Learning to Factorize and Relight a City

Andrew Liu[1(✉)], Shiry Ginosar[2(✉)], Tinghui Zhou[3(✉)], Alexei A. Efros[2(✉)], and Noah Snavely[1(✉)]

[1] Google, Berkeley, USA
{ahliu,snavely}@google.com
[2] UC Berkeley, Berkeley, USA
{shiry,efros}@eecs.berkeley.edu
[3] Humen, Inc., San Francisco, USA

Abstract. We propose a learning-based framework for disentangling outdoor scenes into temporally-varying illumination and permanent scene factors. Inspired by the classic intrinsic image decomposition, our learning signal builds upon two insights: 1) combining the disentangled factors should reconstruct the original image, and 2) the permanent factors should stay constant across multiple temporal samples of the same scene. To facilitate training, we assemble a city-scale dataset of outdoor timelapse imagery from Google Street View, where the same locations are captured repeatedly through time. This data represents an unprecedented scale of spatio-temporal outdoor imagery. We show that our learned disentangled factors can be used to manipulate novel images in realistic ways, such as changing lighting effects and scene geometry. Please visit http://factorize-a-city.github.io/ for animated results.

1 Introduction

> *"The city of Sophronia is made up of two half-cities... One of the half-cities is permanent, the other is temporary."*
> —Italo Calvino, *Invisible Cities*

Imagine taking an image from every possible location on Earth at every possible time instant throughout history. Adelson and Bergen called this hypothetical construct the *plenoptic function* [2]. In practice, of course, it would be impossible to capture or store such a massive dataset. Yet, the data must also be highly redundant and compressible. There will be many images of the same view with slightly different illumination, many images capturing different places under the same conditions, etc. In other words, each image within this hypothetical dataset should have a low intrinsic dimensionality. Rather than store all pixels, we could instead store a small number of intrinsic, disentangled factors representing scene geometry, illumination conditions, etc.—if only we knew what those parameters were and how to reconstruct an image from them.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58548-8_32) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12349, pp. 544–561, 2020.
https://doi.org/10.1007/978-3-030-58548-8_32

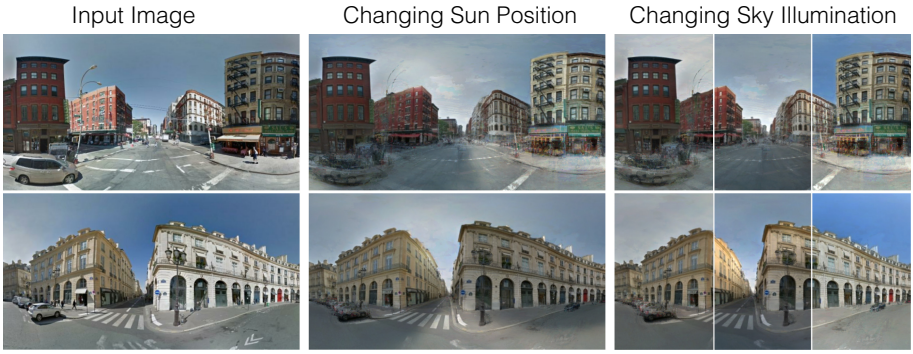

Fig. 1. We learn to disentangle temporally-varying scene factors from permanent ones. We can manipulate the learned factors to relight scenes, e.g., by editing sun position and sky conditions. While we train our model on panoramas of NYC *(top)*, it generalizes at test time to images of other cities such as Paris *(bottom)*.

In this paper, we ask whether we can learn such a lower-dimensional representation from a sparse sampling of the plenoptic function on the scale of an entire city. Until recently, large-scale visual data that varies both in space and, separately, in time was difficult to obtain. Fortunately, there have been systematic efforts to capture the world through projects like Google Street View (GSV). While GSV is known for its worldwide coverage, it has also accumulated many samples of the world over time, powering features like Street View Time Machine (GSV-TM). However, GSV-TM still represents an extremely sparse sampling of the plenoptic function.

We use GSV to learn to factor a city's worth of outdoor panoramas into a single low-dimensional representation. In particular, we organize a large set of historical GSV panoramas of New York City into *assembled timelapses* at 100,000 fixed locations captured over time. These enable us to train an unsupervised model to disentangle two latent factors: illumination factors that vary over time, and geometric scene properties that are more permanent.

Once we learn a disentangled set of latent factors, we can synthesize missing data in our incomplete sampling of the plenoptic function by simply swapping or modifying the underlying factors. As illustrated in Fig. 1, our learned factorization can generate synthetic images of the same scene with completely novel illumination. Our disentangled factors are flexible enough to relight test scenes from a single panorama and can even be applied to entirely new cities like Paris.

2 Related Work

Intrinsic Images. Decomposing images into their underlying components is a well-studied problem [5]. For instance, the classic intrinsic images problem describes images as a combination of *reflectance* (i.e., scene albedo), and *shading* (effects induced by lighting) [1]. This problem is underconstrained as there are an

infinite number of possible solutions for a single image. However, the regularities in natural scenes and lighting conditions allow for priors on the decomposition. While such priors can be manually crafted [4], many recent methods attempt to learn priors from data, using full supervision from synthetic data [23], sparse supervision from human annotations [6,39], or self-supervision from synthetic models [16]. Yet another kind of supervision comes from *timelapse videos* [24], which feature image sequences with constant reflectance but varying illumination. Such work hearkens back to classic work on deriving intrinsic images from image stacks [37], and is an inspiration for our work. However, while intrinsic image methods allow for editing reflectance or shading for a specific image, they use high-dimensional *pixel-level* descriptions of lighting that are not transferable across scenes. In our case, our model learns an illumination descriptor that can be meaningfully transferred from one image to another, e.g., to relight an image with an illumination from a completely different scene. Such "mix-and-match" capabilities are beyond the power of standard intrinsic images.

