Multi-View Intrinsic Images of Outdoors Scenes with an Application to Relighting

SYLVAIN DUCHÊNE and CLEMENT RIANT and GAURAV CHAURASIA and JORGE LOPEZ MORENO[1]
and
PIERRE-YVES LAFFONT[2] and STEFAN POPOV and ADRIEN BOUSSEAU and GEORGE DRETTAKIS
Inria

We introduce a method to compute intrinsic images for a multi-view set of outdoor photos with cast shadows, taken under the same lighting. We use an automatic 3D reconstruction from these photos and the sun direction as input and decompose each image into reflectance and shading layers, despite the inaccuracies and missing data of the 3D model. Our approach is based on two key ideas. First, we progressively improve the accuracy of the parameters of our image formation model by performing iterative estimation and combining 3D lighting simulation with 2D image optimization methods. Second we use the image formation model to express reflectance as a function of discrete visibility values for shadow and light, which allows us to introduce a robust visibility classifier for pairs of points in a scene. This classifier is used for shadow labelling, allowing us to compute high quality reflectance and shading layers. Our multi-view intrinsic decomposition is of sufficient quality to allow relighting of the input images. We create shadow-caster geometry which preserves shadow silhouettes and using the intrinsic layers, we can perform multi-view relighting with moving cast shadows. We present results on several multi-view datasets, and show how it is now possible to perform image-based rendering with changing illumination conditions.

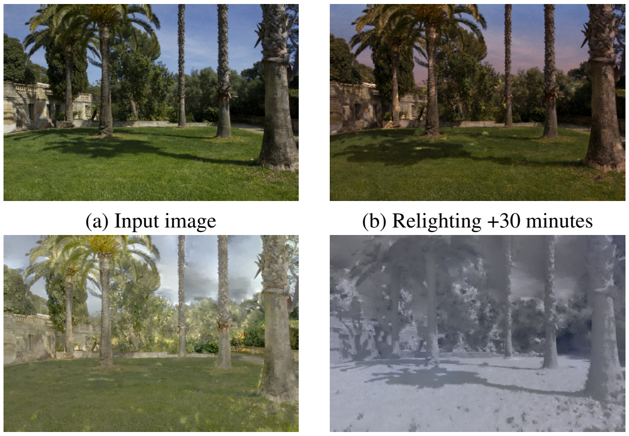
Categories and Subject Descriptors: I.3.3 [**Computer Graphics**]: Picture/Image Generation; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis

General Terms: Algorithms

Additional Key Words and Phrases: Relighting, Intrinsic Images, Shadow Detection, Reflectance, Shading

1. INTRODUCTION

Recent progress on automatic multi-view 3D reconstruction [Snavely et al. 2006; Goesele et al. 2007; Furukawa and Ponce 2007] and image-based-rendering [Goesele et al. 2010; Chaurasia et al. 2013] greatly facilitate the production of realistic virtual walkthroughs from a small number of photographs. However, multi-view datasets are typically captured under fixed lighting, severely restricting their utility in applications such as games or movies – where lighting must often be manipulated. We introduce an algorithm to remove lighting in such multi-view datasets of outdoors scenes with cast shadows, with all photos taken in the same lighting condition. We focus on wide-baseline datasets for easy capture, with a typical density of e.g., a photo per meter for a facade. Our solution decomposes each image into reflectance and shading layers, and creates a representation of movable cast shadows, allowing us to change lighting in the input images. We thus take a step towards overcoming the limitation of fixed lighting in previous image-based techniques, e.g., Image-Based Rendering (IBR). With our approach we can plausibly modify lighting in these methods, without requiring input photos with the new illumination.



(a) Input image

(b) Relighting +30 minutes

(c) Reflectance

(d) Shading

Fig. 1: Our algorithm enables relighting with moving cast shadows from multi-view datasets (a,b). To do so, we separate each image into its reflectance and shading components (c,d).

Each photograph in a multi-view dataset results from complex interactions between geometry, lighting and materials in the scene. Decomposing such images into intrinsic layers (i.e., reflectance and shading), is a hard, ill-posed problem since we have incomplete and inaccurate geometry, and lighting and materials are unknown. Previous solutions achieve impressive results for many specific subproblems, but are not necessarily adapted to automated treatment of multi-view datasets reconstructed with multi-view stereo, especially in the presence of *cast shadows*. For example, previous intrinsic image approaches can require manual intervention, special

[1] current affiliation Universidad Rey Juan Carlos
[2] current affiliation ETH Zurich

0730-0301/2015/15-ARTxx $15.00

http://doi.acm.org/10.1145/2756549

hardware for capture, or restricting assumptions on colored lighting; recent learning-based shadow detectors may not always provide consistently accurate results, and previous inverse rendering methods require pixel-accurate geometry which cannot be automatically created using multi-view stereo. Our datasets can have 30-100 photographs allowing image-based navigation over a sufficient distance for image-based-rendering applications. We thus aim for an automatic method that scales to multi-view datasets while producing consistent quality results over all views under outdoor lighting.

Our method takes the multi-view stereo 3D reconstruction as input; our algorithm is designed to handle the frequent inaccuracies and missing data of such models. The user then specifies the sun direction with two clicks, and we automatically estimate parameters of our image formation model to extract the required reflectance, shading and visibility information.

The first key idea of our approach is to progressively improve the accuracy of the image model parameters with iterative estimation steps, by combining 3D lighting simulation with 2D image optimization.

Our second key idea is to use the image formation model to express reflectance as a function of *discrete* visibility values – 0 for shadow and 1 for light – allowing us to introduce a robust visibility classifier for pairs of points in a scene.

Our method starts by finding a first estimate of sun and environment lighting parameters, as well as visibility to the sun. We then find image regions in shadow and in light, implicitly grouping regions of same reflectance. One significant difficulty of outdoors scenes is that they contain complex cast shadow boundaries. It is thus imperative to extract such boundaries as accurately as possible, which we achieve by labeling shadows using our visibility classifier.

We present two main contributions:

—A method that combines multiple images and coarse 3D information to estimate a lighting model of the scene, including incident indirect and sky illumination as well as the color of sunlight. Inspired by the single-image method of [Lalonde et al. 2009; 2011], we first use the input images to automatically synthesize an approximate environment map. We then use the 3D information to perform lighting simulation and deduce the unknown sunlight color.

—A method to compute multi-view intrinsic layers using shadow labelling and propagation. We use our robust visibility classifier in a graph labelling algorithm to assign light/shadow labels to all pixels except those in penumbra. We complete the computed intrinsic layers by propagating visibility to the remaining pixels in each image. The shadow classification is then used to improve the estimate of environment lighting, resulting in more accurate shading, visibility and reflectance layers.

Our automatic multi-view intrinsic decompositions provide high-quality layers of reflectance and shading. The quality of these decompositions is sufficient to allow us to introduce a novel application, namely multi-view relighting with moving cast shadows. We do this by using the intrinsic layers and creating shadow-caster geometry which preserves shadow boundaries even when the 3D model is inaccurately reconstructed. We demonstrate our approach on several multi-view datasets, and show how it can be used to achieve IBR with illumination conditions different from those of the input photos.

2. RELATED WORK

Our work is related to inverse rendering and relighting, intrinsic images and shadow detection; in the interest of brevity we restrict our discussion to recent work most closely related to ours, and cite surveys where possible.

Inverse rendering and Relighting. A comprehensive survey of inverse rendering methods can be found in [Jacobs and Loscos 2006]. Early work [Yu and Malik 1998; Yu et al. 1999; Loscos et al. 1999] required geometry which was of sufficient quality for pixel-accurate cast shadows; this is also true for more recent work [Debevec et al. 2004; Troccoli and Allen 2008]. The geometry was either manually constructed, or scanned with often specialized equipment; similarly involved processes are also used to capture reflectance. In contrast we target the often imprecise and incomplete 3D geometry reconstructed from casual photographs by automatic algorithms.

Karsch et al. [2011] generate plausible renderings of virtual objects in photographs by performing inverse rendering from coarse hand-made geometry and a single image. Xing et al. [2013] propose a similar single-image approach that also accounts for environment lighting in outdoor scenes. The manual steps and required geometry precision for cast shadow removal make these approaches unsuitable for relighting of multi-view datasets. Similarly, Okabe [Okabe et al. 2006] describes a user-assisted approach to recover normals from a single image and relight it. While the normals provide enough information to compute local shading, cast shadows are not considered.

Photo collections have been used for relighting in [Haber et al. 2009] and [Shan et al. 2013]. These approaches require pictures taken under different lighting conditions while our goal is to allow a user to capture a scene once with a single lighting condition, and permit relighting. In particular, while the image formation model of Shan et al. [2013] is similar to ours, their algorithm leverages cloudy pictures to bootstrap reflectance estimation and tends to bake shadows in reflectance in the absence of sufficient lighting variations. The method of Shih et al. [2013] performs lighting transfer by matching a single image to a large database of time-lapses, but cannot treat cast shadows.

Intrinsic images. An alternative to the accurate reflectance model estimation used in inverse rendering is image decomposition into *intrinsic* layers [Barrow and Tenenbaum 1978], typically shading and diffuse albedo (or reflectance). The recent technical report of Barron and Malik [2013a] provides a good review of intrinsic image methods. Automatic single image methods rely on assumptions or classifiers on the statistics of reflectance and shading [Land and McCann 1971; Shen and Yeo 2011; Zhao et al. 2012; Bell et al. 2014]. In particular, most methods assume a sparse or piece-wise constant reflectance and smooth grey illumination – the so-called *Retinex* assumptions. Closer to our work is the method of Garces et al. [2012] who group pixels of similar chrominance to form clusters that are encouraged to share the same reflectance. Ye et al. [2014] extend the method of Zhao et al. [2012] to videos by enforcing temporal coherence. These methods work well on single objects captured in a controlled setup [Grosse et al. 2009] but tend to fail on outdoor scenes where – as noted by [Laffont et al. 2013] – sun, sky and indirect illumination produce a mixture of colored lighting and produce cast shadows. Our method properly handles such cases by explicitly modeling the influence of sky and indirect illumination and by detecting shadow areas.

User-assisted methods [Bousseau et al. 2009; Shen et al. 2011] can handle colored shading but would be cumbersome for the multi-view image sets we target. Recent work has concentrated on multi-image datasets requiring images with multiple lighting conditions, typically from photo-collections [Liu et al. 2008; Laffont et al. 2012]. Similar to inverse rendering methods, the need for multiple lighting conditions makes their usage more complex for the casual capture context we target. This is also true of intrinsic image methods from timelapse sequences (e.g., [Weiss 2001; Sunkavalli et al. 2007].)

A second class of methods use either multiple images or depth acquired either from sensors or reconstruction. Some of these work on a single image e.g., [Lee et al. 2012; Barron and Malik 2013b; Chen and Koltun 2013]. While they improve over previous work, they may sometimes have difficulty removing cast shadows (see comparisons in Sec. 9.5). The work of [Laffont et al. 2013] is closest to ours, but requires special equipment (chrome ball, grey card) and manual selection of parameters. From an algorithmic standpoint, a major difference is that Laffont et al. treat sun visibility as a continuous variable, while we introduce a binary classifier of shadow regions. We found that explicitly estimating binary visibility improves robustness as it prevents this term to absorb errors from other shading terms in a non-physical way. Comparisons to [Laffont et al. 2013] and other methods in Sec. 9.6 show that our approach generally improves the quality of the decompositions.

