point. Figure 1 shows the source dataset aligned with the template dataset in the appearance space after the warping.

Material and User Input. *AppWarp* is independent of the representation of the BRDF, depending only on a distance metric that can be defined for all representations. This allows us to support any measured BRDF type. Furthermore, it also allows for cross-representation retargeting where source and template are not stored using the same representation. We have tested *AppWarp* on a variety of BRDF representations, from analytic to sampled models, and found it to the work well for all cases.

We have extended the basic algorithm to support heterogenous BSSRDFs by leveraging the *SubEdit* representation [Song et al. 2009]. *SubEdit* decouples the BSSRDF defined for each pair of points into scattering profiles defined independently at each location. This allows us to compute distances and copy template data independently at each point, which are the only assumptions of our algorithm. We also show that our algorithm can be used to enrich albedo images with full reflectance data by using an albedo image as source and a measured material as template datasets.

Finally, we found that for all results in this paper, only a few seconds of user interaction is needed to perform a retargeting, since *AppWarp* is robust with respect to small errors in user input.

Comparison to Prior Work. Ignoring differences in user interaction, the main distinction between our approach and prior work is that we make no assumptions on the distribution of points in the appearance domain. Appearance-space edit propagation [Pellacini and Lawrence 2007; An and Pellacini 2008] linearly interpolates editing parameters, thus matching template appearance only for simpler distributions than the ones we found in our datasets (e.g., see Fig. 1). Appearance transfer methods either assume that appearance space is a function of geometry or weathering [Mertens et al. 2006; Lu et al. 2007; Wang et al. 2006] or that source and target come from the same distribution [Dong et al. 2010; Debevec et al. 2004]. *AppWarp* makes no such assumptions, allowing retargeting of a much larger variety of datasets.

Contributions. In summary, this paper proposes an algorithm for material retargeting that makes the following contributions:

- we present the first general retargeting algorithm that works for very different datasets, including retargeting of BRDFs and BSSRDFs, and enriching albedo images,
- we determine principles sufficient for meaningful retargeting,
- we derive an appearance-space warping algorithm that respects those principles,
- our algorithm is independent of the specific BRDF representation, even allowing cross-representation retargeting.

2 Related Work

Appearance Propagation. Editing measured materials is challenging since edits need to respect the intricate spatial patterns

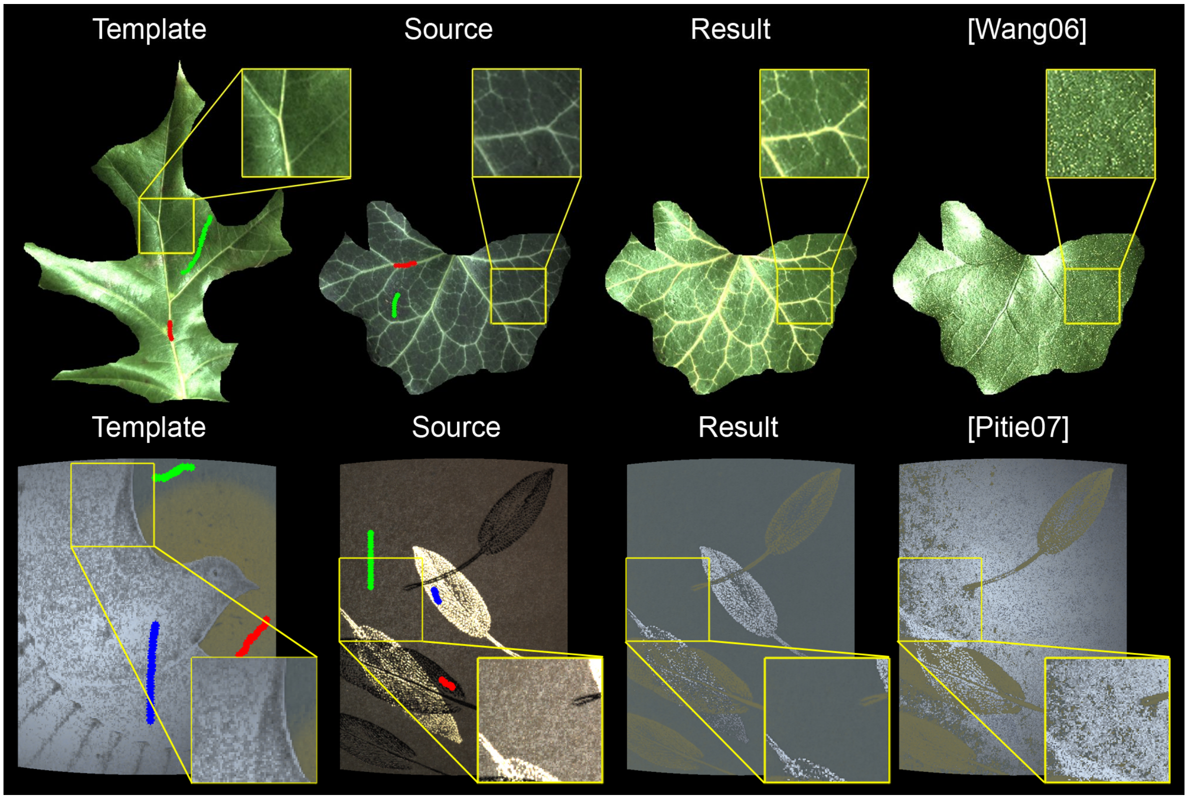

Figure 2: *Top: Comparison of retargeting by* AppWarp *using all appearance-space dimensions (in this case two) versus projecting all points onto one dimension as in suggested in [Wang et al. 2006]. Note how the variance in the data is not preserved after the one-dimensional retargeting. Bottom: Comparison of* AppWarp *with n-dimensional histogram matching [Pitié et al. 2007]. The noise generated by histogram matching is a result of the attempt to match distributions perfectly, thus violating the principle that similar template reflectance functions should be assigned to similar source ones. Data from [Lawrence 2009].*