Inverse Graphics. An alternative way to factor visual appearance is via 3D reconstruction of the scene into underlying physical components like 3D shape, materials, and lighting. Such methods have been successful in several specific domains, including faces [33], single objects [18,41], or indoor scenes trained from synthetic data [25,32]. 3D reconstruction has also been used explicitly as a preprocess to aid in modeling visual appearance [19,26,27,30]. Most relevant to us are Martin-Brualla *et al.* [26], who organized millions of internet photos into a dense 3D and temporal reconstructions, and Meshry *et al.* [27], who employed a dense 3D reconstruction with a neural rendering pipeline to synthesize scene appearances. However, explicit 3D reconstruction methods require hundreds of images to create a 3D model and cannot generalize to novel test-time scenes. In contrast, we choose to handle geometry implicitly—allowing us to holistically learn to disentangle factors across many scenes composed of a few images each, and then generalize to novel settings, even single images.

Some recent inverse graphics methods learn to infer shape, appearance, and materials for new outdoor scenes, not just scenes observed during training. Yu and Smith train on multi-view stereo data using a physics-based inverse graphics model, and can infer explicit scene properties for novel test images, enabling relighting tasks [38]. Our work achieves a similar capability, but relies on a more implicit representation of geometry and illumination that can be learned solely from timelapse data, without requiring depth or surface normals during training.

Timelapse and Webcam Data. Timelapses are a popular source of data for capturing time-related effects. Applications include intrinsic images [24,37], scene-specific factorizations via physical shading models [34], illuminant transfer [21], analysis of worldwide temporal variations [15], motion denoising [31], learning temporal object transformations [40], and weather attribute manipulation [20]. However, prior work is limited by the variety and size of available data. The largest existing set of standard webcam data is the AMOS dataset of Jacobs *et al.* [15], which archived 29,445 webcams and 95 million images. BigTime [24] uses a much smaller set of 6,500 images from 195 timelapse sequences. Both

Fig. 2. *Left:* A Manhattan intersection. *Center:* Multiple Google Street View panoramic captures of this intersection forms an *assembled timelapse* stack. *Right:* The train and test split over the greater NYC area. Training stacks are drawn from the blue region, and test stacks from the yellow region. (Color figure online)

datasets sample *time* much more densely than *space*. In contrast, we leverage the vast amounts of data from Google Street View to create *assembled timelapses* of the same location captured at different times, across a large number of locations. This allows us to collect an order of magnitude more data than previously published [15]. We additionally note that data collection from Street View scales more easily than [15] which requires crawling the internet for webcam streams.

Learning from Street View. Google Street View (GSV), a large dataset of images sampling much of the world's streets, represents a compelling source of data for computer vision research. Researchers have utilized Google Street View images to learn about visual elements [9] or historical architectural styles [22] specific to certain cities like Paris, to predict non-visual city attributes [3,10,28], for localization [11], or to understand the relationship between satellite imagery and street-level views [35]. In our work we use historical GSV Time Machine imagery to observe how the world changes over time by assembling timelapses for a large number of locations. Such a large, comprehensive dataset is key to our unsupervised approach for learning to factor illumination from scene geometry.

3 Google Street View Time Machine Data

Google Street View (GSV) hosts an amazing quantity of panoramas capturing street scenes worldwide. Because GSV repeatedly captures many places over time, it can be treated as a sparse, imperfectly aligned, and irregularly-sampled collection of timelapse videos. These historical images are saved as part of the GSV-Time Machine (GSV-TM), which we mine to collect our dataset.

We focus on New York City, due to the richness of NYC scenes and the relative wealth of data. To assemble timelapses, we collect panoramas within NYC along with their timestamps and camera poses in a geographic coordinate system [8]. We greedily cluster nearby panoramas into sets of eight, which we refer to as *stacks*. The region we use and an example stack are shown in Fig. 2.

From the area shown in Fig. 2 (right) we collect ∼100K assembled timelapse stacks for training (comprised of 800K individual panoramas stitched from 10 million captures) and 16K test stacks. We crop the sky and ground regions such

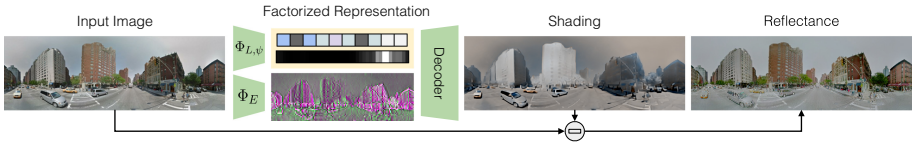

Fig. 3. Disentangling a single image. At test time, we *encode* a single image into disentangled time-varying and permanent factors. We train with the constraint that shading and reflectance images can be *decoded* from this learned factored representation.

that our final panoramas are 960×320. These sRGB panoramas can optionally be gamma-corrected before further processing.

4 Method

Our goal is to discover a low-dimensional representation of the world where temporally varying effects, such as different illumination conditions, are disentangled from permanent objects, such as buildings and roads.

One form of disentanglement is *intrinsic images*, a per-pixel decomposition into reflectance and shading images. However, such a disentangled representation is very low-level—a particular shading image cannot be used to relight a different scene. Instead, we seek to encode an image into higher-level latent factors capturing scene and illumination properties described above, as illustrated in Fig. 3. How can we find such a factorization? Our insight is that we should still be able to *decode* intrinsic images from our factored representation, as illustrated on the right side of Fig. 3. The decoded reflectance and shading images should recombine to form the original image, providing us with an autoencoder-style method for learning our high-level factorization [16]. However, such an image reconstruction framework alone would provide a very weak supervision signal. Our second insight is to learn from huge numbers of *timelapse stacks* mined from GSV-TM. Within such stacks, we assume the scene factors to be constant. This insight is inspired by the work of Li and Snavely, who learn intrinsic images from timelapse videos [24]. In our case we learn a high-level factorization that enables more powerful capabilities.

