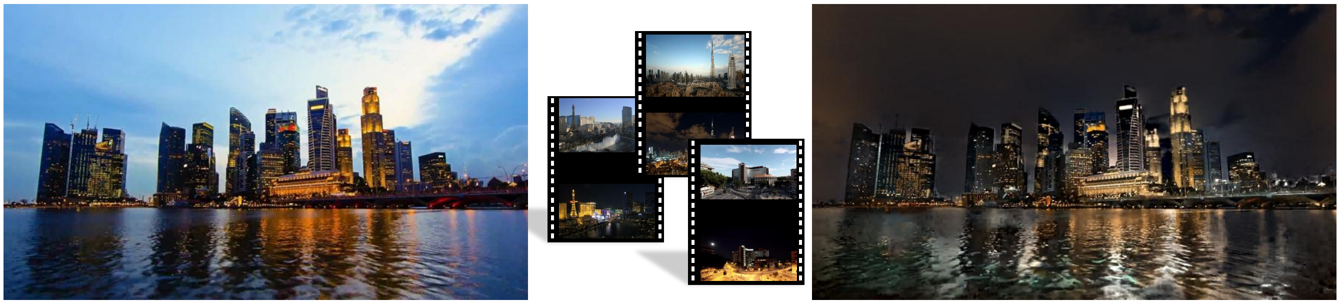
Data-driven Hallucination of Different Times of Day from a Single Outdoor Photo

Yichang Shih
MIT CSAIL

Sylvain Paris
Adobe

Frédo Durand
MIT CSAIL

William T. Freeman
MIT CSAIL



Input image at “blue hour” (just after sunset)

A database of time-lapse videos

Hallucinate at night

Figure 1: *Given a single input image (courtesy of Ken Cheng), our approach hallucinates the same scene at a different time of day, e.g., from blue hour (just after sunset) to night in the above example. Our approach uses a database of time-lapse videos to infer the transformation for hallucinating a new time of day. First, we find a time-lapse video with a scene that resembles the input. Then, we locate a frame at the same time of day as the input and another frame at the desired output time. Finally, we introduce a novel example-based color transfer technique based on local affine transforms. We demonstrate that our method produces a plausible image at a different time of day.*

Abstract

We introduce “time hallucination”: synthesizing a plausible image at a different time of day from an input image. This challenging task often requires dramatically altering the color appearance of the picture. In this paper, we introduce the first data-driven approach to automatically creating a plausible-looking photo that appears as though it were taken at a different time of day. The time of day is specified by a semantic time label, such as “night”.

Our approach relies on a database of time-lapse videos of various scenes. These videos provide rich information about the variations in color appearance of a scene throughout the day. Our method transfers the color appearance from videos with a similar scene as the input photo. We propose a *locally affine model* learned from the video for the transfer, allowing our model to synthesize new color data while retaining image details. We show that this model can hallucinate a wide range of different times of day. The model generates a large sparse linear system, which can be solved by off-the-shelf solvers. We validate our methods by synthesizing transforming photos of various outdoor scenes to four times of interest: daytime, the golden hour, the blue hour, and nighttime.

CR Categories: I.4.3 [Computing Methodologies]: Image Processing and Computer Vision—Enhancement

Keywords: Time hallucination, time-lapse videos

Links: DL PDF

1 Introduction

Time of day and lighting conditions are critical for outdoor photography (e.g. [Caputo 2005] chapter “Time of Day”). Photographers spend much effort getting to the right place at the perfect time of day, going as far as dangerously hiking in the dark because they want to reach a summit for sunrise or because they can come back only after sunset. In addition to the famous golden or magical hour corresponding to sunset or sunrise ([Rowell 2012] chapter “The Magical Hour”), the less-known “blue hour” can be even more challenging because it takes place after the sun has set or before it rises ([Rowell 2012] chapter “Between Sunset and Sunrise”) and actually only lasts a fraction of an hour when the remaining light scattered by the atmosphere takes a deep blue color and its intensity matches that of artificial lights. Most photographers cannot be at the right place at the perfect time and end up taking photos in the middle of the day when lighting is harsh. A number of heuristics can be used to retouch a photo with photo editing software and make it look like a given time of day, but they can be tedious and usually require manual local touch-up. In this paper, we introduce an automatic technique that takes a single outdoor photo as input and seeks to hallucinate an image of the same scene taken at a different time of day.

The modification of a photo to suggest the lighting of a different time of day is challenging because of the large variety of appearance changes in outdoor scenes. Different materials and different parts of a scene undergo different color changes as a function of reflectance, nearby geometry, shadows, etc. Previous approaches have leveraged additional physical information such as an external 3D model [Kopf et al. 2008] or reflectance and illumination inferred from a collection of photos of the same scene [Laffont et al. 2012; Lalonde et al. 2009].

In contrast, we want to work from a single input photograph and allow the user to request a different time of day. In order to deal with the large variability of appearance changes, we use two main strategies: we densely match our input image with frames from a time lapse database, and we introduce an edge-aware locally affine RGB mapping that is driven by the time-lapse data.

First, rather than trying to physically model illumination, we leverage the power of data and use a database of time lapse videos. Our videos cover a wide range of outdoor scenes so that we can handle many types of input scenes, including cityscape, buildings, and street views. We match the input image globally to time-lapse videos of similar scenes, and find a dense correspondence based on a Markov random field. For these steps, we use state-of-the-art methods in scene matching and dense correspondence, modified to fit our needs. These matches allow us to associate local regions of our input image to similar materials and scenes, and to output a pair of frames corresponding to the estimated time of the input and the desired times of day.

Second, given a densely-aligned pair of time-lapse frames obtained from our first strategy, we still need to address remaining discrepancies with our input, both because the distribution of object colors is never exactly the same and because scene geometry never allows perfect pixel alignment. If we apply traditional analogy methods such as Hertzmann et al. [2001] and Efros and Freeman [2001] designed to achieve a given output texture and simply copy the color from the frame at the desired time of day, the results exhibit severe artifacts. This happens because these methods do not respect the fine geometry and color of the input. Instead, our strategy to address variability is to transfer the *variation of color* rather than the output color itself. Our intuition is simple: if a red building turns dark red over time, transferring this time of day to a blue building should result in a dark blue. We leverage the fact that time lapse videos provide us with registered before-and-after versions of the scene, and we locally fit simple affine mappings from RGB to RGB. Because we use these models locally and because our first step has put our input in dense correspondence with a similar scene, we are able to use a simple parametric model of color change. This can be seen as a form of dimensionality reduction because the RGB-to-RGB *mappings* have less variability than the output RGB *distribution*. In addition, we need to make sure that the affine color changes are coherent spatially and respect strong edges of the image. We thus build on ideas from the matting [Levin et al. 2006] and intrinsic decomposition fields [Bousseau et al. 2009] and derive a Laplacian regularization. We perform the transfer by optimizing an L_2 cost function that simultaneously forces the output to be locally affine to the input, and that this affine model should locally explain the variation between the two frames in the retrieved time lapse. We derive a closed-form solution for the optimization, and show that this yields a sparse linear system.

Contributions Our contributions include the following:

- We propose the first time-of-day hallucination method that takes a single image and a time label as input, and outputs a gallery of plausible results.
- We introduce an example-based *locally affine model* that transfers the local color appearance variation between two time-lapse frames to a given image.