Shadow detection. Shadow detection and removal have been studied extensively [Sanin et al. 2012] and most methods take a single image as input. Early approaches include automatic methods (e.g., [Finlayson et al. 2004]) which were demonstrated on images of uncluttered scenes with isolated shadows. More recent automated approaches include [Lalonde et al. 2010; Zhu et al. 2010; Panagopoulos et al. 2013; Guo et al. 2012]. Similarly to these methods, our shadow estimation step (Sect. 6) identifies pairs of lit and shadow points sharing the same reflectance. However, existing work detects such pairs using machine learning [Guo et al. 2012] or by approximating shading and reflectance with brightness and hue [Panagopoulos et al. 2013]. In contrast, we rely on multiple images to estimate an environment map and an approximate 3D geometry, which we use to explicitly compute sun, sky and indirect lighting. This additional information provides us with more accurate estimation of reflectance values between pairs of points, which in turn yields more robust shadow classification. Note also that the shadow classifier described in [Panagopoulos et al. 2013] is designed to provide a rough cue of sun visibility suitable for geometry inference, while we aim for finer shadow boundaries to remove the shadows from the image. Other methods [Wu et al. 2007; Shor and Lischinski 2008] require user assistance for each image which would be impractical for the multi-view datasets we target. We provide comparisons in Sec. 9.5.

3. IMAGE MODEL AND ALGORITHM OVERVIEW

The image model we use is central to our method, since it clearly defines the quantities that need to be estimated. The model will also be used to guide the definition of our iterative process to estimate our multi-view intrinsic decomposition.

3.1 Image Model

We use the following image formation model [Laffont et al. 2013]:

$$I \;=\; R\,(\,v_{\text{sun}}\,L_{\text{sun}}\,\cos(\omega_{\text{sun}}) + S_{\text{sky}} + S_{\text{ind}}), \qquad (1)$$

I is the observed radiance (i.e., pixel value), R is the diffuse reflectance of the corresponding 3D point, L_{sun} is the radiance of the sun, v_{sun} is the sun visibility from the point, ω_{sun} is the angle between the normal n and the direction θ_{sun} to the sun, S_{sky} is the radiance of the visible portion of the sky integrated over the hemisphere Ω centered at n, and S_{ind} is the indirect irradiance integrated over Ω, but excluding the sky. For all cosines we take $\max(0, \cos)$ in practice; all values are RGB except for the cosine. We implicitly assume that R is diffuse.

Using Eq. 1, we can also write reflectance R as a function of visibility:

$$R(v_{\text{sun}}) = \frac{I}{(S_{\text{ind}} + S_{\text{sky}} + v_{\text{sun}}\,L_{\text{sun}}\,\cos(\omega_{\text{sun}}))} \qquad (2)$$

In some cases it is convenient to group sky and indirect lighting into a single *environment shading* term S_{env} and write $S_{\text{sun}} = L_{\text{sun}}\,\cos(\omega_{\text{sun}})$, giving a simpler expression:

$$I \;=\; R\,(\,v_{\text{sun}}S_{\text{sun}} + S_{\text{env}}\,) \qquad (3)$$

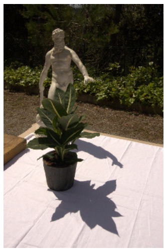
3.2 Input

Our input is a set of linearized raw 12-bit/channel photographs of the scene, captured from different viewpoints at the same time of day and with same exposure. We use Autodesk Recap360 (http://recap360.autodesk.com) for all 3D reconstructions, taking the vertices of the reconstructed mesh as a point cloud. The quality of the meshes is quite high overall with some residual noise for buildings, but often very approximate for structures such as vegetation etc. Such methods also have difficulty reconstructing silhouettes and fine structures. Alternative methods (e.g., structure from motion [Snavely et al. 2006] followed by reconstruction [Goesele et al. 2007; Furukawa and Ponce 2007; Pons et al. 2007]) provide similar quality. In what follows we use the term *proxy* to refer to this – typically incomplete and inaccurate – 3D model.

Our method requires the sun direction θ_{sun}. Automatic methods [Panagopoulos et al. 2013] can be used, however, we prefer to use a simple manual step, which is performed just after reconstruction and guarantees high-quality results. To determine the direction of the sun, a colored version of the point cloud is presented to the user. Each point is assigned the median value of pixels in all images in which this 3D point is visible. The user clicks on a point in shadow and the corresponding 3D point which casts it, allowing the sun direction to be estimated. This simple process is shown in the accompanying video.

3.3 Estimating image-model quantities.

Our algorithm has four main steps, shown in Fig. 2. In each step we compute estimates of the quantities of Eq. 1, which are progressively more accurate. To compute a *reflectance* layer R, we estimate shading S_{tot}, and divide the input image to obtain R; this is performed in Steps 2-4 and the result shown in Fig. 2. In contrast with most previous work, our input contains strong cast shadows. Our goal is to obtain results of sufficient quality to perform relighting: this requires a reflectance layer free of shadow and other residues, as well a good estimate of shadow boundaries, environment shading.



A guiding principle of our approach is that we prefer the explanation of a given scene that favors a smaller number of reflectances, following previous work [Omer and Werman 2004; Barron and Malik 2013b; Laffont et al. 2013]. Consider the scene shown in the inset. There are two explanations for the dark areas on tablecloth: a shadow cast by the statue and plant or blobs painted in grey. Our approach favors the hypothesis with fewer reflectances, which explains the

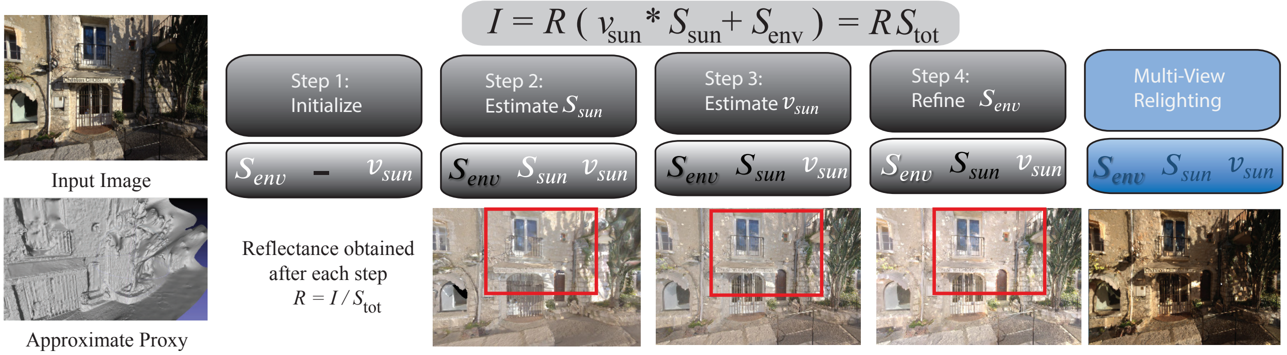

$I = R\,(\,v_{\text{sun}} * S_{\text{sun}} + S_{\text{env}}\,) = R\,S_{\text{tot}}$
Input Image
Approximate Proxy
Step 1: Initialize
S_{env} – v_{sun}
Step 2: Estimate S_{sun}
S_{env} S_{sun} v_{sun}
Step 3: Estimate v_{sun}
S_{env} S_{sun} v_{sun}
Step 4: Refine S_{env}
S_{env} S_{sun} v_{sun}
Multi-View Relighting
S_{env} S_{sun} v_{sun}
Reflectance obtained after each step $R = I\,/\,S_{\text{tot}}$

Fig. 2: Our input are images (top left), an approximate 3D model (proxy) (lower left) and user-supplied sun direction. We use the image formation model (top) and estimate progressively better approximations to each of its parameters. In white, quantities estimated in a given step; quantities in black are fixed at that stage. Step 1: given the proxy we build a sky environment map and compute a first estimate of v_{sun} and S_{env} by ray-tracing the inaccurate 3D model and sky map. Step 2: we refine v_{sun} and estimate S_{sun} using luminance and chromaticity. Step 3: given first estimates of all quantities we perform a graph labelling to further refine v_{sun}. In Step 4 we refine S_{env}, and v_{sun} in penumbra, using the more accurate shadow boundaries now available. Reflectance is estimated at steps 2-4, and we clearly see how the result is progressively improved. Far right: during multi-view relighting, reflectance is fixed, and we can manipulate quantities in blue, resulting in a relit image (lower right; compare to top left).

image as a shadow over a uniformly white tablecloth. Throughout the four steps of our approach, we enforce this hypothesis by finding same-reflectance *pairs* between regions or points in light and shadow, inspired by previous work [Panagopoulos et al. 2013; Guo et al. 2011].

The key novelties of our approach are the automatic estimation of the parameters of Eq. 2, and the introduction of a robust shadow classifier using this information, see Sec. 6. Put together, these encourage the choice of the correct visibility configuration which finds same reflectance regions and implicitly connects (or merges) them via the pairs, thus enforcing the hypothesis.

The fours steps are illustrated in Fig. 2: In Step 1, we find initial values for S_{env} and v_{sun}; in Step 2 we estimate S_{sun}, in Step 3 we obtain accurate shadow boundaries by refining the estimate of v_{sun} and in Step 4 we refine the estimation of S_{env}. The estimated reflectance improves significantly at each step.

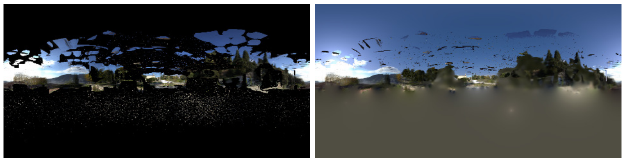


Fig. 3: Left: the partial environment map. Right: completed synthesized environment map.

4. ESTIMATION OF S_{SKY} AND S_{IND}

To compute S_{sky} and S_{ind}, we first automatically compute an environment map to represent light coming from the sky and unreconstructed surfaces[1]. We project all pixels of the input pictures that are not covered by the reconstructed geometry into this map. Fig. 3 shows such a partial environment map where holes correspond to directions either not captured in the input photographs or directions corresponding to rays that intersect the proxy.

We apply a simple color-based sky detector to determine which pixels above the horizon in the map are sky and which are distant objects. More involved approaches [Tao et al. 2009] could be used, but our approach sufficed in all our examples. The horizon is the main horizontal plane of the proxy. The visible portions of the sky give us strong indications on the atmospheric conditions at the time of capture. Inspired by Lalonde et al. [2009; 2011], we estimate the missing sky pixels by fitting the parametric sky model of Perez et al. [1993] from the partial environment map. This model expresses for any direction p the sky color *relative* to the color at zenith as a function of the angle θ_p between p and the zenith, the angle γ_p between p and the sun direction, and the turbidity t that varies with weather conditions [Preetham et al. 1999; Lalonde et al. 2009]. Since the color at zenith is itself an unknown, we need to recover a global per-channel scaling factor to obtain absolute values. We estimate the turbidity t of the sky model f and the scaling factor $\mathbf{k}$ by minimizing

$$\operatorname*{argmin}_{t,\mathbf{k}} \sum_{p\in\mathcal{P}} (\mathbf{k} f(\theta_p, \gamma_p, t) - \mathbf{A}_p) \tag{4}$$

where $\mathcal{P}$ denotes the set of known pixels in the environment map $\mathbf{A}$. We solve this non-linear optimization with the simplex search algorithm (`fminsearch` in Matlab). At each iteration, the search algorithm generates a new value of the turbidity t that we use to update f, and then $\mathbf{k}$ from the new sky values by solving a linear system. We initialize the optimization by setting $t = 3.5$, which corresponds to the turbidity of a clear sky [Preetham et al. 1999]. We fill holes below the horizon line by diffusing color from nearby pixels.

Similarly to Laffont et al. [2013], we compute S_{ind} and S_{sky} by integrating the indirect and sky incoming radiance using ray-tracing. For each 3D point, we cast a set of rays over the hemisphere centered on the point normal. Rays that intersect the sky part of the environment map contribute to S_{sky}, while rays that intersect the

[1] We described a preliminary version of the environment map computation in Chapter 4 of [Laffont 2012].

proxy geometry or the non-sky pixels of the environment map contribute to S_{ind}. We estimate the radiance coming from the proxy geometry by gathering for each vertex the radiance in the images where this vertex appears. We assign the median of the gathered values as the approximate diffuse radiance of the vertex. Given the low frequency nature of these quantities, our approximations are generally sufficient. However, the non-diffuse nature of real surfaces and errors in reconstruction can result in overestimation of indirect light. We thus introduce an approximate attenuation factor which compensates for such errors by scaling with the cosine of the normal of the contributing surface when gathering at each point. Details of the implementation are given in the supplemental material.