found in the data and preserve the richness of the reflectance functions. [Pellacini and Lawrence 2007] proposed an algorithm where users specify editing parameters at a set of sparse locations by drawing rough strokes. The algorithm computes a soft selection for each stroke and propagates the stroke's editing parameters to the rest of the dataset by linearly interpolating their values based on these selections. [An and Pellacini 2008] generalize on this work by proposing a more robust algorithm that works for both images and measured materials. These methods support arbitrary edits while ensuring that regions of similar appearance receive the same edit. In the context of retargeting, two issues arise in attempting to match the template appearance. From a user interaction perspective, significant time is required to determine the editing parameters. [Kerr and Pellacini 2010] showed that minutes are required for analytic representations that have only a few parameters, making it impractical to edit more complex representations such as the microfacet model presented in [Wang et al. 2008] where an artist would have to adjust hundreds of parameters. In retargeting, we entirely obviate the need to specify editing parameters. At a more fundamental level, since the editing parameters are propagated linearly, the overall target appearance is only matched for simpler distributions than the ones we found in our datasets (e.g., see the shape of appearance space in Fig. 1). In *AppWarp*, we employ a nonlinear warp that guarantees that the look of the template is reproduced.

Appearance Transfer. Prior methods in appearance transfer fall into two main categories. Examples of first category include [Mertens et al. 2006], [Lu et al. 2007], and [Wang et al. 2006], where transfer is performed under the assumption that the appearance space is a function of geometric information or weathering degree maps. In our work, we make no such assumptions and allow for a more general retargeting operation. For example, we show in Fig. 2 that using the method proposed in [Wang et al. 2006] for retargeting does not provide satisfactory results. Examples of the second category include [Dong et al. 2010] and [Debevec et al. 2004], where the appearance of a small set of high-dimensional exemplars is transferred to a large set of low-dimensional projections under the assumption that source and target come from the same material, i.e., the point sets have the same distribution in appearance space. *AppWarp* is a more general method that allows retargeting between two different datasets, strongly relaxing the assumptions taken in this prior work.

Color Transfer. In color transfer, the user recolors a source image by matching the color distribution of a template image. Automatic transfer methods attempt to globally match the color distributions of source and template, either by exactly matching the color distributions via histogram matching [Pitié et al. 2007] or by assuming simple distributions such as Gaussian mixtures [Reinhard et al. 2001]. However, artifacts are generated in the results when the source and template distributions are sufficiently dissimilar, which is the case for the majority of image pairs. Region-by-region methods have been introduced to attempt to address this well-known issue, relying on image domain segmentation to perform the clustering (see [An and Pellacini 2010] for a recent review). For corresponding regions, a transfer function is computed by matching distributions, and propagated to the rest of image using edit propagation methods. Our work is inspired by user-guided color transfer algorithms in terms of workflow exposed to users.

The main difference of our work with color transfer methods is that the latter work on low-dimensional vectors, while we work on high-

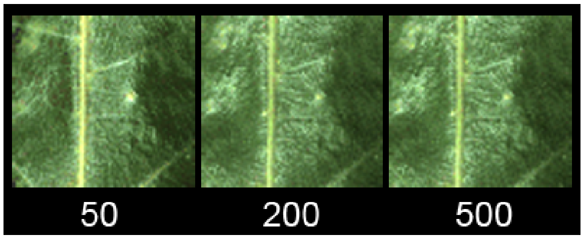

Figure 3: *A crop showing the details of the retargeting result from Fig. 1 computed with different sampling rates in a two dimensional appearance space. From left to right are results generated using 50, 200, and 500 subdivisions for each appearance-space dimension, corresponding to a warp function represented by roughly 600, 3000, 9000 affine matrices. Note that the result becomes stable at roughly 200 subdivisions. Data from [Lawrence 2009].*

dimensional reflectance functions defined in a variety of representations. While the appearance spaces of source and target are low dimensional, these spaces are data dependent and differ between source and target. Color transfer algorithms take advantage of the fact that all colors are defined in the same low-dimensional RGB (or Lab) space by using heuristics valid in these known spaces. Furthermore, most color transfer methods generate new colors not present in the template. In our case, we cannot generate new reflectance functions arbitrarily since this likely leads to unrealistic appearance.

3 Appearance Retargeting

Appearance Retargeting. We define retargeting as the process of applying the reflectance functions defined in a *template* measured material to a *source* measured dataset in a way that maintains the spatial patterns of the source. In general, determining which regions of the template data are applied to which region of the source is arbitrary. In our work, we let the user define this assignment. To minimize user interaction, we ask artists to mark corresponding regions on source and template using pairs of rough strokes. Based on these stroke-to-stroke correspondences, our algorithm computes a point-to-point correspondence between each location in the source to one location in the template. The retargeted dataset is then generated by assigning the corresponding template reflectance for each source point. The remainder of this section describes our algorithm and compares it to other methods. While our method supports BRDFs and BSSRDFs, we focus on the former to simplify the description. Notice that while we chose to use sparse strokes since they have shown to work well in material editing [Pellacini and Lawrence 2007], our algorithm works for any type of sparse, user-defined correspondence between source and target.

Appearance Space. As shown in prior work [Matusik et al. 2003; Wang et al. 2006], we observe that the intrinsic dimensionality of measured material datasets is low due to the self-similarities in appearance among points on the same surface. To improve computational efficiency and numerical stability, we project these datasets into low-dimensional spaces using a distance preserving method. For efficiency reasons, we have chosen to use landmark MDS [de Silva and Tenenbaum 2004] for all the results shown in this paper[1]. For the datasets included in this paper, we found that two or three dimensions are sufficient to capture most of the variance[2]. Since MDS computes projections up to a rotation, we roughly align the resulting spaces onto each other by applying a global affine transformation that aligns the centers of corresponding stroke pairs.