2 Related Work

Image Relighting and Color Transfer Deep Photo [Kopf et al. 2008] successfully relights an image when the geometric structure of the scene is known. Laffont et al. [2012] demonstrates that the intrinsic image derived from an image collection of the same scene enables the relighting of an image. In both cases, the key to producing high-quality results is the availability of scene-specific data. While this additional information may be available for famous landmarks, this data does not exist in many cases. Our system targets a more general case that does not need scene-specific data. It only relies on the availability of time-lapse videos of similar-looking scenes.

Approaches for color transfer such as [Reinhard et al. 2001; Pouli and Reinhard 2011; Pitie et al. 2005] apply a global color mapping to match color statistics between images. They work well in style transfer, but cannot be applied to time hallucination problem because the problem requires dramatic color appearance change. In comparison, our transfer is local and can distinguish the difference in color change between different image regions in the input even if they have a similar color. Our experiments show that our approach yields better results than global transfer.

Similarly to Lalonde et al. [2009], we use time-lapse data to study color appearance variation at different times of a day. Lalonde's work creates successful relit images by modeling the scene geometry manually. In contrast to their technique, our method hallucinates images by automatically transferring the color information from a time-lapse.

Example-based Image Colorization Example-based colorization [Irony et al. 2005] automatically generates scribbles from the example image onto the input gray image, and then propagates colors in a way that is similar to [Levin et al. 2004]. In our problem, the scene color appearance is usually different from the input, so the color palette in the time-lapse is not sufficient. For this, instead of direct copying the color palette from the example, we employ a locally affine model to synthesize the unseen pixels from the time-lapse.

Image Analogies Our work relates to Image Analogies [Hertzmann et al. 2001; Efros and Freeman 2001] in the sense that

input : hallucinated image :: matched frame : target frame

where the matched and target frames are from the time-lapse video. However, we cannot simply copy the patches from target frame onto input image, because the texture and color in input are different from time-lapse video. To accommodate the texture differences, we introduce the local affine models to transfer the color appearance from the time-lapse video to the input.

Image Collections Recent research demonstrates convincing graphics application with big data, such as scene completion [Hays and Efros 2007], tone adjustment [Bychkovsky et al. 2011], and super-resolution [Freeman et al. 2002]. Inspired by the previous success, our method uses a database of 495 time-lapse videos for time hallucination.

3 Overview of our method

The input to our algorithm is a single image of a landscape or a cityscape and a desired time of day. From these, we hallucinate a plausible image of the same scene as viewed at the specified time of day. Our approach exploits a database of time-lapse videos of landscapes and cityscapes seen as time passes (§ 4). This database is given a priori and independent of the user input, in particular, it does not need to contain a video of the same location as the input image.

Our method has three main steps (Fig. 2). First, we search the database for time-lapse videos of scenes that look like the input scene. For each retrieved video, we find a frame that matches the time of day of the input image and another frame at the target time of day (§ 5.1). We achieve these two tasks using existing scene and image matching techniques [Xiao et al. 2010].

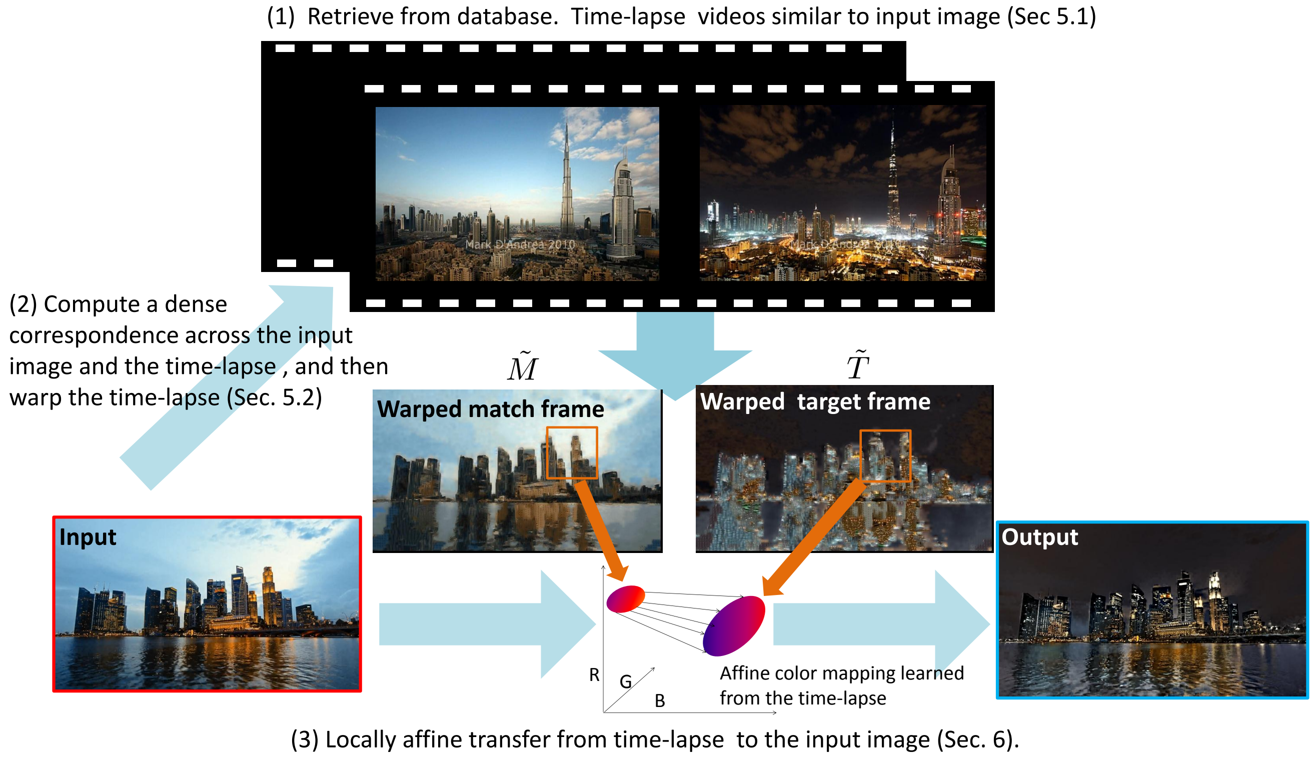

(1) Retrieve from database. Time-lapse videos similar to input image (Sec 5.1)
Mark D'Andrea 2010
(2) Compute a dense correspondence across the input image and the time-lapse , and then warp the time-lapse (Sec. 5.2)
$\tilde{M}$
Warped match frame
$\tilde{T}$
Warped target frame
Input
Output
R
G
B
Affine color mapping learned from the time-lapse
(3) Locally affine transfer from time-lapse to the input image (Sec. 6).

Figure 2: *Our approach has three steps. (1) We first retrieve videos of similar scene with input (§ 5.1), and then (2) find the local correspondence between the input and the time-lapse (courtesy of Mark D'Andrea) (§ 5.2). (c) Finally we transfer the color appearance from the time-lapse to the input (§ 6).*

Next, to locally transfer the appearance from the time-lapse videos, we need to locally match the input and each video. We employ a Markov random field to compute a dense correspondence for each time-lapse video (§ 5.2). We then warp the videos to match the input at the pixel level.

Finally, we generate a gallery of hallucinated results, one for each retrieved time-lapse video. To transfer the appearance variations of a time-lapse video onto the input image, we introduce an example-based transfer technique that models the color changes using local affine transforms (§ 6). This model learns the mapping between the output and input from the time-lapse video, and preserves the details of the input.

4 Database and Annotation

Our database contains 450 time-lapse videos, covering a wide range of landscapes and cityscapes, including city skyline, lake, and mountain view. In the supplemental materials, we will show a mosaic of all the scenes in the database Unlike most web-cam clips [Lalonde et al. 2009] or surveillance camera videos [Jacobs et al. 2007], our time-lapse videos are taken with high-end setups, typically a DSLR camera on a sturdy tripod, that are less prone to over-and under-exposure, defocus, and accidental shake.