The ray-tracing step also provides approximate visibility $\tilde{v}$ towards the sun at each point, with respect to the proxy. The boundaries defined by $\tilde{v}$ can be quite approximate however, as shown in Fig. 4(b). We improve the estimate of v_{sun} in Step 3 (Sec. 6).

5. ESTIMATION OF SUN COLOR L_{SUN}

Now that we have computed illumination from the sky and indirect transfer at all 3D points, we can estimate L_{sun} using Eq. 1 and a pair of points with same reflectance and different visibility. Given two points p_1 and p_2 with the same reflectance, with one in shadow and the other in light, we can compute L_{sun}:

$$L_{\text{sun}} = \frac{I_1 * (S_{\text{sky}2} + S_{\text{ind}2}) - I_2 * (S_{\text{sky}1} + S_{\text{ind}1})}{I_2 * v_{\text{sun}1} * \cos(\omega_1) - I_1 * v_{\text{sun}2} * \cos(\omega_2)} \qquad (5)$$

All quantities for sun, sky and indirect are denoted with appropriate subscripts.

The main difficulty in using this formulation is that we do not yet have accurate reflectance and visibility necessary to find a suitable pair of points. While single-image intrinsic decomposition methods could be used to initialize the reflectance, most existing algorithms are challenged by outdoor scenes with hard shadows that break the Retinex assumptions of a smooth monochrome shading. We conducted preliminary experiments with the Retinex implementation of [Grosse et al. 2009], which confirmed that this algorithm does not remove hard shadows on our scenes. As a result, our calibration algorithm was unable to find pairs of points sharing the same reflectance across shadow boundaries.

Instead of using Retinex, we found it sufficient at this stage to approximate reflectance with the image chrominance and shading with luminance, which we combine with the proxy-based visibility $\tilde{v}$ for a conservative estimate of shadow regions. More precisely, we first perform a K-means clustering on luminance. We found that image histograms typically contain two "extreme" clusters (dark and light, most often corresponding to shadow and light); to better separate intermediate values two additional clusters are required. We thus used K= 4 in our implementation. For each cluster we compute the ratio of the number of points inside the proxy shadow $\tilde{v}$ to the number of points outside. We classify a cluster in shadow if its ratio is lower than the average ratio in the image. We then intersect the value of $\tilde{v}$ (Fig. 4(b)) with the classification of each pixel (Fig. 4(c)), resulting in very confident regions of shadow albeit covering only a limited number of pixels in the image Fig. 4(d). Given this visibility estimate, we sample the shadow boundary regularly and for each sample we detect a lit (resp. shadowed) point away from the penumbra by walking in the two directions perpendicular to the boundary and selecting the pixel with the highest (resp. lowest) luminance and a similar chrominance. In our implementation we stop the walk after 30 pixels in each direction and reject the samples for which no pixels with similar chrominance are found. We also reject the sample if we cross a depth or normal discontinuity along the walk, identified with a Canny filter over the depth and normal map rendered from the proxy. Despite the inaccuracies of the proxy, in our tests only 15% of the selected pairs did not share the same reflectance, or did not cross a shadow boundary. Taken over the entire multi-view dataset, the selected pairs provide multiple estimators for L_{sun}, using Eq. 5 (Fig. 4(e)). We finally compute a robust estimate of L_{sun} as the median of the solutions given by all pairs.

We found that performing the median filter in each RGB channel separately gives the best results. This sparse set of pairs is approximate but sufficient for the calibration task. The later estimation of more accurate visibility boundaries will allow us to find a more reliable and denser set of light/shadow pairs and thus refine the estimate of environment lighting.

At this stage, we have an estimate of all quantities of Eq. 1, namely ω_{sun}, S_{sky}, S_{ind} and L_{sun}; the estimate of $v_{\text{sun}} \approx \tilde{v}$ however is approximate. If we compute reflectance at each 3D point, we obtain approximate results that can have large regions of error (see leftmost image in Fig. 2).

6. ESTIMATING ACCURATE CAST SHADOWS AND INTRINSIC LAYERS

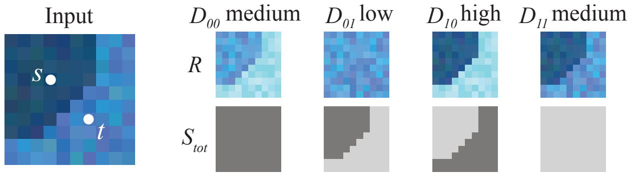
To compute residue-free reflectance layers for each image, we need to refine the accuracy of shadow boundaries and thus v_{sun}. We do this using a graph labeling approach, giving a binary label of shadow or light to all pixels, except those in penumbra. We assign a continuous visibility label to penumbra separately using matting (Sec. 6.2).

6.1 Shadow Labeling

The intuition behind our approach is to find the set of visibility labels that make most points share a similar reflectance, as explained earlier (Sec. 3.3). Consider two points s and t with visibility i and j respectively. Using Eq. 2 we compute the difference between their reflectances as:

$$D_{ij} \;=\; |R_s(i) - R_t(j)|. \qquad (6)$$

Since $i, j \in \{0, 1\}$ we obtain four possible values of D_{ij}. A small value provides us with a strong evidence that s and t share the same reflectance under the corresponding visibility hypothesis. We illustrate this strategy with the following toy examples:



The diagram below shows the case where s is in shadow and t is in light, both on a patch of roughly constant reflectance. Consider the case in the second column, which is the correct configuration: the two points receive a similar reflectance, which makes D_{01} small. In contrast, D_{10} is large, since the incorrect visibility assumptions "pull apart" $R_s(1)$ from $R_t(0)$; see Eq. 2. The two points also receive different reflectances when assigned the same visibility, i.e. D_{00} and D_{11} are larger than D_{01}, although smaller than D_{10}. We can thus concentrate on comparing D_{10} and D_{01}; this provides a robust indicator of the correct visibility labels for

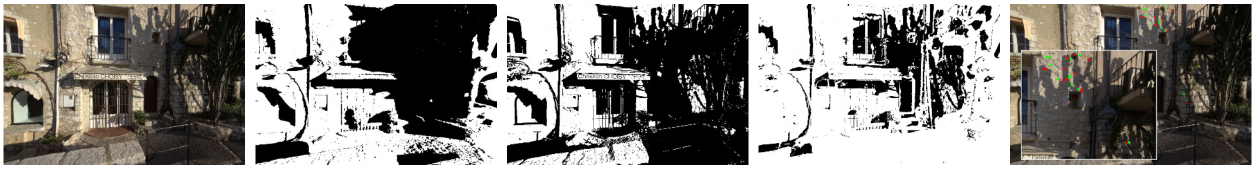
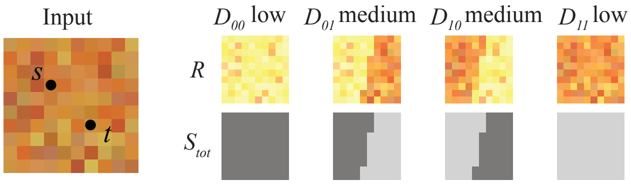

(a) (b) (c) (d) (e)

Fig. 4: The consecutive steps of the algorithm to determine L_{sun} for the "Street" scene. (a) Input image (b) shadow from inaccurate 3D model: the proxy overestimates the geometry of the cactus and creates a "blob" shadow (c) K-means intensity estimation: some black areas are not shadows (d) intersection of (b) and (c): a more reliable subset of shadows are found, which are used in (e) to find pairs used to calibrate the sun.

the pair, under the assumption that the two points share a similar reflectance. Importantly, this information is directional, i.e., if s is in shadow, then we have a strong indication that t is in light.



The second diagram above shows a configuration where s and t have the same label (both in light in this case; both in shadow can be treated symmetrically), also with the same reflectance. Here we can distinguish clearly between same label cases (D_{00} and D_{11}) which give a similar reflectance to s and t compared to the different-label cases. However, we cannot distinguish between the light/light or shadow/shadow case since they both make the two points have a similar reflectance. Pairs of points sharing the same visibility are thus somewhat less informative than pairs of points with different visibility. Both cases however provide reliable information which we use for shadow classification. Finally, points having different reflectances result in high D_{ij} under all four labeling configurations.

We next define an energy that is minimized by the label configuration best explaining the same reflectance hypotheses. Specifically, we detect the pairs of points likely to have the same reflectance and different visibility and use this directional information to initialize the labels at a few confident points (Fig. 5(a)). We then connect these points to their immediate neighbors and to other points with same reflectance and visibility, which allows us to propagate the labeling over the entire image (Fig. 5(b)). We express this approach as a Markov Random Field (MRF) problem over a graph [Szeliski 2010; Kolmogorov 2006], where each node corresponds to a point s with label $x_s \in \{0,1\}$ and each edge (s,t) connects a point s to another point t. Noting $\mathcal{X}$ the set of all labels x_i of all nodes, we have

$$\underset{\mathcal{X}}{\operatorname{argmin}} \sum_{s \in V} \phi_s(x_s) + \sum_{(s,t)\in\mathcal{E}} \phi_{s,t}(x_s, x_t), \quad x_i \in \{0,1\}. \tag{7}$$

V denotes the set of nodes, $\mathcal{E}$ is the set of edges, $\phi_s(x_s)$ is the unary potential deduced from points with same reflectance and different visibility, and $\phi_{s,t}$ is the pairwise potential that favors the propagation of the labels. We detail the computation of the unary and pairwise potentials later in this section.

To solve this optimization, we could naively connect all pixels to all others, and perform the minimization on the resulting graph. This is both inefficient and numerically unstable. We thus apply mean-shift clustering [Comaniciu and Meer 2002] in (L, a, b, x, y) space to segment the image into small regions where we can safely assume uniform reflectance and visibility, simplifying the problem and reducing noise (see Fig. 6). We use bandwidth parameters of 5 in space and 1 in color and a minimum region area of 50 pixels, which results in around 6000 clusters for an image of size 1000 × 700 pixels. The values for $R(0)$ and $R(1)$ for a cluster are computed as the median values for all 3D points projected onto the cluster, except for points in a 3-pixel wide boundary around each cluster.

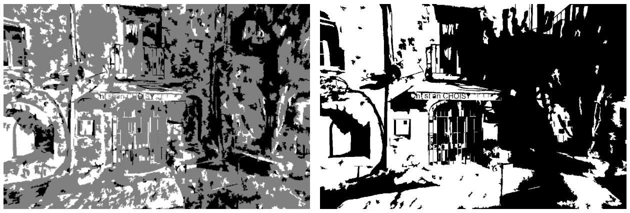
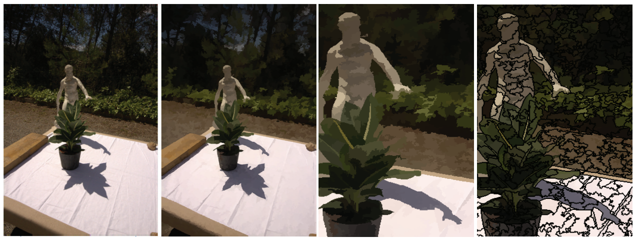

(a) Initialization (b) Final labeling

Fig. 5: (a) Initial labels from unary term, white is in light, black in shadow and grey undefined. (b) Final labels after convergence.

We solve our problem using a publicly available implementation of [Kolmogorov 2006][2]. The potentials take values $l_1 = 1$, $l_0 = 0$ when strongly encouraging one hypothesis over the other, $l_p = 0.8, l_{np} = 0.2$ for the case when one hypothesis is moderately preferred over another ("non-preferred") and $l_{eq} = 0.5$ when both hypothesis are equally encouraged.