For each dataset, the result of dimensionality reduction is a collection of projections of the BRDFs in the dataset. These low-dimensional projections span a low-dimensional space that we refer to as *appearance space*. More specifically, each point $\mathbf{s}_i \in S$ of the source point set S in appearance space corresponds to the BRDF ρ_i defined at surface point h in the source dataset, and each point $\mathbf{t}_j \in T$ of the template point set T in template appearance space corresponds to the BRDF ρ_j defined at surface point k in the template dataset.

Distance Measure. To compute the low-dimensional projections using MDS, we only need to define a distance metric between any two BRDFs. We follow prior work [Wang et al. 2006; Pellacini and Lawrence 2007] and measure the similarity in reflectance function BRDFs ρ and ρ' as:

$$d^2(\rho,\rho') = \int_{\Omega_i}\int_{\Omega_o} (\rho(\omega_i,\omega_o) - \rho'(\omega_i,\omega_o))^2 cos(\omega_i) d\omega_o d\omega_i,$$

where Ω_i and Ω_o are defined with respect to the local frame, so that d measures distances in appearance regardless of the underlying geometry. For anisotropic materials, we pick a local frame by finding the main direction of anisotropy based on the statistical properties of the NDFs. We then rotate anisotropic BRDFs to be aligned with respect to their local frame. Figures 7 and 8 show examples of retargeting with anisotropic materials. To reduce the cost of evaluating this function for all pairs of points, we follow [Pellacini and Lawrence 2007] and approximate this integral by sampling while ensuring that all specular peaks are well accounted for.

3.1 Constrained appearance-space warping

Retargeting as Warping. In appearance space, retargeting can be expressed as mapping the appearance-space coordinates of each point $\mathbf{s}_i \in S$ to a point $\mathbf{t}_j \in T$. This mapping can be expressed as applying a warping function $F : S \rightarrow T$. The goal of our method is to compute a warping function F that respects the principles we listed in the introduction. The application of the warping function is illustrated in Fig. 1. To remain independent of the BRDF representation (principle 3), we compute the warping function solely based on appearance-space coordinates, since these are derived by a distance metric that is defined independently of the chosen BRDF representation.

Warping Function Representation. One might wonder whether a simple warping function works well for retargeting. We found that because of the complex and case-by-case varying shapes of these appearance spaces, simple heuristics don't work as well. Inspired by moving least square techniques [Schaefer et al. 2006], we chose a warp function that is locally linear, thus efficient and stable to fit, and globally nonlinear, thus able to capture the global shape of the data in appearance space. Specifically, we represent the warping function F as a set of affine transformations defined at a sparse set of points in the appearance space. To determine these locations, we first partition the appearance space as a uniform grid of bins. All source points $\mathbf{s}_i$ contained in the same bin b share the same affine transformation M_b specified at the center of the bin $\mathbf{p}_b$. Note that we only need the affine M_b if at least one source point is present in that bin, giving us a sparse sampling of the appearance space. With this representation we can write the warping function as

$$F(\mathbf{s}_i) = M_b\mathbf{s}_i = \mathbf{s}'_i \text{ for } \mathbf{s}_i \in \text{bin } b.$$

We found that the quality of results is not affected by this sampling provided that there are enough samples to capture the global shape

[1] We have also tested landmark Isomap [Tenenbaum et al. 2000]. We found that while the shapes of the point sets computed using Isomap versus MDS can be different, the results are visually the same. We believe this is because our method relies on local distances rather than on global ones.

[2] As discussed in Sec. 3.3, our algorithm does not depend on the particular intrinsic dimensionality of the data, provided it is low, but only on the requirement that the dimensionality of the template dataset is lower or equal to the dimensionality of the source.

of the point set and maintain its variance. Figure 3 shows the typical behavior of how sampling rates affect the quality of results for different binning. For all results in this paper, we found experimentally that using 200 subdivisions for each dimension works well. This corresponds to roughly 2,000 to 10,000 samples depending on the shape of the source dataset.

Warping Function Optimization. To determine the affine matrices M_b, we formulate an optimization problem in such a way that the principles stated above are respected. More specifically we minimize a function of three terms:

$$\arg\min_{M_b} \sum_b \sum_{b' \in N_b} z_{b,b'} ||(M_b - M_{b'})\mathbf{p}_b||^2$$
$$- \frac{|S|}{\sum_k |\hat{S}_k|} \sum_k O(\hat{T}_k, \hat{R}_k) - O(T, R),$$
$$\text{with } z_{b,b'} = exp(-|\mathbf{p}_b - \mathbf{p}_{b'}|^2/2\sigma^2)$$

where O measures the overlap of two point sets, R is the set of points resulting from applying F to each point in S, N_b is the set containing the nearest neighbors of point $\mathbf{p}_b$ in appearance space, $z_{b,b'}$ is the similarity between points b and b' defined as in [Pellacini and Lawrence 2007], $\hat{T}_k$ and $\hat{R}_k$ are the point sets formed by the k-th stroke pair. For all results in this paper we use 10 nearest neighbors as in [Pellacini and Lawrence 2007]. In this function, the first term preserves local neighborhoods, the second ensures overlapping of stroke pairs, and the last maximizes the overlapping of the whole point sets.

Each term of this optimization function corresponds to imposing one of our principles. To enforce that source points with similar appearance are mapped to similar template points (principle 1), thus maintaining the spatial patterns of the source in the result material, the warping function should be locally-smooth as to preserve local neighborhoods after retargeting. To respect the user's intensions (principle 2), the warping function should map stroked source points to the corresponding stroked points in the template. To reproduce the rich variance of the template datasets (principle 4), the warping function should deform the source point set as to maximally overlap with the overall shape of the template one.

To measure the overlap O between two point sets X and Y, we compute the occupancy o of each bin b for each dataset as o_b^X and o_b^Y, where $o_b = 1$ if there is at least one point from the point set present in bin b, and evaluate their overlap as the number of bins where the X and Y have the same occupancy as $O(X,Y) = \sum_b \delta(o_b^X, o_b^Y)$ where δ is the Kronecker delta. Intuitively, the more the two point sets overlap, the larger the value $O(X,Y)$[3]. Note that our overlapping measure O is different than measuring the difference of cumulative distribution functions (CDFs) as histogram matching does. As we will show later in this section, it is in fact our belief that matching CDFs fundamentally violates the principle that similar template points should be assigned to similar source ones.