The most interesting lighting for photographers are daytime, golden hour, blue hour (occurs between golden hour and night), and night-time [Caputo 2005]. For each time-lapse, we label the transition time between the above four different lightings, so that the user can specify the hallucination time by these semantic time labels.

5 Matching Between the Input Image and Time-lapse Data

The first step of our algorithm is to determine the correspondence between the input image and the time-lapse data. We first find a set of time-lapse videos with a similar scene as the input image, and then compute a dense correspondence between the input image for each matched time-lapse video.

5.1 Global Matching

The first step of our algorithm is to identify the videos showing a scene similar to the given input image. We employ a standard scene matching technique in computer vision, adapting the code from Xiao et al. [2010] to time-lapse data. We sample 5 regularly spaced frames from each video, and then compare the input to all these sampled frames. To assign a score to each time-lapse video, we use the highest similarity score in feature space of its sampled frames. We tried the different descriptors suggested in Xiao's paper, and found that the Histograms of Oriented Gradients (HOG) [Dalal and Triggs 2005] works well for our data. We show some sample retrieval results in the supplemental document.

Now that we have a set of matching videos, for each of them, we seek to retrieve a frame that matches the time of day of the input image. We call this frame the *matched frame*. Since we already selected videos with a similar content as the input image, this is a significantly easier task than the general image matching problem. We use the color histogram and L_2 norm to pick the matched frame. We show sample results in supplementary document. Our approach finding matching videos and frames produced good results for our database but we believe that other options may also work well.

5.2 Local Matching

We seek to pair each pixel in the input image I with a pixel in the match frame M. As shown in Fig. 10, existing methods such as PatchMatch [Barnes et al. 2010] and SIFT Flow [Liu et al. 2008] do not produce satisfying result because they are designed to match with a single image and are not designed for videos. We propose a method exploiting the additional information in a time-lapse video by constraining the correspondence field along time. For this, we formulate the problem as a Markov random field (MRF) using a data term and pairwise term.

Similarly to PatchMatch and SiftFlow, for each patch in I, we seek a patch in M that looks similar to it. This is modeled by the data term of the MRF. We use the L_2 norm over square patches of side length $2r+1$. Formally, for pixels $p \in I$ and the corresponding pixel $q \in M$, our data term is:

$$E_1 = \sum_{i=-r}^{+r} \sum_{j=-r}^{+r} \left\| I(x_p+i, y_p+j) - M(x_q+i, y_q+j) \right\|^2 \quad (1)$$

We then leverage the information provided in a time-lapse video. Intuitively, we want the adjacent patches to look similar at any time of the video. This is captured by the pairwise term of the MRF. Formally, we introduce the following notations. For two adjacent pixels p_i and p_j in I, we name Ω the set of the overlapping pixels between the two patches centered at p_i and p_j. For each pixel $o \in \Omega$, we define the offsets $\delta_i = o - p_i$ and $\delta_j = o - p_j$. For the energy we use L_2 norm within each frame t, but L_∞ norm across frames so that the assigned compatibility score corresponds to the worst case over the video V. This gives the pairwise term as:

$$E_2(q_i, q_j) = \max_t \sum_{o \in \Omega} \left\| V_t(q_i + \delta_i) - V_t(q_j + \delta_j) \right\|^2 \quad (2)$$

Denoting λ parameter controlling the importance of the compatibility term compared to the data term, N_i the neighboring pixels of i, one could find q by trying to minimize the energy:

$$\sum_{i \in I} E_1(p_i, q_i) \quad + \quad \lambda \sum_{i \in I, j \in N_i} E_2(q_i, q_j) \quad (3)$$

by considering all possible pairings between a pixel in I with a pixel in V. However, this would be impractical because of the sheer number of possible assignments. We now explain below how to select a small number of candidate patches so that the optimization of Equation 3 becomes tractable.

Candidate Patches A naive way to select a few candidate patches for each location would be to pick the top n patches according to the data term E_1. However, this tends to return patches that are clustered around a small number of locations. This lack of diversity later degrades the transfer. Instead of picking the top candidates, we randomly sample the candidates according to the probability:

$$\frac{1}{Z} \exp\left(-\frac{E_1}{2\sigma^2} \right) \quad (4)$$

where Z is a normalization factor and σ controls how diverse the sampled patches are. This strategy yields a candidate set with more variety, which improves the transfer quality. In practice, we sample 30 patches, and use $\lambda = 0.5$ and $\sigma = 20$. We minimize Equation 3 using Belief Propagation [Yedidia et al. 2000].

Discussion Our sampling strategy is akin to that proposed by Freeman et al. [2000], except that we do not explicitly enforce diversity as they do. Testing their approach in our context would be interesting, but since we obtained satisfying results with the approach described above, we leave this to future work.

6 Locally Affine Color Transfer

The core of our method is the example-based locally affine color transfer. The transfer starts from the input image I, the warped match frame $\tilde{M}$, the warped target frame $\tilde{T}$, and output the hallucinated image O (See Figure 2).

We design the transfer to meet two goals:

- We want it to explain the color variations observed in the time-lapse video. We seek a series of affine models $\{\mathbf{A}_k\}$ that locally describe the color variations between $\tilde{T}$ and $\tilde{M}$.
- We want a result that has the same structure as the input and that exhibits the same color change as seen in the time-lapse video. We seek an output O that is locally affine to I, and explained by the same affine models $\{\mathbf{A}_k\}$.

A naive solution would be to compute each affine model $\mathbf{A}_k$ as a regression between the k^{th} patch of $\tilde{M}$ and its counterpart in $\tilde{T}$, and then independently apply $\mathbf{A}_k$ to the k^{th} patch of I for each k. However, the boundary between any two patches of O would not be locally affine with respect to I, and would make O have a different structure from I, e.g., allows for spurious discontinuities to appear at patch boundaries. Instead of this naive approach, we formulate this problem as a least-squares optimization that seeks local affinity *everywhere* between O and I. We also specifically account for the possibility of the data of being corrupted by noise and compression artifacts.

6.1 L_2-optimal locally affine model

We use a matrix formulation to describe our approach. We use $\mathbf{v}_k(\cdot)$ to denote the k^{th} patch of an image given in argument. For a patch containing N pixels, $\mathbf{v}_k(\cdot)$ is a $3 \times N$ matrix, each column representing the color of a pixel as $(r, g, b)^\mathsf{T}$. We use $\bar{\mathbf{v}}_k(\cdot)$ to denote the patch augmented by ones, i.e., $4 \times N$ matrix where each column is $(r, g, b, 1)^\mathsf{T}$. The local affine functions are represented by 3×4 matrices, $\mathbf{A}_k$. With this notation, the first term in our energy models the need for the $\mathbf{A}_k$ matrices to transform $\tilde{M}$ into $\tilde{T}$. With a least-squares formulation using the Frobenius norm $\|\cdot\|_\mathsf{F}$, i.e., the square root of the sum of the squared coefficients of a matrix, this gives:

$$\sum_k \left\| \mathbf{v}_k(\tilde{T}) - \mathbf{A}_k \, \bar{\mathbf{v}}_k(\tilde{M}) \right\|_\mathsf{F}^2 \quad (5)$$

We also want the output patches to be well explained by the input patches transformed by the $\mathbf{A}_k$ matrices:

$$\sum_k \left\| \mathbf{v}_k(O) - \mathbf{A}_k \, \bar{\mathbf{v}}_k(I) \right\|_\mathsf{F}^2 \quad (6)$$