Unary Potential. As many binary labeling problems, a good initialization is central to obtain a good solution. Given the discussion above, we use pairs of clusters likely to have the same reflectance and different visibility to initialize our unary term. In particular, for a cluster s we find the set $\mathcal{S}$ of k other clusters with the smallest D_{01} and the set $\mathcal{L}$ of k other clusters with the smallest D_{10}. The clusters in $\mathcal{S}$ favor the hypothesis that s is in shadow, while the clusters in $\mathcal{L}$ consider that s is in light. We compute the score of each hypothesis as the sum of the reflectance differences between

[2] http://research.microsoft.com/en-us/downloads/dad6c31e-2c04-471f-b724-ded18bf70fe3/



(a) Input (b) Meanshift clustering (c) Cluster boundaries

Fig. 6: We apply meanshift clustering to decompose the image in small regions of uniform color. We then solve the shadow labeling on a graph of clusters rather than pixels, which reduces the number of unknowns and the impact of noise.

s and the k other clusters

$$H_0 = \sum_{t \in \mathcal{S}} |R_s(0) - R_t(1)| \tag{8}$$

$$H_1 = \sum_{t \in \mathcal{L}} |R_s(1) - R_t(0)| \tag{9}$$

where $R_s(i)$ and $R_t(i)$ are computed with Eq. 2.

If $\frac{H_0}{H_1 + H_0} < \tau_1$, i.e., the hypothesis that s is in shadow is stronger, we set the unary potential to prefer the "in shadow" label:

$$\phi_s(x_s) = \begin{cases} l_0 & \text{for } x_s = 1 \\ l_1 & \text{for } x_s = 0 \end{cases} \tag{10}$$

Conversely, if $\frac{H_1}{H_1 + H_0} < \tau_1$, we set the unary potential to prefer the label "in light":

$$\phi_s(x_s) = \begin{cases} l_1 & \text{for } x_s = 1 \\ l_0 & \text{for } x_s = 0 \end{cases} \tag{11}$$

If neither condition is true we perform a more localized search. We compute two new hypothesis H'_1 and H'_0 in the same manner as Eq. 8, but restrict the k clusters to lie within a *neighborhood* around s. We then check if:

$$\frac{H_1 + H'_1}{H_0 + H_1 + H'_0 + H'_1} < \tau_1 \tag{12}$$

and similarly for the H_0 hypothesis, which can be seen as a more "permissive" hypothesis, since we complement the best global candidates with the best local ones. If one of these conditions is met, we set the potentials the same way as above. If none of the conditions are met, the unary potentials are set to equally prefer either hypothesis:

$$\phi_s(x_s) = l_{eq}, \quad x_s \in \{0, 1\} \tag{13}$$

We used $\tau_1 = 0.1$, corresponding to a 90% confidence level required to make a decision.

Pairwise Interaction Potentials. The goal of our pairwise potentials is to propagate labels between clusters with the same visibility. We first create edges between each cluster s and other clusters with similar reflectance, which we select as the k clusters with smallest D_{00} or D_{11}. For these edges, the values of the potentials are set to strongly encourage the same label to be propagated:

$$\phi_{s,t}(x_s, x_t) = \begin{cases} l_1 & \text{when } x_s = x_t \\ l_0 & \text{when } x_s \neq x_t \end{cases} \tag{14}$$

However, these edges alone are not always sufficient to ensure that the graph forms a single connected component. We prevent isolated components by also connecting each cluster with its immediate neighbors. In the absence of other cues, we define the potential of these weaker connections to encourage clusters with the same color distribution to share the same visibility. We compute the χ^2 histogram distance d_c in Lab space for clusters s and t using the approach described in [Chaurasia et al. 2013]. Clusters s and t are similar for $d_c < \tau_c$; in this case we assume they most probably have the same label:

$$\phi_{s,t}(x_s, x_t) = \begin{cases} l_{np} & \text{when } x_s = x_t \\ l_p & \text{when } x_s \neq x_t \end{cases} \tag{15}$$

If the χ^2 distance is too large however, all potentials are set to equally prefer all possible hypotheses.

$$\phi_{s,t}(x_s, x_t) = l_{eq}, \ \ x_s, x_t \in \{0, 1\} \tag{16}$$

We used $\tau_c = 0.05$ for all our tests, which corresponds to the acceptance probability in the χ^2 test.

At convergence, we obtain accurate shadow boundaries, even though there can be some occasional miss-classifications, e.g., the letters on the store front in Fig. 5(b). Such errors typically occur in small regions that contain few or no 3D points. In the former case, the median reflectance candidates $R(0)$ and $R(1)$ are more likely to be polluted by occasional reprojection errors and specularities, while in the latter case the propagation is solely governed by the χ^2 distance to neighboring regions. Nevertheless, erroneous regions tend to be small in size, and thus do not affect the application to relighting.

6.2 Per-pixel Estimation of v_{sun} and Intrinsic Layers

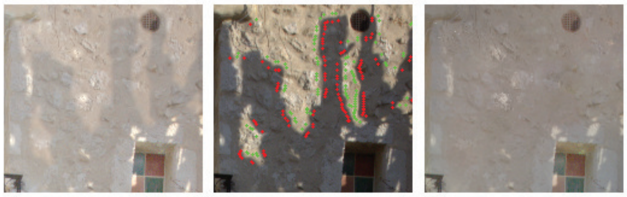
The binary labeling cannot capture soft shadows. We apply Laplacian matting [Levin et al. 2008] to recover continuous variations of visibility in the boundaries between clusters. These correspond to penumbra regions at the frontier of shadow and light clusters, effectively providing a tri-map from the binary shadow mask. We also apply Laplacian matting guided by the input image to propagate the shading values S_{sky}, S_{ind} and S_{sun}, as previously done by Laffont et al. [2012]. We use all 3D points except those in the boundaries between clusters as constraints in this propagation. While these smooth shading layers do not contain shadows, propagating them using the input image as guidance sometimes produces artifacts along shadow boundaries. We reduce these artifacts by excluding a small band along shadow boundaries from the propagation, which we subsequently fill with a color diffusion. The reflectance layer is obtained by dividing the input image by the sum, or total shading S_{tot}

$$S_{\text{tot}} = S_{\text{sky}} + S_{\text{ind}} + v_{\text{sun}} S_{\text{sun}}. \tag{17}$$

The classifier can occasionally miss very fine shadow structures which are however captured by the clusters; we also propagate visibility in the boundary regions between clusters, which generally improves the visual quality for relighting (see Sec. 9.9).

7. REFINING ENVIRONMENT SHADING AND REFLECTANCE ESTIMATION

The quality of the intrinsic layers obtained so far is limited by the accuracy of the different radiometric quantities computed. In particular, the success of using Eq. 2 to compute R is dependent on the approximations in our estimations of S_{env}, L_{sun} and v_{sun}. As



(a) Reflectance before correction

(b) Same-reflectance pairs across shadow boundary

(c) Reflectance after correction

Fig. 7: (a) The reflectance is discontinuous across the shadow boundary due to incorrect estimation of shading. (b) Pairs chosen as constraints to impose the same reflectance on both sides of the boundaries. (c) Corrected reflectance after optimization.

we see in Fig. 7(a), the currently estimated values leave a visible residue in the reflectance layer, which should be continuous (Fig. 7(c)). This discontinuity occurs because the values of S_{env} and L_{sun} were computed using the incomplete and inaccurate 3D reconstruction, and are thus approximate.

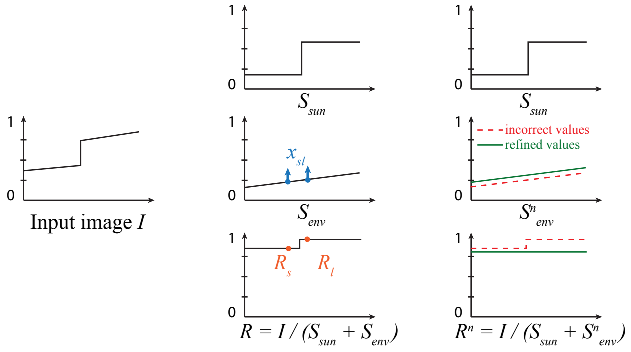


Fig. 8: 1D visualization of S_{env} refinement. Small errors in our estimates of L_{sun} and S_{env} can prevent the reflectance to be continuous across shadow boundaries (middle). We detect pairs of points with similar reflectance on each side of the boundary (orange dots) and compute a local offset of S_{env} (blue) that makes the two reflectances equal (right).

We illustrate this in Fig. 8 where we show a plot of image intensity across a shadow boundary, with the shadow region on the left. In the middle column, we see the decomposition of the image into reflectance (R_s in shadow and R_l in light), and shading, composed of S_{sun} and S_{env}. We will refine the value of shading so that R_s becomes equal to R_l, by adding an offset to S_{env}. We correct S_{env} since it is a continuous quantity over the shadow boundary. Specifically we apply an offset x_{sl} to S_{env} on both sides of the shadow boundary so that R_s becomes equal to R_l (Fig. 8, right).

We first find a dense set of same reflectance light/shadow pixel pairs along the shadow boundaries, Fig. 7(b). For each pair, we compute an offset x_{sl} which makes the two reflectances equal. We then smoothly propagate the offsets to all pixels while preserving the variations of S_{env}, yielding the refined layer S^n_{env}, Fig. 7(c). The values of v_{sun} in penumbra were determined by image-driven propagation, which can sometimes result in high-frequency inaccuracies of v_{sun}. These cannot be captured by the smooth propagation, and we thus treat these pixels separately by correcting the v_{sun} layer.

Implementation details of the above steps for S_{env} refinement are described in the supplemental material.

8. APPLICATION: MULTI-VIEW RELIGHTING WITH MOVING CAST SHADOWS

The automatic nature of the process and the high quality of the intrinsic layers for reflectance R, shading S_{sun}, S_{env} and visibility v_{sun} allow us to introduce the novel application of multi-view relighting.

Creating Shadow Receiver and Caster Geometry. Recall that shadows cast from the proxy are not accurate enough for relighting, since they do not correspond well to shadow boundaries in the image (see Fig. 4(b)). We approximate moving cast shadows by creating a geometric representation of a caster from the shadows in the original image. While creating caster geometry is related to shape-from-shadow techniques [Savarese et al. 2007], such methods require shadows from multiple light sources. In our case, we only have shadows from a single position of the sun. We thus design an approximate algorithm that (a) preserves the original shadow boundaries in the input image as much as possible and (b) allows some motion of the sun.

We first reconstruct the receiver geometry by assigning to each pixel the depth value of the closest projected 3D point. We found that the resulting depth map, while approximate, results in plausible shadows that we can composite over the reflectance image. We then estimate the geometry of the caster such that it produces shadows that match the shadow boundaries in the original images. We identify the shadow boundaries from the shadow classification layer (e.g., Fig. 5, right) as well as from the propagated v_{sun} layer that sometimes capture fine details lost by the binary classifier (Fig. 21, Sec. 9.9). We consider pixels to be in shadow if pixel p is classified as shadow in the former, or if $v_{\text{sun}}(p) < \tau_s$. We used $\tau_s = 0.8$ for all our results. To estimate a 3D caster position at each shadow pixel, we shoot rays in the direction of the sun θ_{sun} and record the distance of the closest intersection with the 3D proxy. Pixels for which the ray does not intersect the proxy receive the distance of the nearest valid pixel. We triangulate the shadow pixels in image space to create a mesh that we lift in the direction of the sun using the recorded distance. Fig. 9 illustrates the resulting 2.5D caster which re-creates the shadow boundary in the image.