Reflectance Copy. To ensure that only template BRDFs are used in the result (principle 5), we map each warped source point to the closest template one. Applying the affine transformations obtained from this optimization gives us a result point set $\mathbf{s}_i' = F(\mathbf{s}_i)$ that closely overlaps with the template point set (e.g., see Fig. 1). For each point $\mathbf{s}_i'$, we perform a nearest neighbor search to find the closest template point $\mathbf{t}_j \in T$. We then assign the template reflectance function ρ_j to the source location i. In general, this copy operation might have the drawback of assigning too many source points to the same template reflectance. We found this not to be the case for our application, since for high resolution datasets the appearance space is densely sampled, thus reducing the possibility of multiple copies. As an alternative, one could "interpolate" a new template BRDF from its nearest neighbors in appearance space. In our experiments, we found that this was not desired since copying the closest BRDF worked well and since, for nonlinear analytic models, properly interpolating BRDFs is a complex operation.

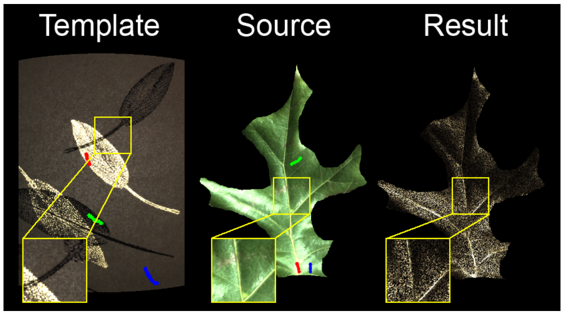

Figure 4: *Example of limitation of* AppWarp. *Without considering the spatial arrangement of samples, two different parts of the paper cannot be assigned to the body of the leaf. Furthermore, since the template is higher dimension than the source, the variance of the template is mapped randomly to the source data, generating what looks like strong noise in the spatial domain. Data from [Lawrence 2009].*

Solving Optimization. Due to the presence of the O terms, computing the affine matrices M_b amounts to solving a large nonlinear optimization. We solve this optimization with an iterative simplex method that requires careful initialization to ensure convergence. To initialize the affine transformations, for each bin c, marked by a source stroke in $\hat{S}$, we assign a corresponding bin c' in the corresponding template stroke $\hat{T}$. This provides us with a set of pairwise correspondences $(\mathbf{p}_c, \mathbf{p}_c')$. We compute this assignment with the n-dimensional histogram matching procedure presented in [Pitié et al. 2007], but we match occupancy rather than CDFs. We then compute an affine matrix at each sample in the source point set using moving least squares [Schaefer et al. 2006] by minimizing the equation $\arg\min_{M_b} \sum_c z_{b,c} |M_b \mathbf{p}_c - \mathbf{p}_{c'}|^2$. Intuitively, this tries to find an affine that can best transform all the stroked bins $\mathbf{p}_b$ in the source to their corresponding $\mathbf{p}_b'$ in the template, where the closer a bin $\mathbf{p}_b$ is to a stroked point $\mathbf{p}_c$, the more the latter influences the final affine transformation. Note that the steps taken for initialization already respect the principles we want to apply for retargeting.

Starting from this initial assignment, we determine the final solution using the iterative simplex method [Lagarias et al. 1998]. Due to the large number of parameters to solve, we choose to optimize over a small subset of parameters (we used 10) chosen randomly at every iteration. In our experiments, we randomly pick bins whose $\delta = 1$, or whose neighboring bins have $\delta = 1$, because they are in areas where the two point sets are not overlapping. We found this process to converge well, also helped by the choice of initial configuration, and produces visually stable results that match the optimization with all parameters.

3.2 Comparison to Prior Work

To the best of our knowledge, this paper is the first to present an algorithm for retargeting measured materials that is general with respect to material type and the distribution of points in appearance space. One could argue though that simpler methods, inspired by color transfer or material transfer, might work for retargeting and obviate the need for our warping algorithm. However, we found the opposite to be true.

[3] While the Hausdorff distance might have also worked well to compute this measure, we found it computationally too expensive to employ.

Histogram Matching. In color transfer, histogram matching [Pitié et al. 2007] is a commonly used statistical method that transfer the color from a template to a source by matching cumulative distribution functions (CDFs) between two n-dimensional datasets. However, this method results in strong visual artifacts when the distributions of the two datasets are dissimilar [Pitié et al. 2007; An and Pellacini 2010]. We found this to be the case for most material retargeting examples we have tested. For example, the bottom row of Fig. 2 shows a comparison retargeting greeting cards using our method and n-dimensional histogram matching applied to the appearance-space coordinates. As can be seen, histogram matching is too strict in respecting CDFs, thus violating the user constraints as well as introducing visible noise in the result. This is a fundamental limitation of this method since its basic principle, i.e., matching cumulative distribution functions (rather than occupancies), violates the principle that similar source appearance is retargeted to similar appearance in the template thus does not lead to meaningful retargeting results.

Appearance Manifolds. [Wang et al. 2006; Gu et al. 2006] analyzed time-varying BRDFs, typically from aging or weathering processes, and found that one-dimensional nonlinear appearance manifolds capture well the degree of aging at each pixel in the dataset. [Wang et al. 2006] uses these manifolds to synthesize new materials by generating one-dimensional weathering degree maps via texture synthesis and looking up the BRDFs from the original data. While this appears similar to our work, there are two main distinctions. First, we do not rely on texture synthesis but seek instead to maintain the source patterns. More importantly, we found that one-dimensional projections do not capture enough of the variance on most measured datasets. For example, it is obvious from the low dimensional projections shown in Fig. 1 that one dimension is not sufficient to capture major variances in the datasets. This means that retargeting in one dimension will create unsatisfactory results. The top row of Fig. 2 shows a comparison between retargeting in one and two dimensions. Note how the one-dimensional projection not only loses the richness of the original datasets, but also breaks the original spatial patterns of the data, whereas our result resembles the template while exhibiting no visual artifacts.