Finally, we add a regularization term on the $\mathbf{A}_k$ matrices for the case when Equation 5 is under-constrained e.g., $\mathbf{v}_k(\tilde{M})$ is constant. For this we regularize $\mathbf{A}_k$ using a global affine model $\mathbf{G}$, the regression by the entire picture of $\tilde{M}$ and $\tilde{T}$, with the Frobenius norm. Formally, we solve

$$O = \arg\min_{O, \{\mathbf{A}_k\}} \sum_k \left\| \mathbf{v}_k(O) - \mathbf{A}_k \, \bar{\mathbf{v}}_k(I) \right\|^2 + \epsilon \sum_k \left\| \mathbf{v}_k(\tilde{T}) - \mathbf{A}_k \, \bar{\mathbf{v}}_k(\tilde{M}) \right\|^2 + \gamma \sum_k \left\| \mathbf{A}_k - \mathbf{G} \right\|_\mathsf{F}^2 \quad (7)$$

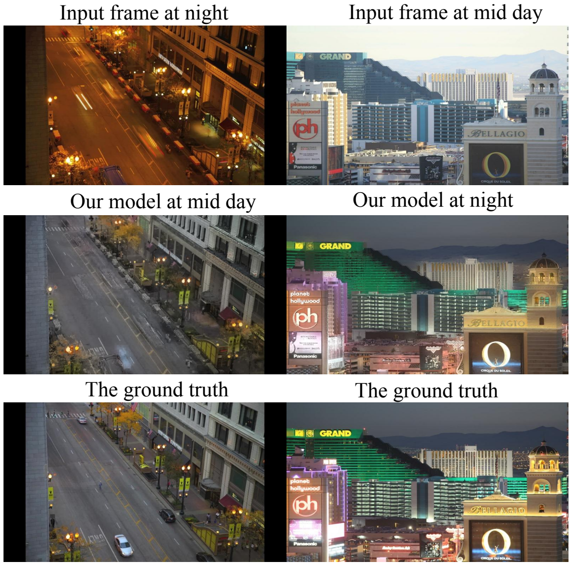


Figure 3: *Our locally affine model is expressive enough to approximate dramatic color change, such as night-to-day and day-to-night (left and right column). As a sanity check, we pick two frames from the same video at different times as input and target. We also use the input as matched frame and apply our model. In this situation, if local affine transforms correctly model the color changes, the output should closely match the target, which the ground truth in this scenario. Our result shows that this is the case.*

where ϵ and γ control the relative importance of each term.

Discussion Equation 5 alone would correspond to standard local linear regression. With such formulation, overlapping affine transforms would be independent from each other and they could potentially predict widely different values for the same pixel. With Equation 6, overlapping transforms are explicitly constrained to produce consistent values, which forces them to produce a result coherent over the whole image.

Closed-Form Solution In this section, we derive a closed-form solution for Equation 7. We follow a strategy similar to Levin et al. [2006] and Bousseau et al. [2009] and remove the $\mathbf{A}_k$ functions from the equations by expressing them as a function of the other variables. That is, assuming that O is known, Equation 7 becomes a standard linear least-squares optimization problem with the $\mathbf{A}_k$ matrices as unknowns. Denoting $\mathbf{Id}_n$ an $n \times n$ identity matrix, this leads to:

$$\begin{aligned}\mathbf{A}_k = & \left(\mathbf{v}_k(O)\bar{\mathbf{v}}_k(I)^\mathsf{T} + \epsilon \mathbf{v}_k(\tilde{T})\bar{\mathbf{v}}_k(\tilde{M})^\mathsf{T} + \gamma \mathbf{G}\right) \\ & \left(\bar{\mathbf{v}}_k(I)\bar{\mathbf{v}}_k(I)^\mathsf{T} + \epsilon \bar{\mathbf{v}}_k(\tilde{M})\bar{\mathbf{v}}_k(\tilde{M})^\mathsf{T} + \gamma \mathbf{Id}_4\right)^{-1} \qquad (8)\end{aligned}$$

Then, defining $\mathbf{B}_k = \left(\bar{\mathbf{v}}_k(I)\bar{\mathbf{v}}_k(I)^\mathsf{T} + \epsilon \bar{\mathbf{v}}_k(\tilde{M})\bar{\mathbf{v}}_k(\tilde{M})^\mathsf{T} + \gamma \mathbf{Id}_4\right)^{-1}$, a minimizer of Equation 7 is:

$$\begin{aligned} O &= \mathbf{M}^{-1}\,\mathbf{u} \\ \text{with: } \mathbf{M} &= \textstyle\sum_k \mathrm{lift}_k\left(\mathbf{Id}_N - \bar{\mathbf{v}}_k(I)^\mathsf{T}\mathbf{B}_k\bar{\mathbf{v}}_k(I)\right) \\ \mathbf{u} &= \textstyle\sum_k \mathrm{lift}_k\left(\left(\epsilon \mathbf{v}_k(\tilde{T})\bar{\mathbf{v}}_k(\tilde{M})^\mathsf{T} + \gamma \mathbf{G}\right)\mathbf{B}_k\bar{\mathbf{v}}_k(I)\right)\end{aligned}$$

where $\mathrm{lift}_k(\cdot)$ is an operator that lifts matrices and vectors expressed in the local indexing system of the k^{th} patch into larger matrices and vectors indexed in the global system of the image.

Model Expressivity We demonstrate the expressivity of our model by taking a frame from a time-lapse as input, and hallucinating to another time using the same time-lapse. In Figure 3 we show this model can express dramatic color appearance, such as day-to-night and night-to-day. We test on various scenes in the supplemental materials. For all results in this paper, we use $\epsilon = 0.01$, $\gamma = 1$ (pixel value $\in [0, 255]$), $N = 25$ (5×5 patch). We compare the choice of affine model versus linear model in the supplemental materials. The residuals show locally affine model is better than linear model.

Link with Illumination Transfer If the patches in I and the warped time-lapse are Lambertian, then our method becomes illumination transfer. In this case, the local affine model degenerates to diagonal matrix with the last row equal to zeros. The non-zero components are the quotient of the illuminations between the target and the match frame. For non-Lambertian patches, such as sky and water, our method produces visually pleasing results by using non-diagonal components in the model.

Link with the Matting Laplacian $\mathbf{M}$ in Equation 9 is similar to the Matting Laplacian [Levin et al. 2006], except that the local scaling factor $\mathbf{B}_k$ is $\left(\mathbf{v}_k(I)^\mathsf{T}\mathbf{v}_k(I) + \epsilon \mathbf{v}_k(\tilde{M})^\mathsf{T}\mathbf{v}_k(\tilde{M}) + \gamma \mathbf{Id}_k\right)^{-1}$ whereas for the Matting Laplacian, it is $\left(\mathbf{v}_k(I)^\mathsf{T}\mathbf{v}_k(I) + \gamma \mathbf{Id}_k\right)^{-1}$. That is, in addition to the covariance of the input data, our method also accounts for the covariance of the example data.

6.2 Dealing with Noisy Input

The affine mapping has a side effect that it may magnify the noise existing at the input image, such as sensor noise or quantization noise. This problem usually appears when the affine model is under-constrained, which may lead into large coefficients in the affine model. We propose a simple yet effective solution to avoid the noise magnification. We first use bilateral filtering to decompose the input image into a detail layer and a base layers, the latter being mostly noise-free. We then apply our locally affine transfer to the base layer instead of the input image. Finally, we obtain the final result by adding the detail layer back to the transferred base layer. Since the base layer is clean, the noise is not magnified. Compared to directly taking the input image, we significantly reduce the noise, as shown in Figure 4.