Incorrect reconstruction and numerical imprecision can result in erroneous triangles that partly re-project on lit pixels. We remove such triangles by visiting all pixels in light and casting rays in the sun direction. If a triangle of the caster mesh is intersected by more than ϵ such rays, it is removed. We used $\epsilon = 3$ for all our results. Our shadow labeling also sometimes mis-classifies pixels as shadow in small regions. To filter these errors we cluster the pixels in shadow and remove small clusters (less than 100 pixels) and clusters for which less than 30% of the pixels yield and intersection with the proxy. We adjust the reflectance of such pixels to bring them in light using the v_{sun} layer.

Moving Shadows and Adjusting Shading. To move shadows, we simply change the sun direction θ_{sun} and cast rays from each pixel in that direction. We compute intersections against the caster using the Intel *embree* library, which provides interactive feedback for the images shown here (see also the accompanying video). Our caster geometry only reproduces the shadows captured in the image. As a result, discontinuities can appear when the shadow move away from the border. We complete the missing shadow in these areas using the shadow of the proxy geometry. Finally we apply a

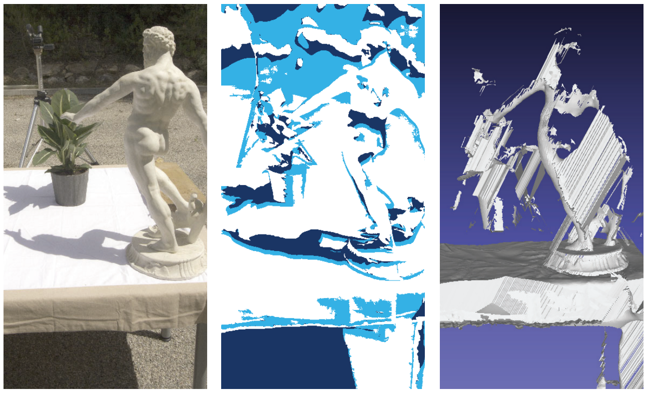


Fig. 9: Left: Input image. Middle: detected shadow pixels in blue, shadow from the proxy in dark blue. Right: The caster mesh generated from these shadow pixels.

small Gaussian blur on the new shadow layer to mimic soft shadows and to fill small holes caused by disconnected triangles in the caster mesh.

We render an approximate effect of illumination changes by adjusting the sun shading intensity according to a cosine factor with respect to elevation and the horizontal plane, and shifting S_{sky} towards red in the morning and afternoon. We also diminish S_{ind} with a similar amount to maintain the illusion of shading change. Finally, we detect sky pixels and change their color near the horizon.

9. RESULTS AND COMPARISONS

We first present results of our decomposition algorithm for a number of real-world scenes, as well as application to relighting. We then provide five evaluations of our method: 1) A ground-truth quantitative evaluation of our algorithm and comparison to [Laffont et al. 2013]; 2) A ground-truth comparison of our synthetic relighting with real photographs taken at different times. 3) A visual comparison of our algorithm with state-of-the art intrinsic image methods and shadow classifiers; 4) A comparison of our automatic sunlight calibration and environment map estimation with the method of Laffont et al. [2013], which uses a grey card and chrome ball; 5) An evaluation of the robustness of our approach to decreasing number of input images.

In supplemental material, floating point versions of all layers of our decompositions are provided; however, different tone mapping had to be applied to each image to allow visibility of the results in this document.

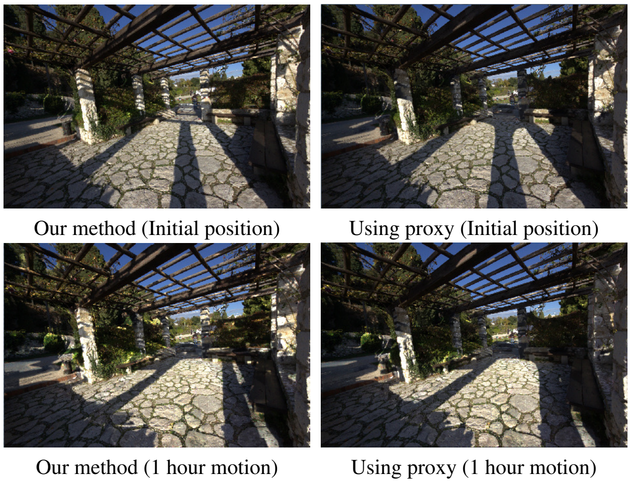
9.1 Intrinsic Decomposition Results

We present results on a variety of scenes. We show two test scenes with a small number of objects (Plant, Fig. 16) and Toys (Fig. 10, top row). We also show three natural scenes with buildings, vegetation and thin structures (Fig. 10). In most cases we obtain reflectance layers with little shadow and lighting residue, which are thus suitable for relighting. The shadow classifier and visibility layers are also of high quality overall; occasional miss-classifications are usually in small regions, which can be detected and be removed when moving the shadows for relighting. The strongest errors occur in scenes with poor geometric reconstruction, as is the case in the second and third row of Fig. 10 where large portions of the tree as well as the small wall in front of the scene are missing. Such holes in the geometry affect all the steps of our algorithm, from the computation of indirect lighting to the initialization of shadow regions and sun calibration. As a result, our shadow classifier has moderate success in identifying the shadow over the ground. Finally, the ground is dominated by variations of grey reflectance, which adds to the difficulty of shadow detection as some of these variations are well explained as shadows. The results for all views in each datasets are provided as supplemental material.

9.2 Relighting and Image-Based Rendering

We show results for relighting of the Villa scene in Fig. 1 and Fig. 12, for Street in Fig. 2, for Monastery in Fig. 11 and for Plant in Fig. 15. We performed relighting of up to 2 hours away from the time of capture, after which the shadow starts to break apart (Fig. 12). The maximum time variation that our method can achieve depends on the complexity of the shadow caster and the quality of its 3D reconstruction.

Our relighting approach can be used for image-based rendering (IBR) and changing lighting conditions. In the accompanying video we show an IBR view interpolation and free-viewpoint navigation path in the Villa dataset in which we use the algorithm of [Chaurasia et al. 2013]. We record the path, change lighting conditions and play it back with the new illumination, since all the input images used for IBR have been updated.



Our method (Initial position)

Using proxy (Initial position)

Our method (1 hour motion)

Using proxy (1 hour motion)

Fig. 11: Synthetic relighting. Our method reproduces the initial image well (upper left), and maintains shadow detail during relighting (lower left). In contrast, the proxy shadow looses many fine details (right).

9.3 Ground Truth Decomposition Evaluation

We purchased a model of a scene which has a similar appearance to the real environments we target, with realistic textures for the building, densely foliaged trees and we used a physically-based sky model [Preetham et al. 1999]. We used an in-house path-tracer to render 44 images, which we took as input for our complete pipeline. We also rendered the corresponding layers of reflectance and shading for quantitative comparison.

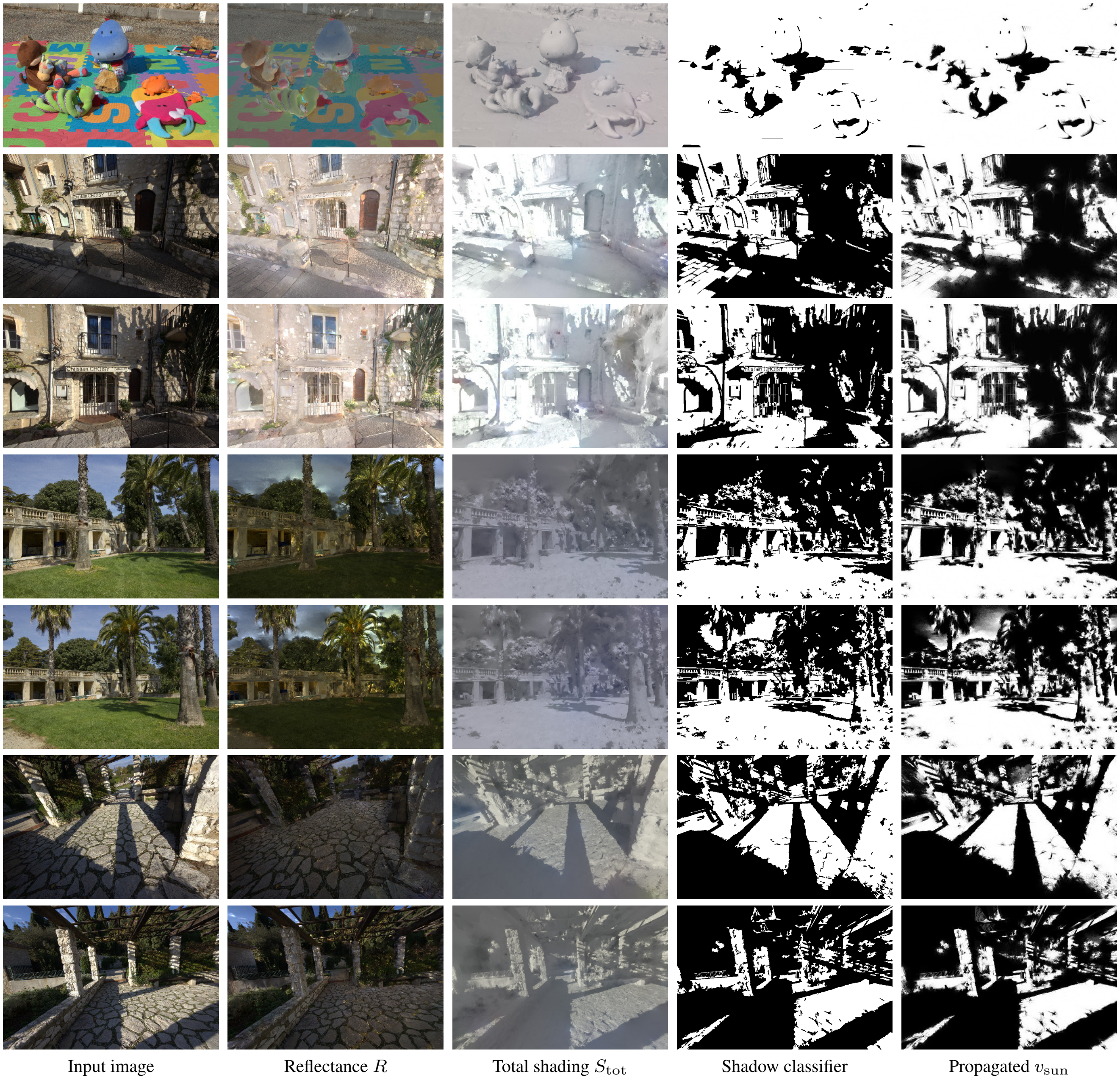
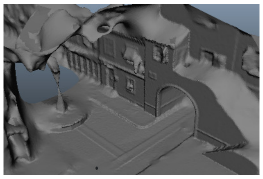
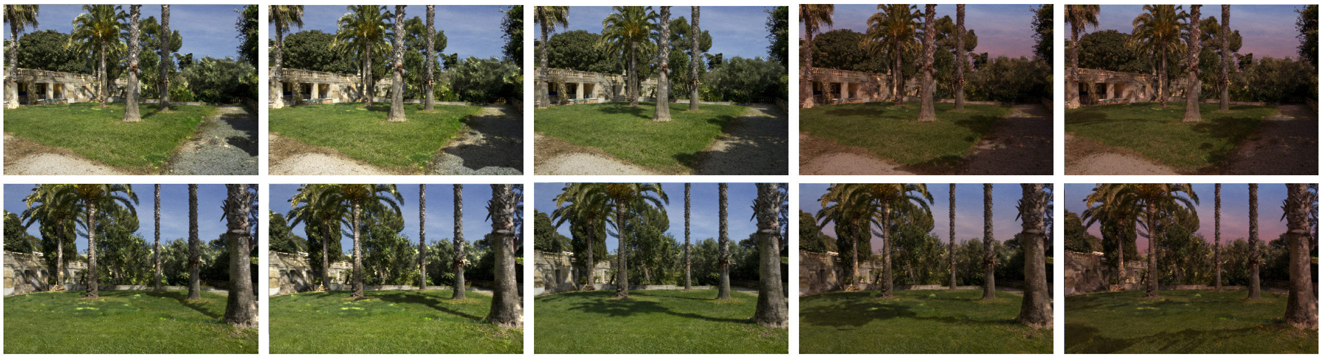


Fig. 10: Our extracted layers on a variety of scenes: toys, urban (top), vegetation (middle), thin structures (bottom).