3.3 Limitations

The underlying assumption of our algorithm is that retargeting is purely an appearance-space operation. It is possible though that a user might want to retarget different parts of an object with the same appearance to different template locations, as has been shown in local color transfer [An and Pellacini 2010]. While this might be desirable, it inevitably requires the introduction of discontinuities and gradient reversals in the appearance domain that generate visible artifacts in the result. We believe that if these spatial retargeting operations are desired, our retargeting approach needs to be coupled with soft selection and propagation methods, but we leave this to future work.

Another assumption of our algorithm is that the intrinsic dimensionality of the template dataset is lower or equal to that of the source. Without this constraint, some of the variance of the template would be assigned randomly to source points causing noise in the result. While this is clearly a limitation of our approach, we feel that in this situation it is not possible to perform retargeting effectively, since this mismatch in dimensionality means that the source data has inherently less variance than the template which simply cannot be used as guidance. In fact we feel that this is an advantage of our method, compared to other ones such as histogram matching, since we have a clear definition of when transfer will fail (lack of variance in the source compared to the template in appearance space).

Figure 4 shows an example where both these limitations are present (see Table 1 for the dimensionality of the appearance spaces).

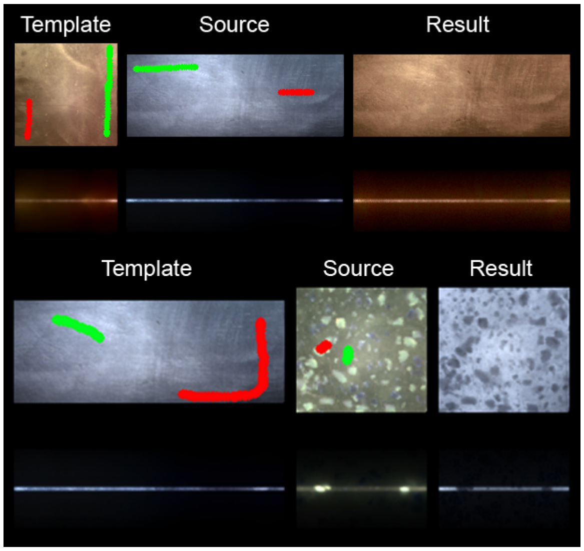

Figure 5: *Retargeting results between heterogeneous BSSRDF datasets of waxes and marble. For each dataset, the top row is rendered under a large area light, while the bottom is rendered using a line light to show the scattering effects. Data from [Song et al. 2009].*

4 Extensions

Surface scattering. The behavior of subsurface scattering materials is modeled by the bidirectional subsurface scattering distribution function. The BSSRDF $S(x_i, \omega_i; x_o, \omega_o)$ relates the outgoing radiance $L(x_o, \omega_o)$ at a point x_o to the incoming radiance $L(x_i, \omega_i)$ at x_i as

$$L(x_o, \omega_o) = \int_A \int_\Omega S(x_i, w_i; x_o, w_o) L(x_i, w_i)(n(x_i)\dot{\omega}_i) d\omega_i dx_i,$$

where A is the area around x_o, Ω is the hemisphere around x_i, and $n(x_i)$ is the normal at x_i. In subsurface scattering materials, the appearance at any surface point is not independent from other surface points. This non-local behavior makes it impossible to directly use our retargeting algorithm that relies on copying reflectance independently at different surface location. However, given a proper representation of BSSRDFs, retargeting can indeed be performed.

SubEdit [Song et al. 2009] models the non-local behavior of the diffuse component of the BSSRDF as the product of two local scattering profiles, defined at the outgoing and incoming points. These scattering profiles are one-dimensional functions P_x defined at each surface location x independently from one another. Intuitively, the scattering profiles measure the decay in light scattering with respect to distance from the point x. Distances between two BSSRDFs S_x and S_y defined at x and y can be defined as:

$$d^2(S_x, S_y) = \int_0^\infty ||P_x(r) - P_y(r)||^2 r dr,$$

where P_x and P_y are the scattering profiles of surface point x and y. In [Song et al. 2009], the authors show how independent edits to scattering profiles generate plausible BSSRDFs. We follow a similar approach for retargeting. We use the above distance metric to compute an appearance space for the scattering profiles, and employ our algorithm to warp the appearance space points and copy the scattering profiles independently from one to another to retarget the original BSSRDFs. Figure 5 shows that our algorithm gives satisfactory results for heterogenous subsurface scattering materials as well.

Material	Representation	Resolution	Sampled bins	Dim.
green oak	linear Phong	500 × 600	2392	2
brown oak	linear Phong	512 × 562	3227	2
green ivy	linear Phong	497 × 453	1806	2
dove	Ward	220 × 255	3496	3
leaf card	linear Phong	300 × 400	3942	3
yellow wax	BSSRDF	110 × 110	5324	2
blue wax	BSSRDF	88 × 232	6018	2
art stone	BSSRDF	107 × 112	3287	2
rusted metal	linear Phong	795 × 365	4952	3
ring texture	texture	408 × 303	9654	3
scissor	rgbn	1000 × 446	14292	3
maple wood	Marschner	352 × 303	5258	2
walnut wood	Marschner	407 × 351	5622	2
burning wood	Torrance	512 × 512	1210	2
drying wood	Torrance	300 × 300	2827	2
copper patina	Microfacet	497 × 473	5682	3

Table 1: *Statistics for the material datasets shown in the paper, including reflectance representation, data size, number of samples used in optimization, and the dimensionality of the dataset.*