4.1 Encoder-Decoder Architecture

Figure 3 shows our encoder-decoder architecture with its learnt factored representation. Given an image, our encoders produce latent factors, capturing various temporal and permanent effects, that can be decoded to a log-shading intrinsic image. We use the intrinsic images equation ($\log(\text{Reflectance}) = \log(\text{Image}) - \log(\text{Shading})$) to compute a reflectance image by subtracting the temporally varying effects, represented by the shading image, from the original image.

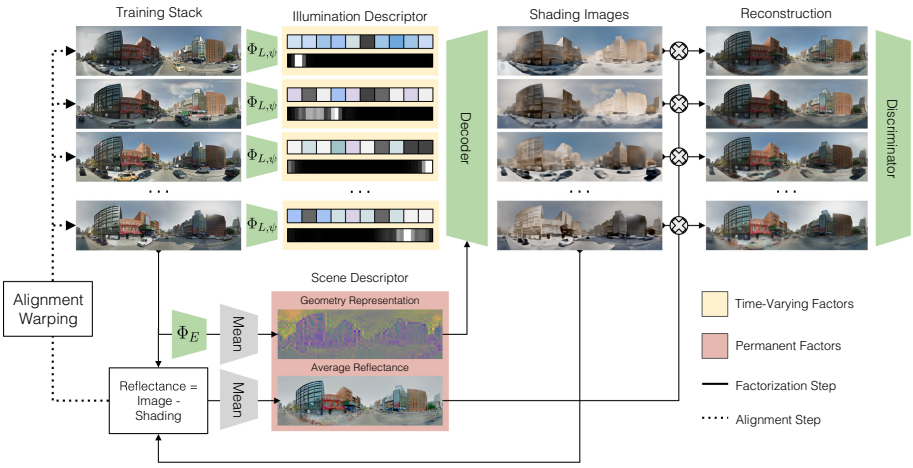

Fig. 4. Training with timelapses. We train encoders to disentangle an assembled timelapse stack into two factors: *illumination descriptors* that capture the time-varying aspects of each image, and a single *scene descriptor* that captures the permanent elements of the entire timelapse stack, such as the scene geometry. We train a generator to transform the disentangled factors into shading and reflectance images from which we can reconstruct the original images. As indicated by the dotted pathways, we also simultaneously solve for the alignment of the individual frames in the input timelapse.

Our model's latent factors are organized into two sets of descriptors, as shown in Fig. 4: an *illumination descriptor* represents temporally varying aspects of the scene and a *scene descriptor* represents the permanent aspects.

Illumination Descriptor: Our illumination descriptor captures the factors of the world that encode temporal variation like lighting. This descriptor is comprised of two disentangled sub-factors:

The *lighting context* $L \in \mathbb{R}^{32}$ is a global latent feature that captures the overall ambient illumination properties, such as atmospheric conditions and cloud cover. Our lighting context encoder Φ_L encodes an image to this embedding.

The *sun azimuth angle*, φ is an explicit factor representing the horizontal position of the sun in a given panorama. We model sun azimuth explicitly because, unlike illumination patterns, variations in sun azimuth have a simple geometric meaning, with a value in the range $[-\pi, \pi]$. Despite this simple parameterization, the effect of sun azimuth on a rendered scene is highly complex. Therefore an explicit azimuth factor allows our model to combine the factor's underlying mathematical simplicity with a network's ability to model complex behaviors.

Rather than regress to a scalar angle, we instead represent φ internally as a discretized distribution over sun angle (with $k = 40$ bins). Inspired by prior work on illumination estimation [12], our azimuth encoder Φ_φ is a horizontally fully-convolutional network that takes as input a panorama, and produces a 40-way softmax distribution φ, where each bin corresponds to the probability

that the sun azimuth is located in the bin's corresponding angular range. Note that given this discrete distribution over angles, we can differentiably compute a single scalar angle as the (circular) expectation of the distribution, $\bar{\varphi}$. This predicted scalar sun angle is used by our decoder for normalizing sun position.

Scene Descriptor: Our scene descriptor captures the permanent structure of the world that is invariant to the temporally varying effects described above. We also divide this descriptor into two disentangled sub-factors:

The *geometry representation* is a spatial map of learned features that captures scene properties (e.g. surface normals and material properties) that are independent of illumination, but nonetheless are important to determining the rendering of a shading images. The fully convolutional encoder Φ_E outputs $E \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 16}$ where H and W are the resolution of a panorama.

The *reflectance image* is an RGB estimate of the underlying scene albedo. In contrast to the shading image, we chose to not use an encoder-decoder to compute reflectance for two reasons: (1) neural networks can have difficulties preserving high-frequency textures that are important for visual quality and (2) it suffices to predict only one intrinsic image component because its complement component has a closed form solution based on the intrinsic images equation.

Decoder: Given a set of learned factors (sun azimuth angle $\bar{\varphi}$, lighting context L, and geometry factor E), our decoder G is trained to generate an outdoor shading image. To facilitate training of G, one insight is that it is easier to learn to synthesize shading images with a fixed sun azimuth angle than with all possible angles. Further, we can normalize a panorama by its predicted sun azimuth angle by simply rotating it by the negative of that angle (i.e., circular horizontal translation). Hence, our decoder operates as follows: (1) use the predicted sun azimuth angle $\bar{\varphi}$ to rotate the geometry factor image E to a fixed sun angle, (2) decode the sun-normalized geometry image with lighting context L to a shading image, and (3) rotate the result back to the original coordinate frame.