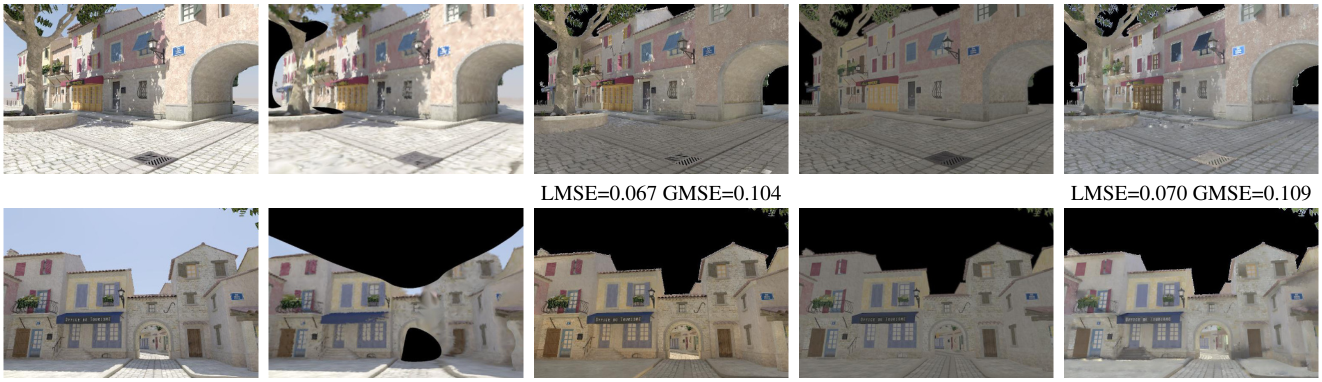
Multi-view stereo has difficulty with synthetic models and textures, and the quality of the reconstruction is poor, as can be seen in the inset; large portions of the tree are not well reconstructed and the overall geometry is coarse and approximate.

Figure 13 provides a visual and quantitative comparison of our reflectance against ground-truth and the result of Laffont et al. [2013]. We selected the parameters of [Laffont et al. 2013] that produce the best decomposition. The two methods yield results of similar quality as measured by the LMSE and GMSE error metric [Grosse et al. 2009]. However, close inspection reveals that most of our error is due to mis-classification of small shadow regions, which yields strong yet localized deviation from ground-truth, while [Laffont et al. 2013] fails to completely remove the shadow of the tree on the wall, which yields a low yet extended erroneous region (Fig. 13, top, far right). This different type of error is due to the fact that the method of Laffont et al. does not explicitly estimate binary visibility and does not refine the estimation of environment shading; our approach yields results more suitable for relighting.



(a) - 2 hours (b) - 1 hour (c) Input image (d) + 1 hour (e) + 2 hours and 30 minutes

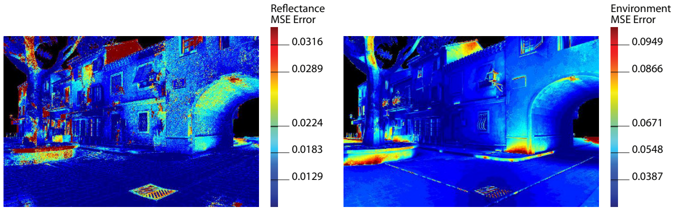
Fig. 12: Relit images for different times of day. While our method can produce drastic motions of the shadows (a-d), the shadow of the central tree breaks apart after a deviation of more than 2 hours (e).



(a) Input image (b) Reconstructed proxy (c) Our reflectance (d) Ground truth reflectance (e) [Laffont et al. 2013]

Fig. 13: Comparison between our method, [Laffont et al. 2013] and ground truth reflectance rendered from a synthetic scene. Our method produces a few strong yet localized errors due to mis-classification of small regions in the shadow of the tree. In contrast, [Laffont et al. 2013] exhibits a low yet extended deviation from ground truth in the shadow region. The two methods are quantitatively similar according to the LMSE and GMSE error metrics.

We provide additional comparisons on real-world scenes in Section 9.5. Figure 14 visualizes our error on reflectance and environment shading. This visualization reveals that a significant part of our error is due to the approximate environment shading, especially in areas where this component dominates sun shading.



(a) Reflectance error (b) S_{env} error

Fig. 14: Visualization of MSE error for our reflectance (a) and refined environment shading S_{env} (b).

9.4 Ground Truth Relighting Evaluation

We captured several lighting conditions for the Plant scene to allow a ground truth comparison. We only used multi-view capture of the central image (i.e., a single lighting condition) for all intrinsic decomposition and relighting computations. We show the results in Fig. 15. We can see that the cast shadow becomes more approximate as we move away from the time of capture used by our algorithm, but the overall appearance is plausible. A slight residue of the original penumbra remains visible in the reflectance, which is due to the non-diffuse nature of the white tablecloth we placed on the table. Since the camera is close to the glossy lobe of this surface, our assumption of diffuse reflectance reaches its limits and our refinement step is not sufficient to fully correct for the remaining errors. Note also that our synthetic shadows have the same color as the shadow in the input image because they are computed from an estimate of the same sun color and sky model. In reality the appearance of the sky changed over time, which explains why the real shadow is darker in some pictures.

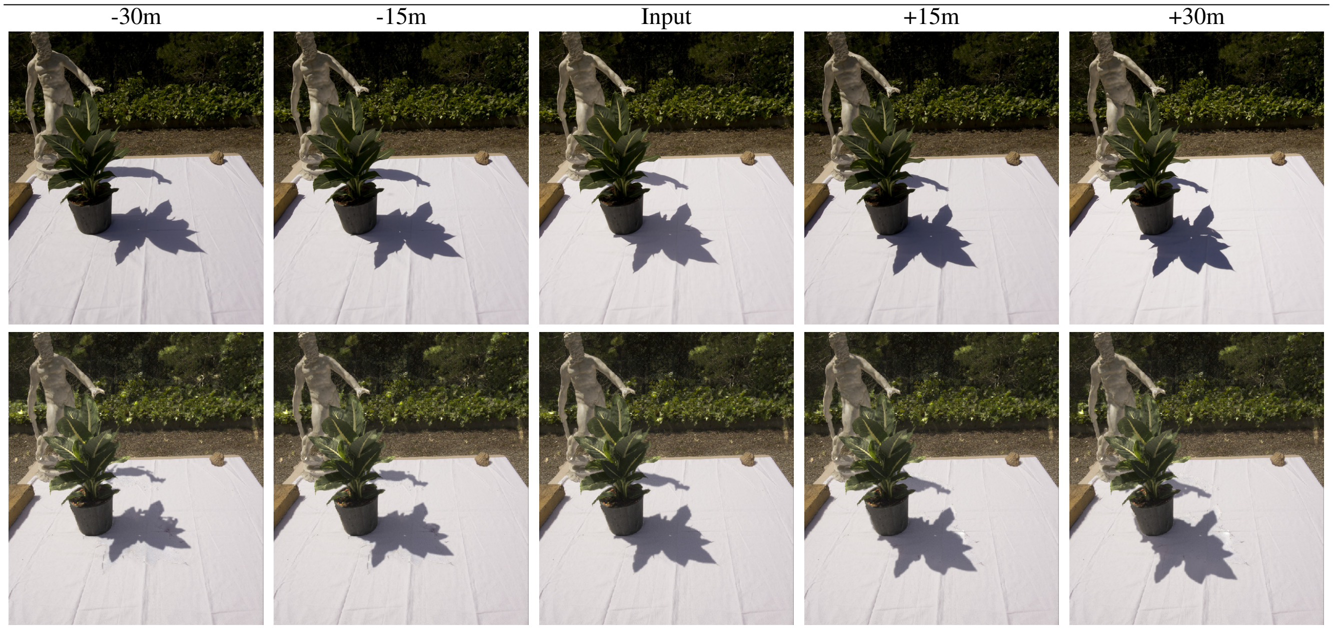


Fig. 15: Above: real photographs taken at different times than those used for the algorithm. Below: relit images using our algorithm.

9.5 Comparisons with Intrinsic Image Algorithms

Our method takes as input multiple images and a 3D reconstruction. For comparisons we focus on recent methods that also combine images and 3D information. Specifically we compare to two single-image methods based on RGB+depth data [Barron and Malik 2013b; Chen and Koltun 2013] and the multi-view method of [Laffont et al. 2013]. From the results presented in these papers, these methods outperform single-image solutions that do not use depth, which are typically derived from the Retinex algorithm. We used the original code of these papers, and reported results to the authors who ensured that we set parameters correctly.

We present two test scenes for comparisons in Fig. 16 and the supplemental document. The first ("Plant") is a simple scene, with a cast shadow on a tablecloth. The proxy reconstruction is of quite high quality except for the plant. From the results we can clearly see that the single image methods are not suited to outdoor scenes with cast shadows, and there is always a residue in the reflectance layer. Our algorithm benefits from the better 3D reconstruction provided by multi-view stereo. The method of [Laffont et al. 2013] also has some residue due to their use of approximate non-binary visibility values that tend to compensate for errors in the estimated shading. By enforcing binary visibility we obtain robust shadow classification, and consequently correcting reflectance across shadow boundaries in a reliable manner, our method produces better results overall.

Recall that, compared to [Laffont et al. 2013], all steps in our approach are automatic, removing the need for the chrome ball, grey card, parameter setting and inpainting steps. Figure 17 provides a comparison between our automatic decomposition and a downgraded version of our algorithm where we used the captured chrome ball and grey card calibration of [Laffont et al. 2013]. Our calibration estimates a sun color of $(2.7, 2.3, 2.4)$ while the grey card yields $(2.7, 2.7, 2.7)$. Our estimated environment map captures the overall color distribution of the sky and ground and results in a reflectance on par with the one obtained with a chrome ball and manual calibration.

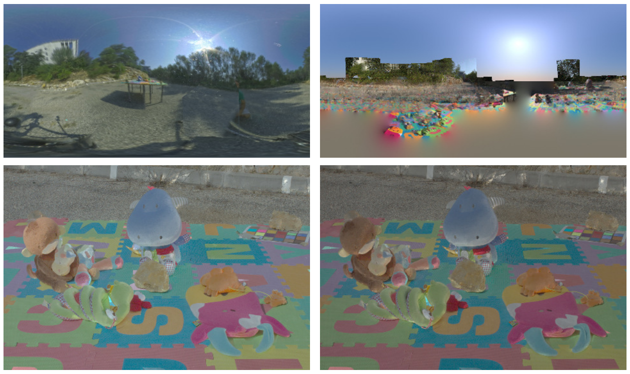


Fig. 17: Comparison using a chrome ball and grey card (left) and our synthesized environment map with automatic calibration (right). Although our environment map misses details on the ground and horizon, it captures the overall color distribution of the ground and sky, yielding reflectance results (lower row) visually similar to the ones obtained with additional information.

Floating point versions for all layers in the figures are provided as supplemental material. We also present additional comparisons for the Toys dataset, and we discuss the different tradeoffs between the artifacts in each approach.