Figure	Template	Source		Timing	
			Init (s)	Opt (s)	Copy (s)
Fig. 1	Green oak	Brown oak	8	12	2
Fig. 2	Green oak	Green ivy	9	8	2
Fig. 2	Dove	Leaf card	11	9	1
Fig. 5	Yellow wax	Blue wax	13	26	1
Fig. 5	Blue wax	Art stone	11	18	1
Fig. 6	Rusted metal	Ring texture	13	20	1
Fig. 6	Rusted metal	Scissor	13	29	1
Fig. 7	Maple wood	Walnut wood	10	15	2
Fig. 8	Burning wood	Drying wood	10	11	2
Fig. 8	Copper	Rusted metal	12	13	1

Table 2: *Timing for all retargeting results shown in this paper, split into initialization, iterative optimization and final copy. Note how some of the retargeting results are cross-type, i.e., source and template do not have the same representation.*

Enrichment. Our algorithm can also be extended to datasets other than materials that can be represented in low dimensional spaces. One useful application might be to use a measured material as template to enrich diffuse texture maps and derive a SVBRDF dataset based on the diffuse value only. The only requirement for this to work is that the dimensionality of the template is less than or equal to the dimensionality of the diffuse map. An example of diffuse texture enrichment can be found in Fig. 6 top row, where we render the source texture and result appearance on a pillow geometry. We can also also enrich *rgbn* images [Toler-Franklin et al. 2007], i.e., diffuse map and normals per pixel, as shown in Fig. 6 bottom row. In each case, the specular component of the reflectance is transferred successfully to produce a believable look that exhibits no visual artifacts. This might be useful to quickly enrich diffuse texture maps found in commonly used texture libraries.

5 Results and analysis

Throughout this paper, we tested our algorithm across a wide variety of materials, including organic objects (Figs. 1, 2), metals (Figs. 6, 8), greeting cards (Fig. 2), waxes (Fig. 5) and woods (Figs. 7, 8). These very different datasets can be handled by our method since we make no assumptions on the distribution of points in the appearance domain. We believe that these results would otherwise be very challenging to achieve even for experts. To appreciate the quality of results best, we refer the reader to the supplemental materials for full resolution rendering results of all example transfers.

Performance and Scalability. Table 1 lists the datasets we used throughout the paper. We report the material representation type, its resolution, the number of affine transformations used in the algorithm and the reduced dimensionality of the dataset. For all materials we tested, the intrinsic dimensionality is between two and three, and the number of affine transformations are typically in the range of a few thousand. Table 2 lists the execution time for each retargeting example in this paper. Retargeting takes on average around 30 seconds with an unoptimized, single-thread implementation executed on an Intel Core 2 running at 2.83 GHz with 4 GB RAM. Due to sparse sampling, the execution time of our algorithm is nearly independent of data size, supporting high resolution datasets well. In terms of memory, our algorithm grows linearly with the size of the data, only requiring the storage of the reflectance projections.

Sensitivity to stroke locations. Like all stroke-based methods, the results of our retargeting algorithm depend on the user's intentions expressed as stroke placements. However, we found that precise positioning of strokes is not necessary, since our algorithm is stable against changes in stroke location and sizes. Figure 7 shows an example of retargeting between polished woods, where two sets of sparse strokes are tested on the retargeting pair. Although the locations of strokes are very different between these two sets, the user's intentions are the same: transfer the appearance of the grains and body of the maple to the grains and body of the walnut respectively. We show that the results of retargeting are visually the same. We believe the main reason for this stability is that due to low dimensionality of the appearance space, a very sparse set of stroked points is sufficient to sample it well, thus capturing user's intentions in a numerically-stable manner.

Material Types. One major benefit of our algorithm is that it is fundamentally independent from the underlying material representation for BRDFs since reflectance distances can be defined for all models. We found that our algorithm works well with a large variety of material representations and with allowing cross-type retargeting between different representations, further strengthening the generality of our approach. In the remainder of this section, we quickly summarize the representations used.

- *Analytic BRDFs*. We support a variety of analytic representations. The dove greeting card in Fig. 2 is represented as anisotropic Ward [Ward 1992]. The burning and drying wood from [Gu et al. 2006] in Fig. 8 use a modified Cook-Torrance BRDF [Cook and Torrance 1982]. Finally, Fig. 7 uses the finished wood representation from [Marschner et al. 2005].
- *Linear Basis Projection*. [Holroyd et al. 2010] models reflectance at each surface point of a spatially-varying material as a linear combination of a small set of basis BRDFs. We employ such a representation for Figs. 1, 2, 6, 8 where we use five Phong BRDFs of varying specular roughness as basis.
- *Microfacet*. Microfacet-based representations model the BRDF by tabulating a normal distribution function for each point, from which reflectance can be computed [Ashikhmin et al. 2000]. Figure 8 shows an example of retargeting an anisotropic copper patina, represented as a microfacet model, to rusted metal, represented as linear basis projection.
- *SubEdit BSSRDFs*. Figure 5 shows examples of retargeting BSSRDFs by employing the *SubEdit* representation. We refer the reader to Section 4 for further details.

6 Conclusion and Future work

In this paper, we introduce a general formulation for retargeting measured materials where the spatially-varying reflectance of a source dataset is edited by applying the reflectance function of a template dataset, while still maintaining the spatial patterns shown in the source. We perform retargeting as an appearance-space warp-