7 Results and Comparison

Figure 5 illustrates the result of our transferring approach, which transfers the color changes between the target and matched frame to the input. The result produced by our method is more visually pleasing than using only the target frame.

Figure 6 shows our method applied to two day-time images. For each of the two images, we hallucinate 4 times of day: “day”, “golden hour” (i.e., just before sunset), “blue hour” (i.e., just after sunset), and “night”. We use the top two time-lapse videos retrieved in our database, each produces a different plausible hallucination, thereby enabling the exploration of various possible renditions of the desired time of day. These results at 4 times of day illustrate the ability of our approach to cope with dramatic appearances. We observed that the appearance of city-scape time-lapse usually has larger variability than natural landscape, and so the renditions produced by cityscape input usually have more variations. Figure 7 shows the hallucination works from various scenes. Figure 8 show that our approach also handles input images taken at different times of day.

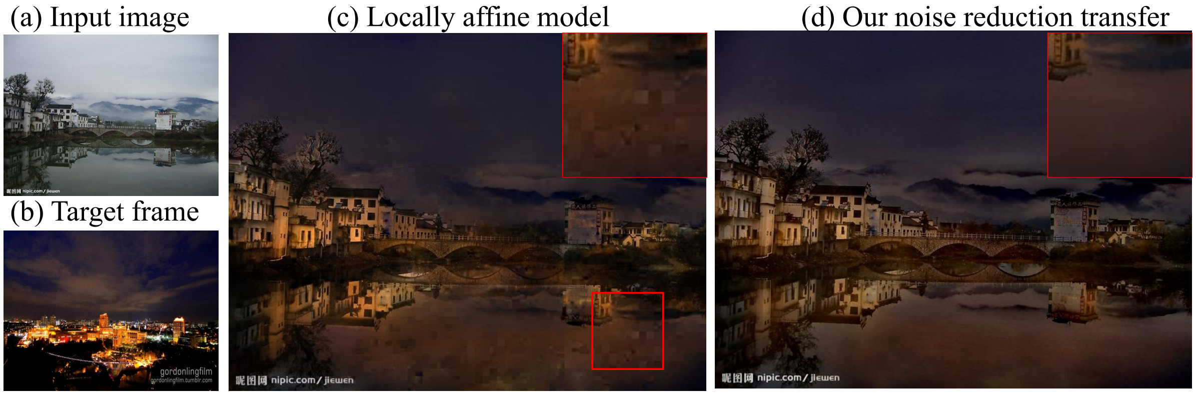
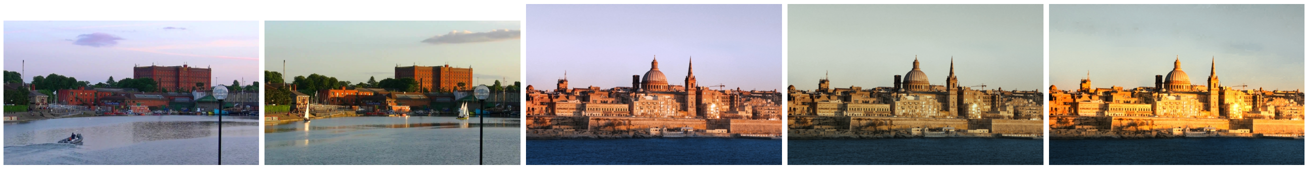


Figure 4: *The noise in JPEG input (a) results in artifact at the output of locally affine model (c). Our noise-robust affine model significantly reduces the noise (d). Image courtesy of Jie Wen (a) and Reanimated Studio https://vimeo.com/34362088 (b).*



(a) matched frame (b) target frame (c) input (d) Photoshop Match Color (e) our result

Figure 5: *Producing a golden hour rendition of a scene that contains warm colors (c) using a direct color transfer from image (b) generates a weak effect (d). We created this result with the Photoshop Match Color function. In comparison, our approach transfers the color transformation between the matched frame (a) and the target frame (b) and captures the strong color change characteristic of the golden hour (e).*

Figure 9 compares our hallucinated image to an actual photo of the same scene, and shows that, while our result is different, it is nevertheless plausible. In the supplemental material, we provide the result of our technique applied to all the landscapes and cityscapes within the first 101 images of the MIT-Adobe 5K dataset [Bychkovsky et al. 2011].

Figure 10 shows that in our context, our MRF-based method to compute the dense correspondence field performs better than PatchMatch [Barnes et al. 2010] and SIFT Flow [Liu et al. 2008]. This is because we exploit the information across the time-lapse frames, as opposite to only using the target frame. Figure 11 demonstrates that our local affine transform model preserves image details better than an edge-aware filter like the Joint Bilateral Filter [Eisemann and Durand 2004; Petschnigg et al. 2004] or the Guided Filter [He et al. 2010].

Performance We measure the average performance using 16 inputs in MIT-Adobe 5K dataset [Bychkovsky et al. 2011]. We scale all input images to a 700-pixels width. For each input, the matching takes 25 seconds total, split into 23 seconds for local matching and 2 seconds for global matching. For each hallucinated result, the transfer takes 32 seconds. We use conjugate gradient descent in Matlab and incomplete Cholesky decomposition as a preconditioner to solve the linear system.

7.1 Comparison to Previous Work

Figure 12 compares our approach to techniques based on a global color transfer [Reinhard et al. 2001; Pitie et al. 2005]. While these methods succeed to some degree, their results are not always as accurate as ours. In comparison, our results are cleaner. The local nature of our approach allows it to make better matches, e.g., sky to sky and building to building.

We also tried to compare with the technique of HaCohen et al. [2011] that first finds dense correspondences and then performs a parametric color transfer. We found their method is not applicable in our case, because our target frame is a different scene from the input image. For all the examples in Results section, their implementation reported that no match was found.

Another thread in recent research that demonstrates successful image illumination transfer uses rich information of the scene, such as Deep Photo, which leverages depth map and texture of the scene [Kopf et al. 2008], or Laffont et al. [2012], which uses intrinsic image and illumination from a collection of images of the same scene. In supplemental material, we show that our results are on par with these methods even though our approach uses a generic database of time-lapse videos instead of scene-specific data.

Discussion While the methods of Pitié et al. [2005] and Reinhard et al. [2001] directly transfer the colors of the target image, our approach transfers the color transformation from the matched frame to the target frame. This may produce less intuitive outputs than a direct color transfer. However, in practice, users do not see the target frame and as a consequence, have no expectation to match its look. And, more importantly, transferring the color transformation allows us to be less sensitive to the image content. For instance, Figure 5 shows that a direct color transfer produces a weak golden hour look because it ignores that the input photo has a content that contains warm colors. In comparison, our approach transfers the color transformation and warms up the image a lot more, which corresponds to the change observed in the time-lapse video, and produces a more convincing golden hour rendition.