9.6 Comparisons with Shadow Classifier Algorithms

Figure 18 shows a comparison with two single-image shadow classifier methods [Zhu et al. 2010] and [Guo et al. 2011]. Our classifier works well in most cases, and compares favorably to the previous approaches. The method of [Guo et al. 2011] often gives very good

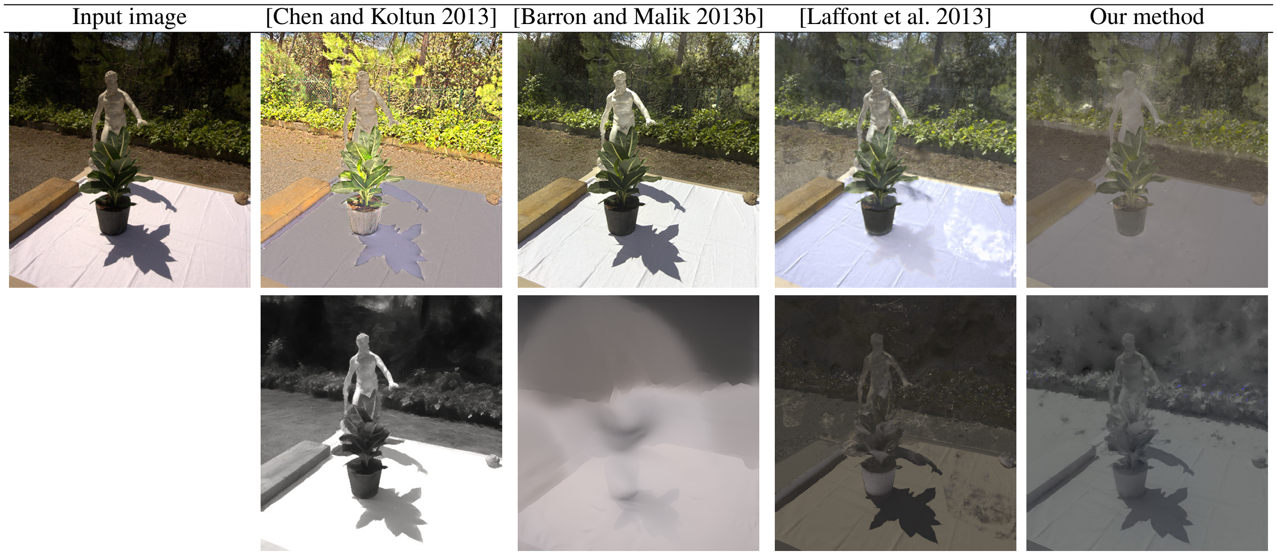


Fig. 16: Comparisons with existing intrinsic image methods, reflectance and shading respectively top and bottom row. Results are shown with scale factor and gamma-correction. Our approach removes the hard shadow, which allows us to subsequently relight the scene.

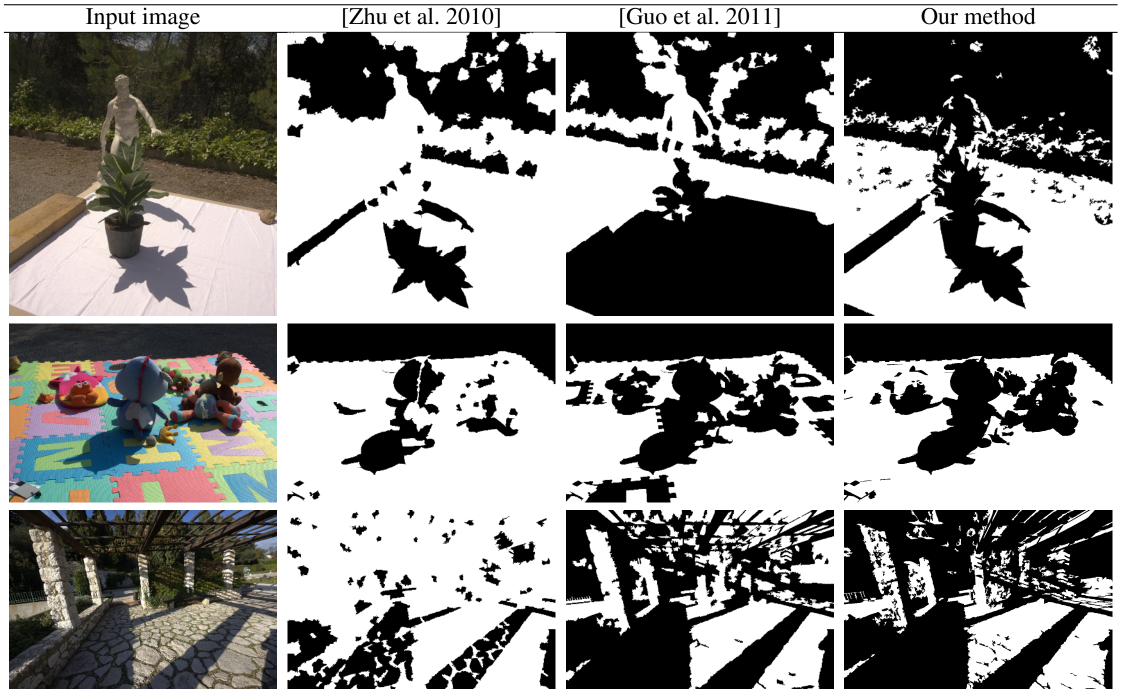


Fig. 18: Comparison with existing shadow classifiers. [Zhu et al. 2010] misses shadow details while [Guo et al. 2011] tends to produce false positives. Our approach leverages 3D information to avoid such errors.

results (last row), but can sometimes reports false positives or fails (top row).

9.7 Impact of Number of Input Images

Table I details the number of images used for each scene, along with the number of vertices of the proxy geometry. As is often the case with multiview stereo reconstruction, we found it easier to capture a large number of images rather than attempting to find the smalest set of images that would be sufficient to run our method. In theory, lowering the number of input images can impact several aspects of our pipeline. First, using fewer images results in fewer samples to estimate the diffuse radiance of the proxy geometry (Sec. 4)

	Street	Monastery	Villa	Statue	Toys
#images	61	61	138	60	73
#proxy	2Mi	2.2Mi	6Mi	4.6Mi	4.6Mi

Table I. : Number of input images and number of vertices of the estimated 3D proxy, for each dataset.

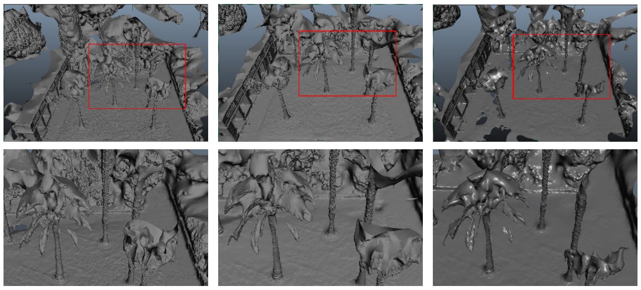


Fig. 19: 3D reconstruction with 138, 68 and 34 views. The reconstruction is increasingly incomplete as we lower the number of images. See Fig. 20 for the corresponding intrinsic decompositions.

and fewer candidate pairs for sun calibration (Sec. 5). More importantly, using fewer images yields a sparser 3D reconstruction which can miss significant parts of the geometry, lowering the quality of our initial estimation of visibility and indirect lighting. A sparser reconstruction also provides fewer point constraints for our shadow labelling algorithm (Sec. 6).

We conducted a small experiment to evaluate the practical impact of the number of input images on the quality of the end decomposition. Figure 20 shows that despite reducing the number of images from 138 to 34 our algorithm produces consistent results. This success is due to the fact that our shadow labeling algorithm leverages image information to identify accurate shadow regions even when the shadow caster is not well reconstructed, as shown in Figure 19.

9.8 Timings

The following table shows the average computation times on a 2.3Ghz E5-2630 PC for each step. Steps 1-3 are implemented in C++, with the exception of the optimization for the sky model which is in Matlab. The entry for Step 4 reports the time to solve the system using an unoptimized Matlab implementation.

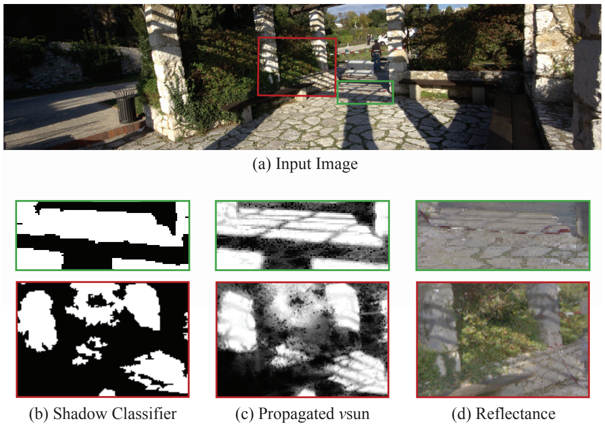
Step	1. Init	2. Estimate S_{sun}*	3. Estimate v_{sun}*	4. Refine S_{env}*
Time	5 min	1 min	3 min	3 min

Table II. : Average timings for each step. Step 1 is total timing for the entire dataset, while for Step 2-4 we report the timing per image marked by *.

9.9 Limitations

The quality of the initial reconstructed model affects all stages of our approach. Some geometry is required in the initialization step, most notably for the computation of S_{ind} and S_{sky}, but also for the L_{sun} estimation. If the geometry is completely incorrect, the initial estimates will not be sufficient for the method to work.

As mentioned previously, while the shadow classifier is overall very successful, it can occasionally miss-classify some regions, especially for fine structures. The propagation of visibility in the clusters can correct some of these (green region in Fig. 21)). In other regions however, the reflectance will contain some residue (red region in Fig. 21).



(a) Input Image

(b) Shadow Classifier

(c) Propagated $v$sun

(d) Reflectance

Fig. 21: Failure case: The fine structures (a few pixels wide) were not captured by the the classifier. In some cases the propagated visibility corrects these (green region), but in others the error remains as a residue in reflectance (red region).

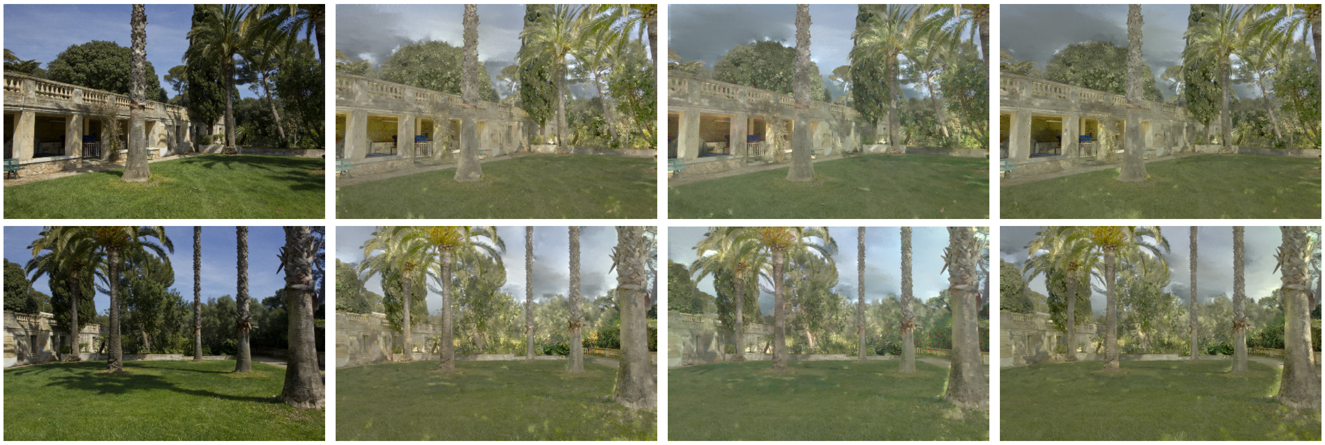
10. CONCLUSION AND DISCUSSIONS

The method we presented is the first to allow automated intrinsic image decomposition for multi-view datasets, providing reflectance, shading and cast shadow layers at a quality level which is suitable for relighting. We are thus able to introduce multi-view relighting, and demonstrate its utility for IBR with changing illumination.