We use the Spatial Adaptive Instance Normalization (SPADE) generator of Park *et al.* [29] to model the complex interactions between geometry and illumination in our decoder G. The SPADE generator takes the lighting context L as the network's noise input. We apply the insights from above and rotate the geometry representation E by $-\bar{\varphi}$ before using it as the SPADE conditioning.

While some prior works model shading with a grayscale image, such a model cannot capture real-world, colored illumination. Inspired by Sunkavalli *et al.* [34], we augment our decoder's gray-scale shading predictions with a bi-color assumption by additionally predicting two global color illuminants c_1 and c_2, corresponding to sunlight and skylight, and a per-pixel mixing weight M that models how much each pixel is illuminated by the sun or sky. For further details about the decoder architecture, please refer to the supplemental material.

4.2 Training

Learning to factor single images without *any* supervision is challenging—there is simply not enough information in a single image to disentangle scene factors

Fig. 5. Alignment results. We show stack averages, cropped for emphasis, before and after our alignment process. Aligning the estimated permanent reflectances rather than the input images results in good alignment and therefore crisp stack averages.

from illumination factors. However, a GSV-TM stack depicts the same underlying permanent scene under diverse temporally varying illuminations, providing a useful training signal. Our training procedure, shown in Fig. 4, learns to disentangle factors *within* a stack by separating the permanent geometry of the scene shared by all images in the stack from the varying lighting. The trained model can be applied to a single image at test time.

Given a timelapse stack, we run our encoder on individual frames to get a stack of encoded geometry representations and illumination descriptors. Because we assume the stack's geometry to be constant across time, we average the encoded geometry maps over the stack, resulting in a single shared geometry map, $\bar{E}$. From this shared geometry map, and the per-image illumination factors, our decoder produces a stack of shading and reflectance image pairs. As with geometry, we wish the scene's albedo to be constant across time. Accordingly, we impose a reflectance consistency loss $\mathcal{L}_{\mathsf{RC}}$ that computes the L_1 distance between pairs of reflectance images from different frames. This loss encourages the encoder-decoder network to remove temporal variation from the encoded permanent factors such that the reflectances are constant across a stack.

As demonstrated in the right half of Fig. 4, we average the stack's reflectance images across frames to get the stack's shared reflectance. The shared reflectance is recomposited with the shading image of each frame in the stack to reconstruct the original pixels of each input frame. These reconstructions are used to drive the learning process via image synthesis losses.

4.3 Stack Alignment

Unlike traditional webcam data, our assembled GSV-TM timelapses do not come from stationary cameras. While each stack consists of nearby panoramas, they are not perfectly co-located and aligned. As shown in Fig. 5, the average of the stack reveals visible misalignment artifacts resulting from this parallax.

We could use 3D reconstruction methods as the basis for image alignment, but opted for a simpler 2D approach inspired by image congealing [13], and

compute 2D warps that best align the images in each stack. Given a raw stack of imperfectly aligned images, we define Θ, an 8×32 grid of per-image control points initialized as the identity warp. The control points define a 2D spline used to differentiably warp each image within a stack to align with the rest.

To find the control points that best align images within a stack, we run gradient descent to minimize pixel alignment error. While one could use original image pixels to measure misalignment, we found that photometric differences across the stack due to varying lighting conditions led to poor alignments. Instead, we compute error on estimated *reflectance* images by reusing our previously defined reflectance consistency loss, $\mathcal{L}_{\mathsf{RC}}$, to update alignment parameters. This approach is indicated by the dotted pathway in Fig. 4. By jointly minimizing alignment and intrinsic image decomposition, we create a positive feedback loop—as timelapse alignment improves, factorization becomes easier and vice versa.

4.4 Losses

Our losses are optimized over alignment parameters Θ, factorization encoders Φ_L Φ_φ, and Φ_E, and decoder G. We train a multi-scale patch discriminator [14,36] D to ensure that the stack reconstructions with shared reflectances look realistic.

Our primary loss for learning the disentanglement is the reflectance consistency loss $\mathcal{L}_{\mathsf{RC}}$ described in Sect. 4.2. We include standard image generation losses on the reconstructed stack to ensure high quality synthesis results: a perceptual loss $\mathcal{L}_{\mathsf{VGG}}$ [17], an adversarial loss $\mathcal{L}_{\mathsf{GAN}}$ [7], and a feature matching loss $\mathcal{L}_{\mathsf{FM}}$ [36]. Finally, because intrinsic images have a fundamental color ambiguity, we also include a white light penalty, $\mathcal{L}_{\mathsf{WL}}$ that biases our encoder-decoder towards white-balanced reflectance outputs. Our overall objective function is:

$$\min_{\Theta} \max_{D} \min_{G,\Phi_L,\Phi_\varphi,\Phi_E} \mathcal{L}_{\mathsf{RC}} + \mathcal{L}_{\mathsf{Gen}} + \mathcal{L}_{\mathsf{GAN}} \tag{1}$$

where $\mathcal{L}_{\mathsf{Gen}}$ is a weighted sum of $\mathcal{L}_{\mathsf{FM}}, \mathcal{L}_{\mathsf{WL}}, \mathcal{L}_{\mathsf{VGG}}$ that measures the generative quality of the reconstructed images. We include additional descriptions, alignment results, insights, and analysis for reproducibility in the supplemental material.

5 Experiments

We evaluate our factorization method in two ways: 1) we compare to intrinsic image decomposition baselines in the single-scene setting, and 2) we apply our method to the task of transferring illumination descriptors across different scenes, a new capability enabled by our disentanglement. In both cases, we measure success by the quality of reconstructed images derived from swapping their disentangled factors with ones borrowed from other images as in [39].