User Study A successful hallucinated image should look natural to a human observer. Inspired by image inpainting [Hays and Efros 2007], we performed a user study to quantitatively evaluate whether

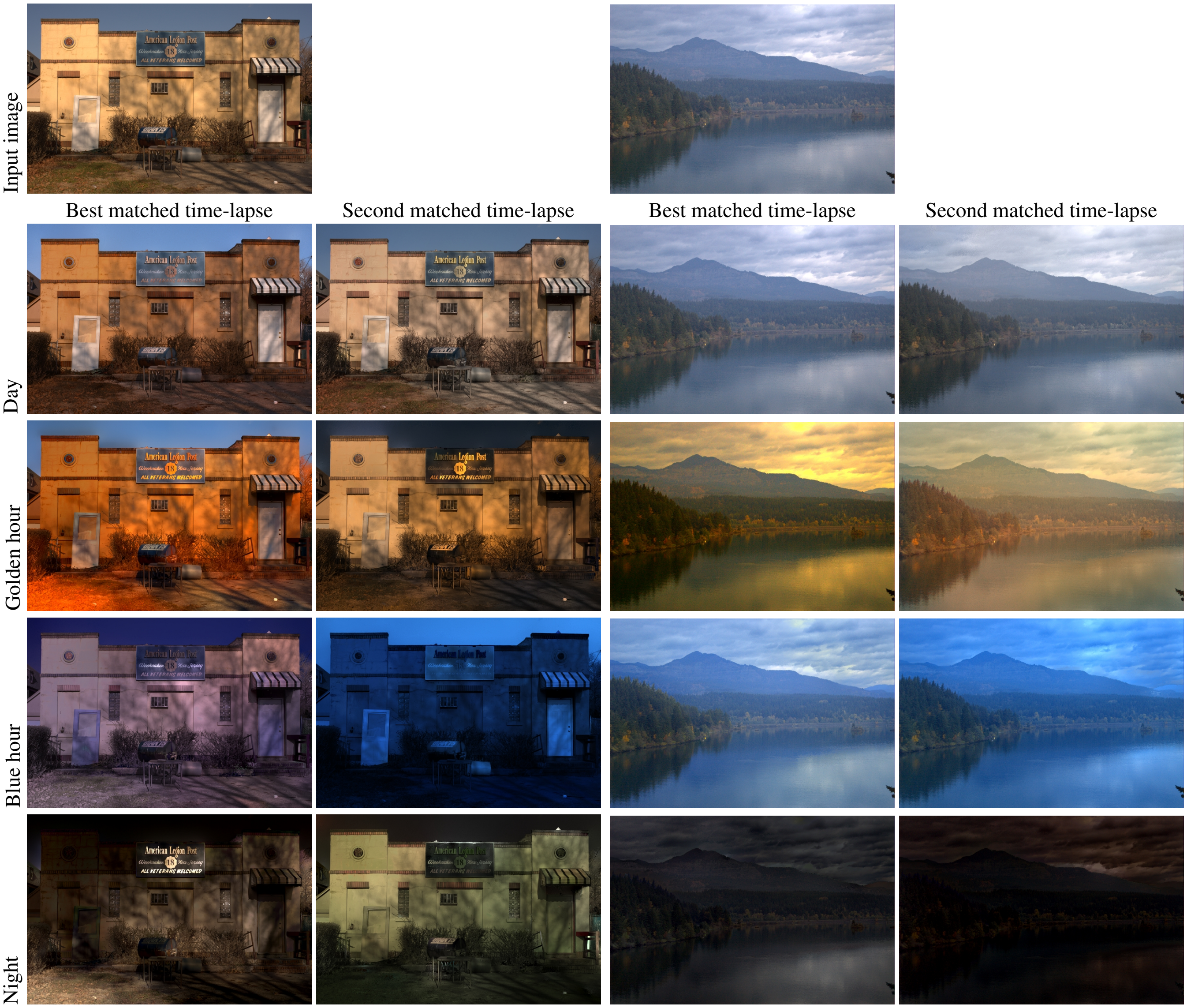


Figure 6: *We hallucinate the house and lake at four different times of day. Each time, we show the results for two retrieved videos.*

human observers believe our results are real images.

We performed the study with 9 images randomly selected from 9 different time-lapse video. For each image, we randomly selected 6 or 7 target frames from the top 10 retrieved videos. Then we generated hallucinated images with our approach and Reinhard's method [2001]. As baseline, we randomly selected 6 or 7 frames from the input image's time-lapse video. In total, we used 59 results of 9 different scenes for each method. We then mixed the output from our method, Reinhard's technique with real time-lapse frames, and randomized the order. For each image, we ask 5 testers if the image is real or fake.

We performed this task on Amazon Mechanic Turk. 55.2% of our results were classified real. In comparison, the percentage was 66.4% for the real time-lapse frames and 48.8% for Reinhard's method [2001]. As expected our approach does not perform as well as actual video frames, but, nonetheless users prefer our method to Reinhard's method.

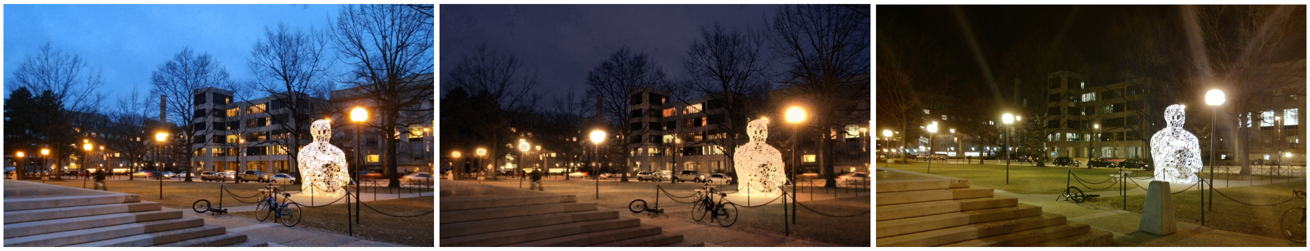
7.2 Applications

In addition to time hallucination, our method can be used for different graphics applications.

Lighting and Weather Transfer In Figure 13, the matched and target frames are selected close in time but the target is more sunny. Our algorithm successfully transfers the sunshine to the input image to create a sunny output.

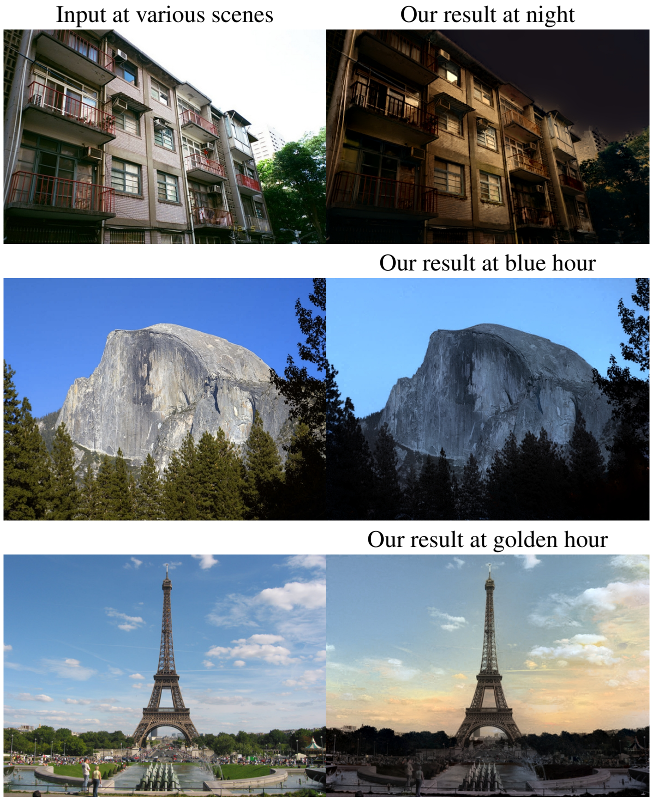
Similarly, we can transfer weather conditions by choosing a target with a different weather from the input. In Figure 14, we create a cloudy image from a sunny input by transferring the color properties of a cloudy target image.

Hallucinating Paintings Figure 15 shows that our approach also applies to paintings, even though our method is designed for realistic photo.