Our approach opens up many possibilities for future work. Currently, our approach assumes outdoors scenes with sunlight and well-defined cast shadows. For scenes with overcast sky, the problem is simpler, since the variation between shadow and light is much smoother. Precise determination of shadow boundaries is thus unnecessary. However, our approach must be extended to handle such soft boundaries, possibly with a new soft shadow classifier approach. We have shown our results using multi-view stereo, but other methods to acquire 3D data and images (e.g., Kinectfusion with RGB information [Nießner et al. 2013]) could be used in our algorithm with no significant changes. However, a single image with depth would probably not provide enough information to initialize S_{sky} and S_{ind}, which are required to bootstrap our progressive estimation of reflectance, shading and shadows.

Another direction for future work is the development of a more complete image formation model that incorporates non-diffuse behavior. This is an exciting fundamental research direction which requires a completely new approach to intrinsic image decomposition.

Apart from IBR, other applications of multi-view relighting are possible, for example in compositing for post-production, where lighting changes often involve a significant amount of manual



Input images

Reflectance with 138 views

Reflectance with 68 views

Reflectance with 34 views

Fig. 20: Decreasing the number of input images does not have a significant impact on the quality of the decomposition. See Fig. 19 for the corresponding 3D reconstruction.

work. In conclusion, by allowing multi-view relighting, our solution takes an important step in making image-based methods a viable alternative for digital content creation.

Acknowledgements

We thank S. Paris for proof-reading the document and providing valuable suggestions, R. Guerchouche for help in capture and preprocessing, L. Robert and E. Gallo for their support and help. This work was funded by an industrial collaboration between Inria and Autodesk. We acknowledge funding from the EU FP7, grant agreement 611089 - project CR-PLAY (www.cr-play.eu), and grant agreement 288914 - project VERVE (www.verveconsortium.eu).

REFERENCES

BARRON, J. AND MALIK, J. 2013a. Shape, illumination, and reflectance from shading. Tech. rep., Berkeley Tech Report.

BARRON, J. T. AND MALIK, J. 2013b. Intrinsic scene properties from a single RGB-D image. *CVPR*.

BARROW, H. G. AND TENENBAUM, J. M. 1978. Recovering intrinsic scene characteristics from images. *Computer Vision Systems 3*, 3–26.

BELL, S., BALA, K., AND SNAVELY, N. 2014. Intrinsic images in the wild. *ACM Transactions on Graphics (Proc. SIGGRAPH) 33,* 4.

BOUSSEAU, A., PARIS, S., AND DURAND, F. 2009. User-assisted intrinsic images. *ACM Trans. Graph. 28,* 5, 1–10.

CHAURASIA, G., DUCHENE, S., SORKINE-HORNUNG, O., AND DRETTAKIS, G. 2013. Depth synthesis and local warps for plausible image-based navigation. *ACM Trans. on Graphics (TOG) 32,* 3, 30:1–30:12.

CHEN, Q. AND KOLTUN, V. 2013. A simple model for intrinsic image decomposition with depth cues. In *ICCV*. IEEE.

COMANICIU, D. AND MEER, P. 2002. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 24,* 5, 603–619.

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. Tech. rep., USC Institute for Creative Technologies.

FINLAYSON, G. D., DREW, M. S., AND LU, C. 2004. Intrinsic images by entropy minimization. In *ECCV*. 582–595.

FURUKAWA, Y. AND PONCE, J. 2007. Accurate, dense, and robust multiview stereopsis. In *Proc. CVPR*.

GARCES, E., MUNOZ, A., LOPEZ-MORENO, J., AND GUTIERREZ, D. 2012. Intrinsic images by clustering. *Computer Graphics Forum (Proc. EGSR) 31,* 4.

GOESELE, M., ACKERMANN, J., FUHRMANN, S., HAUBOLD, C., AND KLOWSKY, R. 2010. Ambient point clouds for view interpolation. *ACM Transactions on Graphics (TOG) 29,* 4, 95.

GOESELE, M., SNAVELY, N., CURLESS, B., HOPPE, H., AND SEITZ, S. M. 2007. Multi-view stereo for community photo collections. In *ICCV*.

GROSSE, R., JOHNSON, M. K., ADELSON, E. H., AND FREEMAN, W. T. 2009. Ground-truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*.

GUO, R., DAI, Q., AND HOIEM, D. 2011. Single-image shadow detection and removal using paired regions. In *CVPR, 2011*. IEEE, 2033–2040.

GUO, R., DAI, Q., AND HOIEM, D. 2012. Paired regions for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence 99*.

HABER, T., FUCHS, C., BEKAER, P., SEIDEL, H.-P., GOESELE, M., AND LENSCH, H. 2009. Relighting objects from image collections. In *CVPR*. 627–634.

JACOBS, K. AND LOSCOS, C. 2006. Classification of illumination methods for mixed reality. In *Computer Graphics Forum*. Vol. 25. Wiley Online Library, 29–51.

KARSCH, K., HEDAU, V., FORSYTH, D., AND HOIEM, D. 2011. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics (TOG) 30,* 6, 157.

KOLMOGOROV, V. 2006. Convergent tree-reweighted message passing for energy minimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 28,* 10, 1568–1583.

LAFFONT, P.-Y. 2012. Intrinsic image decomposition from multiple photographs. Ph.D. thesis, University of Nice Sophia-Antipolis.

LAFFONT, P.-Y., BOUSSEAU, A., AND DRETTAKIS, G. 2013. Rich intrinsic image decomposition of outdoor scenes from multiple views. *IEEE Trans. on Visualization and Computer Graphics 19,* 2, 210–224.

Laffont, P.-Y., Bousseau, A., Paris, S., Durand, F., and Drettakis, G. 2012. Coherent intrinsic images from photo collections. *ACM Transactions on Graphics (SIGGRAPH Asia Conference Proceedings) 31*.

Lalonde, J.-F., Efros, A. A., and Narasimhan, S. G. 2009. Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. *ACM Transactions on Graphics (SIGGRAPH Asia 2009) 28,* 5 (December).

Lalonde, J.-F., Efros, A. A., and Narasimhan, S. G. 2010. Detecting ground shadows in outdoor consumer photographs. In *European Conference on Computer Vision*.

Lalonde, J.-F., Efros, A. A., and Narasimhan, S. G. 2011. Estimating the natural illumination conditions from a single outdoor image. *International Journal of Computer Vision*.

Land, E. H. and McCann, J. J. 1971. Lightness and retinex theory. *Journal of the optical society of America 61,* 1.

Lee, K. J., Zhao, Q., Tong, X., Gong, M., Izadi, S., Lee, S. U., Tan, P., and Lin, S. 2012. Estimation of intrinsic image sequences from image+depth video. In *Proceedings of European Conference on Computer Vision (ECCV)*. 327–340.

Levin, A., Lischinski, D., and Weiss, Y. 2008. A closed-form solution to natural image matting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 30,* 2, 228–242.

Liu, X., Wan, L., Qu, Y., Wong, T.-T., Lin, S., Leung, C.-S., and Heng, P.-A. 2008. Intrinsic colorization. *ACM Transactions on Graphics (proc. of SIGGRAPH Asia) 27*, 152:1–152:9.

Loscos, C., Frasson, M.-C., Drettakis, G., Walter, B., Granier, X., and Poulin, P. 1999. Interactive virtual relighting and remodeling of real scenes. In *Proceedings of the 10th Eurographics conference on Rendering*. 329–340.

Niessner, M., Zollhöfer, M., Izadi, S., and Stamminger, M. 2013. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*.

Okabe, M., Zeng, G., Matsushita, Y., Igarashi, T., Quan, L., and yeung Shum, H. 2006. Single-view relighting with normal map painting. In *Proceedings of Pacific Graphics 2006*. 27–34.

Omer, I. and Werman, M. 2004. Color lines: Image specific color representation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*. Vol. 2. IEEE, II–946.

Panagopoulos, A., Wang, C., Samaras, D., and Paragios, N. 2013. Simultaneous cast shadows, illumination and geometry inference using hypergraphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 35,* 2, 437–449.

Perez, R., Seals, R., and Michalsky, J. 1993. All-weather model for sky luminance distribution – Preliminary configuration and validation. *Solar Energy 50,* 3, 235–245.

Pons, J.-P., Keriven, R., and Faugeras, O. 2007. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. *The International Journal of Computer Vision 72,* 2 (Apr), 179–193.

Preetham, A. J., Shirley, P., and Smits, B. 1999. A practical analytic model for daylight. In *SIGGRAPH*. 91–100.

Sanin, A., Sanderson, C., and Lovell, B. C. 2012. Shadow detection: A survey and comparative evaluation of recent methods. *Pattern recognition 45,* 4, 1684–1695.

Savarese, S., Andreetto, M., Rushmeier, H., Bernardini, F., and Perona, P. 2007. 3d reconstruction by shadow carving: Theory and practical evaluation. *International journal of computer vision 71,* 3, 305–336.

Shan, Q., Adams, R., Curless, B., Furukawa, Y., and Seitz, S. M. 2013. The visual turing test for scene reconstruction. In *Proc. of International Conference on 3D Vision (3DV '13)*. IEEE Computer Society, Washington, DC, USA, 25–32.

Shen, J., Yang, X., Jia, Y., and Li, X. 2011. Intrinsic images using optimization. In *CVPR*.

Shen, L. and Yeo, C. 2011. Intrinsic images decomposition using a local and global sparse representation of reflectance. In *CVPR*. 697–704.

Shih, Y., Paris, S., Durand, F., and Freeman, W. T. 2013. Data-driven hallucination of different times of day from a single outdoor photo. *ACM Transactions on Graphics (TOG) 32,* 6, 200.

Shor, Y. and Lischinski, D. 2008. The shadow meets the mask: Pyramid-based shadow removal. *Computer Graphics Forum 27,* 2, 577–586.

Snavely, N., Seitz, S. M., and Szeliski, R. 2006. Photo tourism: exploring photo collections in 3d. In *ACM transactions on graphics (TOG)*. Vol. 25. ACM, 835–846.

Sunkavalli, K., Matusik, W., Pfister, H., and Rusinkiewicz, S. 2007. Factored time-lapse video. *ACM Transactions on Graphics (proc. of SIGGRAPH) 26,* 3.

Szeliski, R. 2010. *Computer Vision: Algorithms and Applications*, 1st ed. Springer-Verlag New York, Inc., New York, NY, USA.

Tao, L., Yuan, L., and Sun, J. 2009. Skyfinder: attribute-based sky image search. In *ACM Transactions on Graphics (TOG)*. Vol. 28. ACM, 68.

Troccoli, A. and Allen, P. 2008. Building illumination coherent 3d models of large-scale outdoor scenes. *International Journal of Computer Vision 78,* 2-3, 261–280.

Weiss, Y. 2001. Deriving intrinsic images from image sequences. In *ICCV*. 68–75.

Wu, T.-P., Tang, C.-K., Brown, M. S., and Shum, H.-Y. 2007. Natural shadow matting. *ACM Transactions on Graphics 26,* 2, 8.

Xing, G., Zhou, X., Peng, Q., Liu, Y., and Qin, X. 2013. Lighting simulation of augmented outdoor scene based on a legacy photograph. *Computer Graphics Forum (proc. Pacific Graphics) 32,* 7, 101–110.

Ye, G., Garces, E., liu, Y., Dai, Q., and Gutierrez, D. 2014. Intrinsic Video and Applications. *ACM Trans. Graph. (SIGGRAPH) 33,* 4.

Yu, Y., Debevec, P., Malik, J., and Hawkins, T. 1999. Inverse global illumination: recovering reflectance models of real scenes from photographs. In *SIGGRAPH*. 215–224.

Yu, Y. and Malik, J. 1998. Recovering photometric properties of architectural scenes from photographs. In *SIGGRAPH*. 207–217.

Zhao, Q., Tan, P., Dai, Q., Shen, L., Wu, E., and Lin, S. 2012. A closed-form solution to retinex with nonlocal texture constraints. *IEEE Trans. PAMI 34*, 1437–1444.

Zhu, J., Samuel, K. G. G., Masood, S., and Tappen, M. F. 2010. Learning to recognize shadows in monochromatic natural images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2010)*.