Data. At test time, our network can take as input either an assembled timelapse stack or a single panorama. In order to align test-time stacks like those shown

in Fig. 5, we estimate spline parameters by computing a gradient for alignment only, while keeping the weights of the factorization part of the network frozen. Below, we present results for stack as well as single-image inputs.

In particular, we show single-image test-time results on GSV imagery from cities never seen during training, such as Paris, as well as images from the Outdoor Laval HDR dataset [12]. This dataset contains HDR panoramas of outdoor scenes that are tonemapped to sRGB to match GSV. We use this data to compare to existing sRGB intrinsic image methods and to test generalization from GSV to a different domain of panoramas.

Baselines. Given the novelty of our problem, we perform model ablations to measure the individual benefits of various components. All ablated models are trained with the same losses and number of iterations as our full method. We report results on the following ablations:

- **Mono-color shading**: We ablate the bi-color shading by training our model with a mono-color assumption similar to that of Li and Snavely [24].
- **w/o alignment training**: Trained without the alignment feedback loop.
- **w/ unaligned test stacks**: Uses unaligned test stacks to measure the effect of ablating alignment at training (above) vs. at both training and test time.
- **w/o azimuth encoder**: Our model trained without an azimuth encoder nor normalizing for sun position.

Additionally, we consider the following baselines:

- **Pixel nearest neighbor**: Given a target image, we find the pixel-wise nearest neighbor in its aligned stack and report the error resulting from using that image as our synthesized result.
- **Weiss's MLE Intrinsics** [37]: use handcrafted priors on gradients extracted from image sequences.
- **Zhou *et al.*** [39]: learn to mimic human judgments of relative reflectance.
- **Li and Snavely's BigTime** [24] learn shading priors from image sequences.

5.1 Within-Scene Decomposition

Intrinsic image methods aim to decompose an image into shading and reflectance. The quality of a decomposition is measured by its ability to separate illumination effects, like cast shadows, from permanent properties such as albedo. In Fig. 6, we show reflectance and shading computed from a single image using our method and the two deep learning baselines. Both BigTime and Zhou *et al.* fail to remove cast shadows, as seen by residual shadows encoded in their reflectance. Unlike Zhou *et al.*, our method produces shading images that are piecewise smooth, as expected for planar surfaces like building facades. BigTime struggles in outdoor settings because their single global illuminant cannot predict multiple illumination colors. Finally, both baselines incorrectly encode blue sky pixels as reflectance despite the fact that sky color is a temporal property. To further illustrate the advantages of our method over these baselines, Fig. 7 shows the

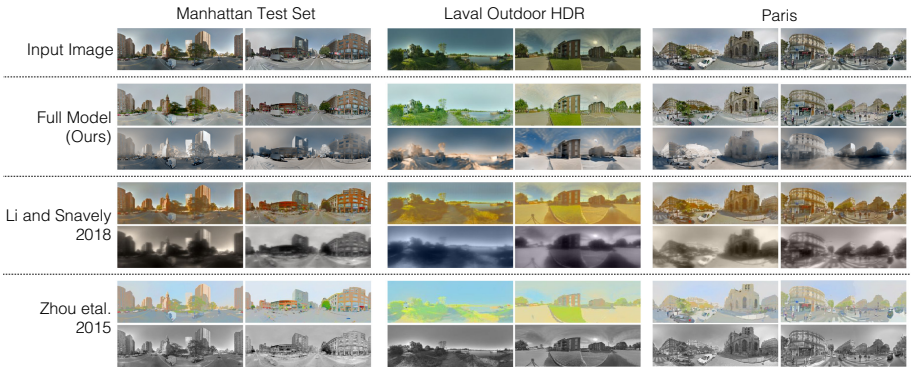

Fig. 6. Qualitative results on an intrinsic image decomposition task. We compare single-image decompositions of our method with Li and Snavely [24] and Zhou *et al.* [39]. Compared to the baselines, our reflectance images do not have residual shadows. Our method, trained on NYC, generalizes at test-time to Laval Outdoor HDR Panoramas [12] as well as to GSV imagery from Paris.

results of relighting pairs of images of the same scene by swapping reflectances within the pair. Unlike the baselines, our clean reflectance image allows us to relight the scene successfully.

Scene Consistency Verification. Since MLE Intrinsics [37] only works on timelapse stacks of single scenes, we devise a way to quantitatively compare to their method. We split our aligned test stacks to two smaller substacks of 4 images each. For each substack, each method predicts a single reflectance image and four shading images. Since both substacks capture the same underlying scene, the predicted reflectances should be consistent across the two. As in the case of single images (Fig. 6), we can test the consistency of the predicted reflectance for the depicted scene by swapping the predicted reflectance images between the two substacks and reconstructing the four input images in each substack from their shading and *swapped* reflectance images. We refer to this experiment as *scene consistency verification* because the reconstruction error is minimized when the predicted reflectances are identical for the two substacks.

We report the mean squared reconstruction error (MSE) between the input stack and the swap reconstructions in Table 1. Our method outperforms the three baselines at image reconstruction in this setting. We speculate that prior methods are hindered by their reliance on hand-defined shading priors and limited training data. In contrast, our massive dataset provides enough supervision for learning a good decomposition without shading priors. Interestingly, ablating the azimuth encoder does not degrade performance on this task, suggesting that a simpler setup is sufficient for within-scene illumination transfer.