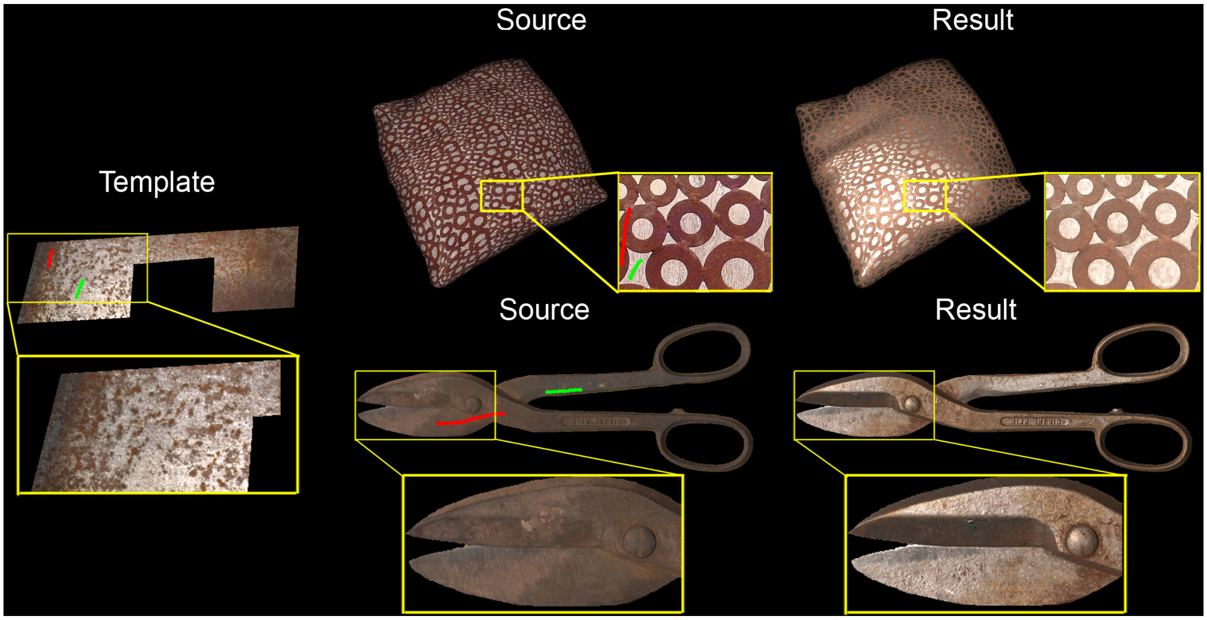

Figure 6: *Two results of using a rusted metal strip as template to enrich (top) a diffuse texture applied to a pillow and (bottom) an* rgbn *image. Data from [Slavens 2005; Toler-Franklin et al. 2007; Wang et al. 2008].*

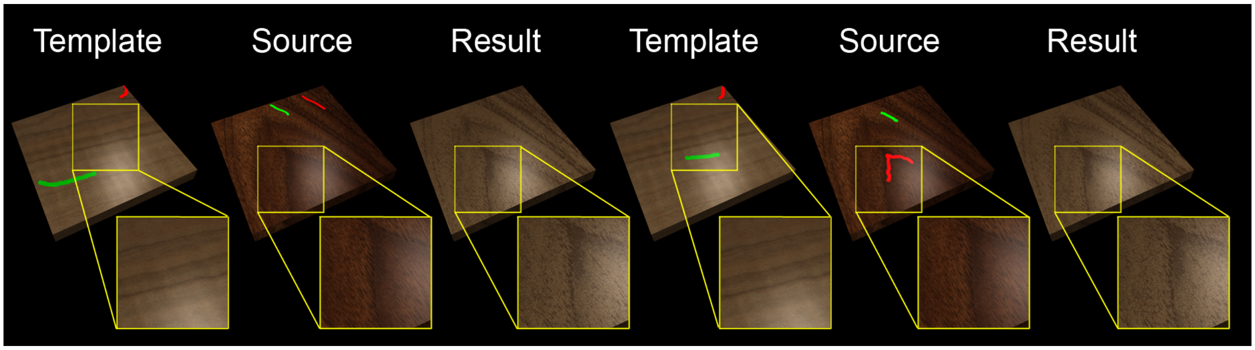

Figure 7: *Retargeting results of wood datasets computed using different stroke positions to show the stability of our approach with respect to stroke placement. Data from [Marschner et al. 2005].*

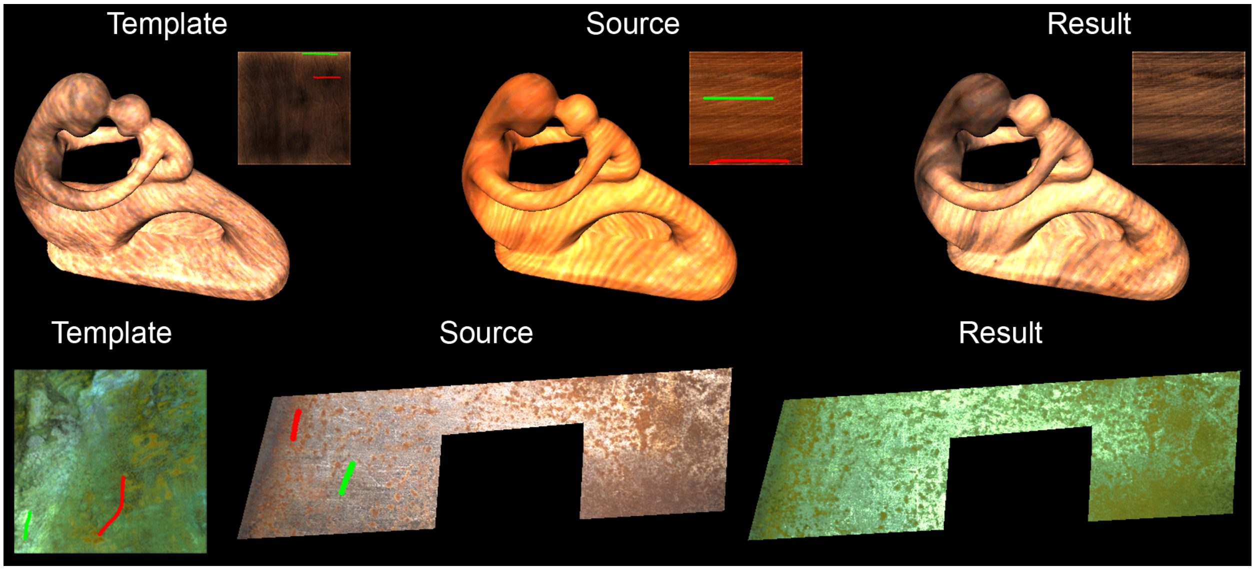

Figure 8: *Top: Retargeting result of wood samples. Note the original textures are shown here on the right corner of each rendering. Bottom: Retargeting result of metal samples. Data from [Gu et al. 2006; Lawrence 2009].*

ing operation that requires minimal user input and runs in tens of seconds. We show the effectiveness, stability, and generality of our algorithm by performing retargeting on a number of measured material datasets represented using various reflectance models. For future work, we would like to explore how to extend our framework to beyond the appearance domain, for instance to encompass spatial domain operations into the retargeting process. We also like study how to utilize local geometric information, such as normals and tangents, to further enhance retargeting results.