(a) input (b) our hallucinated result (c) actual night photo of the same scene

Figure 9: *We hallucinate a photo at night, and compare to a reference photo at the same location at night. Our result (b) is different from the actual photo (c) but nonetheless looks plausible.*



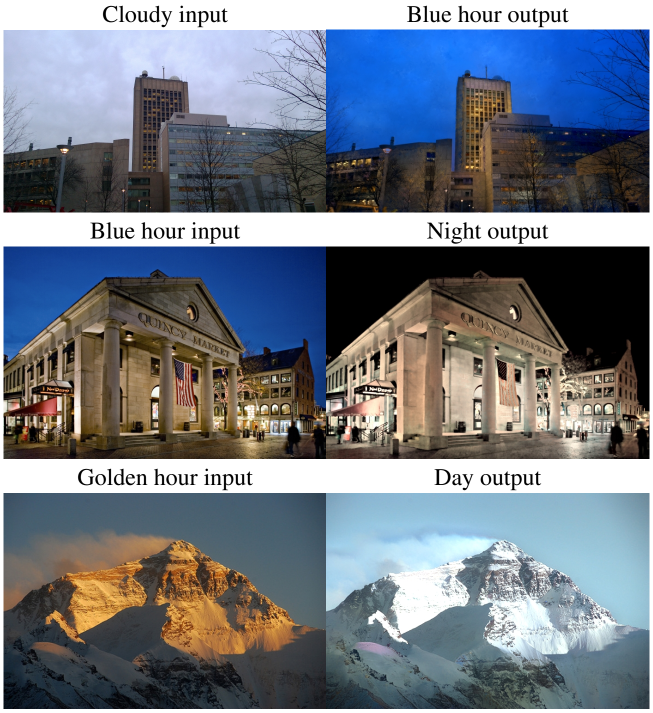
Input at various scenes

Our result at night

Our result at blue hour

Our result at golden hour

Figure 7: *Our approach works for various scenes, including a building, a mountain, and a famous landmark. The dramatic changes for different times of day are visually plausible.*



Cloudy input

Blue hour output

Blue hour input

Night output

Golden hour input

Day output

Figure 8: *Our method can take input at various time of day: cloudy, blue hour, and golden hour.*

Synthetic Time-lapse Video By interpolating between the hallucinations at four different times, we generate continuous lighting changes. We include several examples in supplemental video. We envision that this could also be used to enable users to choose an arbitrary time of day, e.g., with a slider that selects a frame of the synthetic time-lapse video.

8 Discussion and Conclusion

The main novelty of this paper is the idea of leveraging time-lapse database for light transfer. Compared to data-driven image completion which leverages millions images [Hays and Efros 2007], it is surprising that with only 450 videos we can achieve convincing results. This is due to our contributions in a example-based locally affine model.

Limitation Our method still has some limitations. If an object is not static or nearly static in the scene, there may be problems finding correspondences. For example, time-lapse videos do not have humans in the scene, so we do not have a proper model for human skin. Moving clouds in the sky can also cause flickering when synthesizing a new time-lapse video with our method using frame-by-frame transfer. Picking a few keyframes and interpolating between them would perform better as shown in the companion video, but the motion of the clouds would still not be captured.

Our method can hallucinate results that, while visually plausible, may not be physically accurate, for example, shadows and highlights that are not consistent. Even if an hallucination is technically successful, the result may not always be visually pleasing. For instance, landscapes at night may be overly dark due to the lack of lights.

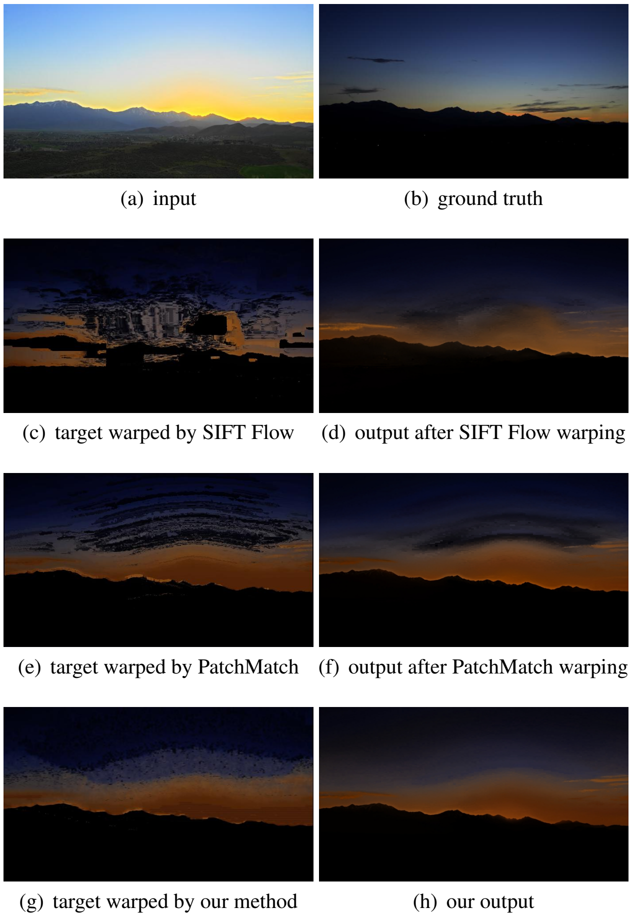

(a) input
(b) ground truth
(c) target warped by SIFT Flow
(d) output after SIFT Flow warping
(e) target warped by PatchMatch
(f) output after PatchMatch warping
(g) target warped by our method
(h) our output

Figure 10: *We picked a frame in a time-lapse video (a) and hallucinate it at night. We compare the warped target frame and the final output using PatchMatch [Barnes et al. 2010], SIFT Flow [Liu et al. 2008], and our approach. Since the input comes from a time-lapse video, we can also compare to ground truth (b). Warping the target frame using PatchMatch or SIFT Flow produces unsightly discontinuities (c,e) that are still visible in the final outputs (d,f). In comparison, our algorithm does not introduce strong discontinuities in the warped frame (g) and produces a better result (h). While none of the outputs (d,f,h) is similar to the ground truth (b), ours is more plausible and visually more pleasing.*

The ability to choose among several results rendered from different time-lapse videos helps mitigate these issues.

Future Work Our method can be applied to many graphic applications. For example, in scene completion and image-based rendering, our approach could hallucinate images from different times of a day into a similar time as a pre-processing step.

Beyond the graphics application, perhaps a deeper question is this: *can we learn the image feature evolution along time by observing enough time-lapse data?* We are excited at more research using time-lapse data.

Acknowledgements

We thank Jianxiong Xiao for the help and advice in scene matching code, SIGGRAPH ASIA reviewers for their comments, and acknowledge the funding from NSF No.0964004 and NSF CGV-1111415.