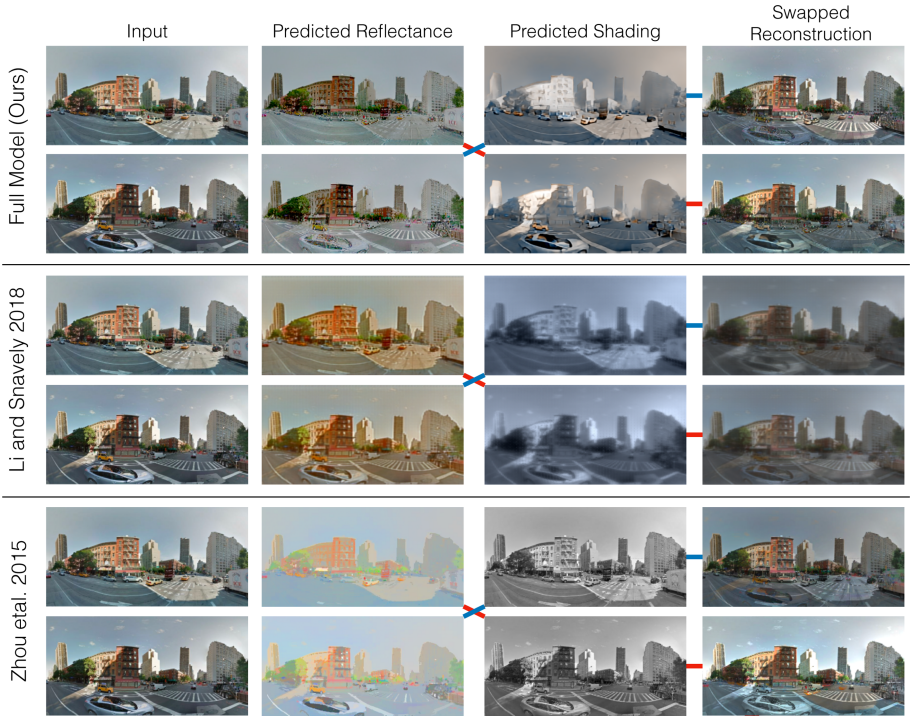

Fig. 7. Transferring illumination within a scene. Given a pair of images of the same scene under different illuminations (*left*), we disentangle the permanent and varying factors and decode their reflectance and shading (*middle*). To test the permanency of the estimated reflectance for the depicted scene, we swap reflectances within the pair and combine them with the estimated shading to reconstruct the original images (*right*). Red and blue paths connect the components used to reconstruct each image. Our method produces a reflectance, clean of any lighting, which can be safely swapped between captures of the same scene and still result in good reconstructions. (Color figure online)

5.2 Cross-Scene Factorization

Unlike intrinsic images methods, our factorization allows us to transfer illumination descriptors *across* scenes. Using our disentangled factors, we can synthesize a given scene under completely new lighting conditions, borrowed from a *different* location. For the purpose of evaluating the success of this cross-scene relighting process, we devise a way to compare the novel synthesis to ground truth. Namely, because illumination changes relatively slowly, we assume that images captured within 5 min across the city have the same illumination descriptor. Hence, we can relight a given scene, A, captured at time T_1 using illumination descriptors transferred from a different location, B, captured at time T_2. We then compare the resulting synthetic image of scene A at time T_2 to ground truth captures of scene A captured at a time close to T_2.

Table 1. Relighting results. We define two image reconstruction tasks for evaluation. *Scene consistency verification* evaluates whether the estimated reflectance is consistent across multiple captures of a single scene. *Space-time completion* evaluates the ability to transfer illumination across different scenes. We report MSE reconstruction error. Lower is better.

Model	Consistency	Completion
Full model (ours)	**0.071**	**0.196**
Mono-color shading	0.077	0.215
w/o alignment training	0.082	**0.201**
w/ unaligned test stacks	0.090	0.210
w/o azimuth encoder	**0.072**	0.240
Pixel nearest neighbors	0.274	0.278
MLE Intrinsic [37]	0.114	—
BigTime [24]	0.180	—
Zhou *et al.* [39]	0.217	—

We name this task *space-time matrix-completion*. A row in the matrix represents a unique point in "space" and a column represents a unique point in "time". A single panorama represents an entry in this matrix at the row corresponding to its depicted scene and column corresponding to its capture time. We can withhold entries in the matrix and reconstruct them by combining a scene descriptor derived from images in the same row, with an illumination descriptor extracted from a different scene from the same column. Table 1 shows the reconstruction MSE for each ablation between held-out and reconstructed views. Our full model and the *w/o alignment training* ablation show significant improvements over other ablations.

While alignment training does not significantly affect the performance of our model on this task (*w/o alignment training*), its performance degrades significantly on unaligned stacks (*w/ unaligned test stacks*). This indicates that alignment may be optional during training but is crucial for reconstruction. Additionally, unlike with the substack swap task, explicitly representating sun azimuth improves transferability of lighting descriptors across scenes.

6 Applications

We now present applications where we synthetically modify a panorama. These applications are uniquely enabled by our intrinsic factorization that disentangles time-varying effects from the permanent scene properties.

Changing Sun Position. Our model disentangles sun azimuth angle from scene and lighting context factors. Once a scene is factorized, we can visualize what a scene looks like when the sun angle is changed. Figure 8 shows examples of test

scenes synthesized with new sun azimuth angles. Note that cast shadows and illumination on building faces change realistically with the rotation.