7 Acknowledgements

We would like to thank Jason Lawrence and Pieter Peers for their discussion on the paper. This work was supported by NSF (CNS-070820, CCF-0746117), Intel, and the Sloan Foundation.

References

AN, X., AND PELLACINI, F. 2008. AppProp: all-pairs appearance-space edit propagation. *ACM Transactions on Graphics 27*, 3, 40:1–40:9.

AN, X., AND PELLACINI, F. 2010. User-controllable color transfer. *Computer Graphics Forum (EG) 29*, 2, 263–271.

ASHIKHMIN, M., PREMOŽE, S., AND SHIRLEY, P. 2000. A microfacet-based BRDF generator. In *SIGGRAPH*, 65–74.

COOK, R. L., AND TORRANCE, K. E. 1982. A reflectance model for computer graphics. *ACM Transactions on Graphics 1*, 1, 7–24.

DE SILVA, V., AND TENENBAUM, B. 2004. Sparse multidimensional scaling using landmark points. *Technical Report, Stanford University*.

DEBEVEC, P., TCHOU, C., GARDNER, A., HAWKINS, T., POULLIS, C., STUMPFEL, J., JONES, A., YUN, N., EINARSSON, P., LUNDGREN, T., FAJARDO, M., AND MARTINEZ, P. 2004. Estimating surface reflectance properties of a complex scene under captured natural illumination. *Technical Report, University of Southern California Institute for Creative Technologies Graphics Laboratory*, ICT-TR-06.

DONG, Y., WANG, J., TONG, X., SNYDER, J., LAN, Y., BEN-EZRA, M., AND GUO, B. 2010. Manifold bootstrapping for SVBRDF capture. *ACM Transactions on Graphics 29*, 4, 98:1–98:10.

GU, J., TU, C.-I., RAMAMOORTHI, R., BELHUMEUR, P. N., MATUSIK, W., AND NAYAR, S. K. 2006. Time-varying surface appearance: acquisition, modeling and rendering. *ACM Transactions on Graphics 25*, 3, 762–771.

HOLROYD, M., LAWRENCE, J., AND ZICKLER, T. 2010. A coaxial optical scanner for synchronous acquisition of 3d geometry and surface reflectance. *ACM Transactions on Graphics 29*, 4, 99:1–99:12.

KERR, W. B., AND PELLACINI, F. 2010. Toward evaluating material design interface paradigms for novice users. *ACM Transactions on Graphics 29*, 4, 35:1–35:10.

LAGARIAS, J. C., REEDS, J. A., WRIGHT, M. H., AND WRIGHT, P. E. 1998. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization 9*, 112–147.

LAWRENCE, J., 2009. Personal communication.

LU, J., GEORGHIADES, A. S., GLASER, A., WU, H., WEI, L.-Y., GUO, B., DORSEY, J., AND RUSHMEIER, H. E. 2007. Context-aware textures. *ACM Transactions on Graphics 26*, 1, 3:1–3:22.

MARSCHNER, S. R., WESTIN, S. H., ARBREE, A., AND MOON, J. T. 2005. Measuring and modeling the appearance of finished wood. *ACM Transactions on Graphics 24*, 3, 727–734.

MATUSIK, W., PFISTER, H., BRAND, M., AND MCMILLAN, L. 2003. A data-driven reflectance model. *ACM Transactions on Graphics 22*, 3, 759–769.

MERTENS, T., KAUTZ, J., CHEN, J., BEKAERT, P., AND DURAND, F. 2006. Texture transfer using geometry correlation. In *Rendering Techniques*, Eurographics Symposium on Rendering, 273–284.

PELLACINI, F., AND LAWRENCE, J. 2007. AppWand: editing measured materials using appearance-driven optimization. *ACM Transactions on Graphics 26*, 3, 54.

PITIÉ, F., KOKARAM, A. C., AND DAHYOT, R. 2007. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding 107*, 1-2, 123–137.

REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Computer Graphics and Applications 21*, 5, 34–41.

SCHAEFER, S., MCPHAIL, T., AND WARREN, J. D. 2006. Image deformation using moving least squares. *ACM Transactions on Graphics 25*, 3, 533–540.

SLAVENS, M., 2005. "rust cirlces 3". CC BY 2.0 image on Flickr.

SONG, Y., TONG, X., PELLACINI, F., AND PEERS, P. 2009. SubEdit: a representation for editing measured heterogeneous subsurface scattering. *ACM Transactions on Graphics 28*, 3, 31:1–31:10.

TENENBAUM, J. B., DE SILVA, V., AND LANGFORD, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500, 2319–2323.

TOLER-FRANKLIN, C., FINKELSTEIN, A., AND RUSINKIEWICZ, S. 2007. Illustration of complex real-world objects using images with normals. In *International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, ACM, 111–119.

WANG, J., TONG, X., LIN, S., PAN, M., WANG, C., BAO, H., GUO, B., AND SHUM, H.-Y. 2006. Appearance manifolds for modeling time-variant appearance of materials. *ACM Transactions on Graphics 25*, 3, 754–761.

WANG, J., ZHAO, S., TONG, X., SNYDER, J., AND GUO, B. 2008. Modeling anisotropic surface reflectance with example-based microfacet synthesis. *ACM Transactions on Graphics 27*, 3, 41:1–41:9.

WARD, G. J. 1992. Measuring and modeling anisotropic reflection. In *SIGGRAPH*, ACM, 265–272.