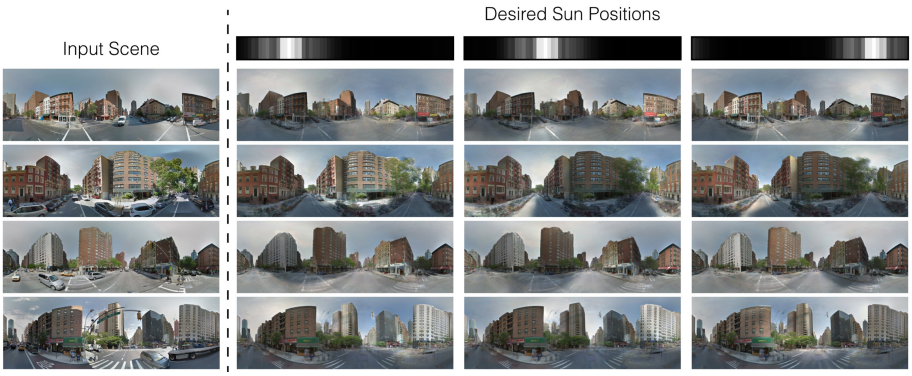

Fig. 8. Manipulating sun position. We can specify the sun position for an input scene and relight it realistically. Please see the supplemental video for full animations.

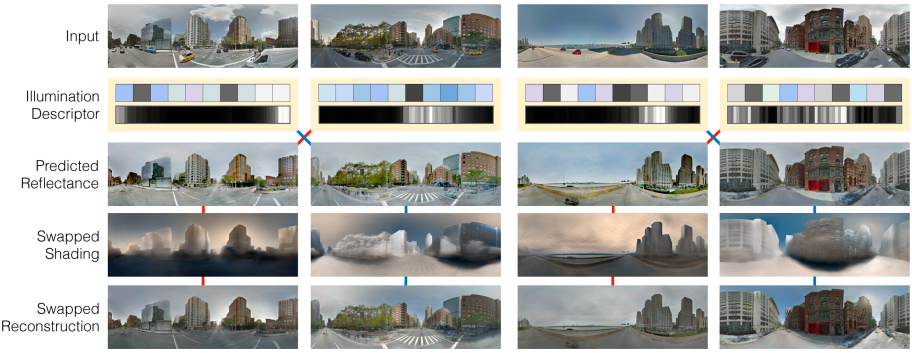

Fig. 9. Changing sky illumination. We can relight *novel* scenes by transferring the disentangled time-varying factors from one scene to another. Here we swap the illumination descriptors of a pair of input scenes to visualize what each scene might look like under a new illumination. The red and blue paths indicate the components used to reconstruct each relit scene. (Color figure online)

Relighting a *Novel* Scene. Our lighting context encodes the stylistic quality of illumination. As shown in Fig. 9, we can transfer the whole illumination descriptor, including sun azimuth, from one panorama to another with a new scene geometry. Results for transferring *only* lighting context can be found in the supplemental material. The supplemental material also demonstrates relighting a spatial sequence of panoramas from different times to a fixed illumination, thus producing a virtual drive through Manhattan.

Editing Scene Geometry. While shading and azimuth capture the essence of time, the scene descriptor encodes structures. By copy-pasting regions of the scene descriptors, we can transplant the buildings into new panoramas and relight them to match the scene. Please see the supplementary for results.

7 Discussion

We proposed a novel source of large-scale timelapse data from historical Street View data, and a learning-based method for factorizing temporal and permanent variations across imagery covering an entire city. Our learned factorization outperforms state-of-the-art intrinsic images methods, and enables cross-scene style transfer via manipulating our learned factors.

Our method has a few limitations. First, the scene descriptor learns to encode transient objects like cars. While moving objects are temporal effects, the network chooses to encode them in the scene descriptor, resulting in wispy cars appearing in the generator output. Second, high-frequency details such as cast shadows from tree branches are difficult to synthesize. Third, when the alignment module fails, the shared reflectance of a stack will appear blurry. Please see the supplemental material for examples of failure cases. Finally, when our permanence assumptions fail to hold—for instance when buildings are repainted or rebuilt—our assumption that the scene descriptor is constant across time is violated.

Despite these limitations, our work points towards a new approach to modeling and synthesizing the space of outdoor scenes, wherein we can learn to separate factors that persist at different time scales. An intriguing direction for future work is to expand to a richer range of timescales, for instance modeling transient effects (moving people, cars, etc.), effects with annual cycles (e.g., seasons), long-term changes like weathering, etc.

Acknowledgements. We would like to thank Richard Tucker, Richard Bowen, Ameesh Makadia, and Vincent Sitzmann for insightful discussions. We would also like to thank Angjoo Kanazawa and Tim Brooks for their help with preparing the manuscript. This work is supported, in part, by NSF grant IIS-1633310.

References

1. Adelson, E.H., Pentland, A.P.: The perception of shading and reflectance. In: Knill, D.C., Richards, W. (eds.) Perception as Bayesian Inference, pp. 409–423. Cambridge University Press, New York (1996)
2. Adelson, E.H., Bergen, J.R.: The plenoptic function and the elements of early vision. In: Landy, M., Movshon, J.A. (eds.) Computational Models of Visual Processing, pp. 3–20. MIT Press, Cambridge (1991)
3. Arietta, S.M., Efros, A.A., Ramamoorthi, R., Agrawala, M.: City forensics: using visual elements to predict non-visual city attributes. IEEE Trans. Visual Comput. Graphics **20**(12), 2624–2633 (2014). https://doi.org/10.1109/TVCG.2014.2346446

4. Barron, J.T., Malik, J.: Shape, illumination, and reflectance from shading. Trans. Pattern Anal. Mach. Intell. **37**(8), 1670–1687 (2015)
5. Barrow, H.G., Tenenbaum, J.M.: Recovering intrinsic scene characteristics from images. Comput. Vis. Syst. **2**(3–26), 2 (1978)
6. Bell, S., Bala, K., Snavely, N.: Intrinsic images in the wild. ACM Trans. Graphics (SIGGRAPH) **33**(4), 159:1–159:12 (2014). https://doi.org/10.1145/2601097.2601206
7. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: International Conference on Learning Representations (2019)
8. Klingner, B., Martin, D., Roseborough, J.: Street view motion-from-structure-from-motion. In: Proceedings of the International Conference on Computer Vision (ICCV) (2013)
9. Doersch, C., Singh, S., Gupta, A., Sivic, J., Efros, A.A.: What makes Paris look like Paris? ACM Trans. Graphics (SIGGRAPH) **31**(4), 101:1–101:9 (2012)
10. Gebru, T., et al.: Using deep learning and google street view to estimate the demographic makeup of neighborhoods across the United States. Proc. Natl. Acad. Sci. **114**(50), 13108–13113 (2017). https://doi.org/10.1073/pnas.1700035114
11. Gronat, P., Obozinski, G., Sivic, J., Pajdla, T.: Learning and calibrating per-location classifiers for visual place recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2013
12. Hold-Geoffroy, Y., Sunkavalli, K., Hadap, S., Gambaretto, E., Lalonde, J.F.: Deep outdoor illumination estimation. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), July 2017
13. Huang, G.B., Jain, V., Learned-Miller, E.: Unsupervised joint alignment of complex images. In: Proceedings of the International Conference on Computer Vision (ICCV) (2007)
14. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2016)
15. Jacobs, N., Roman, N., Pless, R.: Consistent temporal variations in many outdoor scenes. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 1–6, June 2007. https://doi.org/10.1109/CVPR.2007.383258
16. Janner, M., Wu, J., Kulkarni, T.D., Yildirim, I., Tenenbaum, J.: Self-supervised intrinsic image decomposition. In: Neural Information Processing Systems, pp. 5936–5946. Curran Associates, Inc. (2017)
17. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
18. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11219, pp. 386–402. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01267-0_23
19. Laffont, P.Y., Bazin, J.C.: Intrinsic decomposition of image sequences from local temporal variations. In: Proceedings of the International Conference on Computer Vision (ICCV), December 2015
20. Laffont, P.Y., Ren, Z., Tao, X., Qian, C., Hays, J.: Transient attributes for high-level understanding and editing of outdoor scenes. ACM Trans. Graphics (SIGGRAPH) **33**(4), 1–11 (2014)

21. Lalonde, J.F., Efros, A.A., Narasimhan, S.G.: Webcam clip art: appearance and illuminant transfer from time-lapse sequences. ACM Trans. Graphics (SIGGRAPH) **28**(5), 1–10 (2009)
22. Lee, S., Maisonneuve, N., Crandall, D., Efros, A.A., Sivic, J.: Linking past to present: discovering style in two centuries of architecture. In: IEEE International Conference on Computational Photography (ICCP) (2015)
23. Li, Z., Snavely, N.: CGIntrinsics: better intrinsic image decomposition through physically-based rendering. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11207, pp. 381–399. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01219-9_23
24. Li, Z., Snavely, N.: Learning intrinsic image decomposition from watching the world. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2018)
25. Li, Z., Shafiei, M., Ramamoorthi, R., Sunkavalli, K., Chandraker, M.: Inverse rendering for complex indoor scenes: shape, spatially-varying lighting and SVBRDF from a single image. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2020)
26. Martin-Brualla, R., Gallup, D., Seitz, S.M.: Time-lapse mining from internet photos. ACM Trans. Graphics (SIGGRAPH) **34**(4), 62:1–62:8 (2015). https://doi.org/10.1145/2766903
27. Meshry, M., et al.: Neural rerendering in the wild. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2019)
28. Naik, N., Philipoom, J., Raskar, R., Hidalgo, C.: Streetscore - predicting the perceived safety of one million streetscapes. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 793–799, June 2014. https://doi.org/10.1109/CVPRW.2014.121
29. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2019)
30. Philip, J., Gharbi, M., Zhou, T., Efros, A.A., Drettakis, G.: Multi-view relighting using a geometry-aware network. ACM Trans. Graphics (SIGGRAPH) **38**(4) (2019). http://www-sop.inria.fr/reves/Basilic/2019/PGZED19
31. Rubinstein, M., Liu, C., Sand, P., Durand, F., Freeman, W.T.: Motion denoising with application to time-lapse photography. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR), pp. 313–320, June 2011
32. Sengupta, S., Gu, J., Kim, K., Liu, G., Jacobs, D.W., Kautz, J.: Neural inverse rendering of an indoor scene from a single image. In: Proceedings of the International Conference on Computer Vision (ICCV) (2019)
33. Sengupta, S., Kanazawa, A., Castillo, C.D., Jacobs, D.W.: SfSNet: learning shape, reflectance and illuminance of faces in the wild. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2018)
34. Sunkavalli, K., Matusik, W., Pfister, H., Rusinkiewicz, S.: Factored time-lapse video. ACM Trans. Graphics (SIGGRAPH) (2007). SIGGRAPH 2007. ACM, New York. https://doi.org/10.1145/1275808.1276504
35. Vo, N.N., Hays, J.: Localizing and orienting street views using overhead imagery. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 494–509. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_30
36. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2018)
37. Weiss, Y.: Deriving intrinsic images from image sequences. In: Proceedings of the International Conference on Computer Vision (ICCV) (2001)

38. Yu, Y., Smith, W.A.: InverseRenderNet: learning single image inverse rendering. In: Proceedings of the Computer Vision and Pattern Recognition (CVPR) (2019)
39. Zhou, T., Krähenbähl, P., Efros, A.A.: Learning data-driven reflectance priors for intrinsic image decomposition. In: Proceedings of the International Conference on Computer Vision (ICCV) (2015)
40. Zhou, Y., Berg, T.L.: Learning temporal transformations from time-lapse videos. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 262–277. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_16
41. Zhu, J.Y., et al.: Visual object networks: image generation with disentangled 3D representations. In: Neural Information Processing Systems (2018)