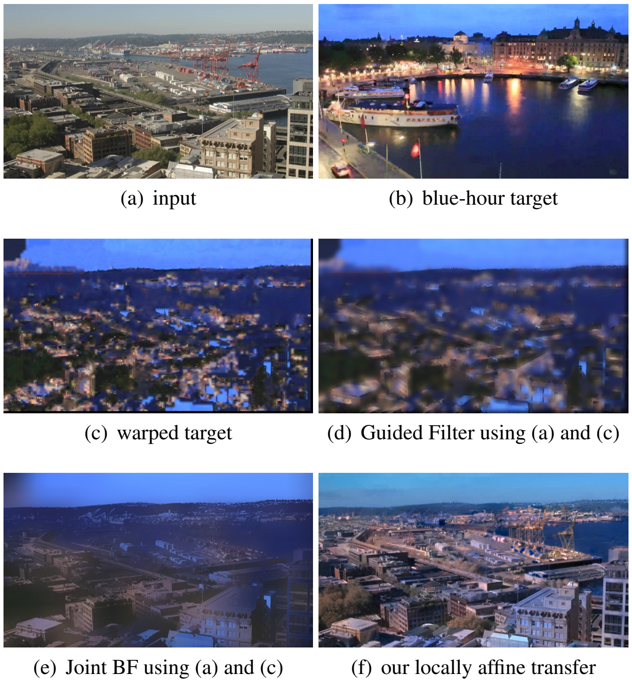

(a) input
(b) blue-hour target
(c) warped target
(d) Guided Filter using (a) and (c)
(e) Joint BF using (a) and (c)
(f) our locally affine transfer

Figure 11: *We compare out model to the Joint Bilateral Filter [Eisemann and Durand 2004; Petschnigg et al. 2004] and the Guided Filter [He et al. 2010]. For these filters, we use the warped target as the input, and the original input as guidance. In both cases, the results exhibit significant loss of details. In comparison, our approach produces sharp outputs.*

References

BARNES, C., SHECHTMAN, E., GOLDMAN, D., AND FINKELSTEIN, A. 2010. The generalized patchmatch correspondence algorithm. *Computer Vision–ECCV 2010*, 29–43.

BOUSSEAU, A., PARIS, S., AND DURAND, F. 2009. User-assisted intrinsic images. In *ACM Transactions on Graphics (TOG)*, vol. 28, ACM, 130.

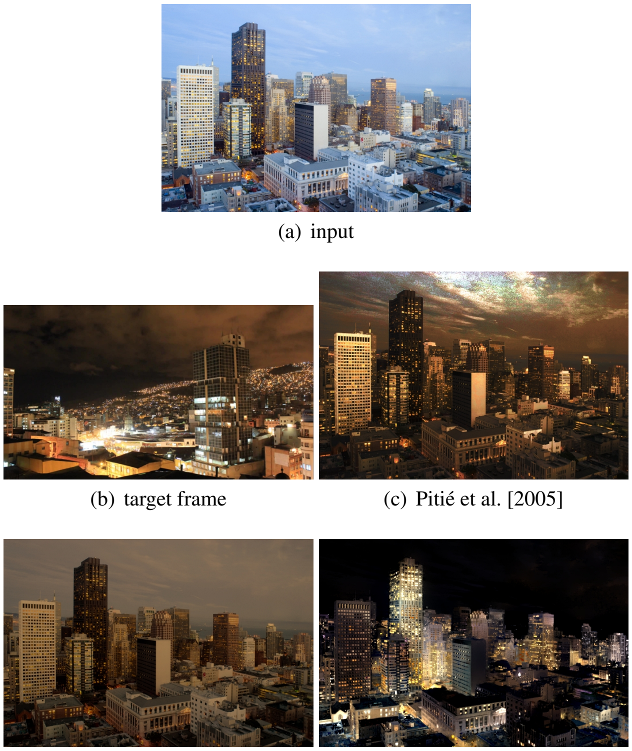
BYCHKOVSKY, V., PARIS, S., CHAN, E., AND DURAND, F. 2011. Learning photographic global tonal adjustment with a database of input/output image pairs. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 97–104.

CAPUTO, R. 2005. In *Potography field guide*, National Geographics, 104–115.

DALAL, N., AND TRIGGS, B. 2005. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, IEEE, 886–893.

EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. In *ACM SIGGRAPH*. In *Computer Graphics* Proceedings, Annual Conference Series.

EISEMANN, E., AND DURAND, F. 2004. Flash photography enhancement via intrinsic relighting. In *ACM Transactions on Graphics (TOG)*, vol. 23, ACM, 673–678.



(a) input

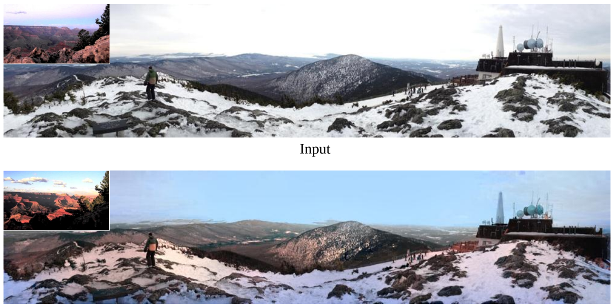
(b) target frame

(c) Pitié et al. [2005]

(d) Reinhard et al. [2001]

(e) our result

Figure 12: *Global methods generate only moderately convincing results (c,d). In comparison, our local affine transforms provide more flexibility in modeling spatially varying color changes, which produces a better result (e).*



Input

Light transfer

Figure 13: *The two target frames shown in the insets are taken at close times but under different lighting conditions. Our method increase the vibrancy by transferring the lighting to an ordinary photo.*

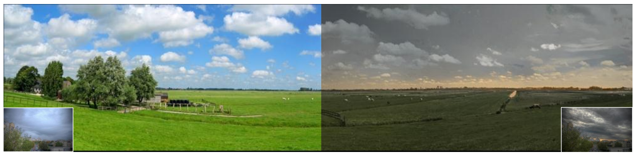


Figure 14: *We hallucinate the weather for the right half of this panorama. We transfer the difference between two kinds of weather in the time-lapse to a photo.*

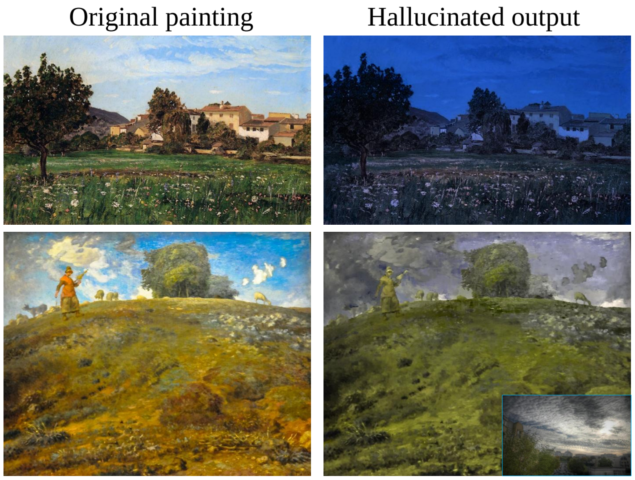


Figure 15: *Paintings in realism. From top to bottom:- "In the Auvergne", Jean-Francois Millet. "Lourmarin", Paul-Camille Guigou. We hallucinate the top one to blue hour, and handpick a cloudy frame for the bottom one.*

FREEMAN, W., PASZTOR, E., AND CARMICHAEL, O. 2000. Learning low-level vision. *International journal of computer vision 40*, 1, 25–47.

FREEMAN, W., JONES, T., AND PASZTOR, E. 2002. Example-based super-resolution. *Computer Graphics and Applications, IEEE 22*, 2, 56–65.

HACOHEN, Y., SHECHTMAN, E., GOLDMAN, D., AND LISCHINSKI, D. 2011. Non-rigid dense correspondence with applications for image enhancement. *ACM Transactions on Graphics (TOG) 30*, 4, 70.

HAYS, J., AND EFROS, A. 2007. Scene completion using millions of photographs. In *ACM Transactions on Graphics (TOG)*, vol. 26, ACM, 4.

HE, K., SUN, J., AND TANG, X. 2010. Guided image filtering. *Computer Vision–ECCV 2010*, 1–14.

HERTZMANN, A., JACOBS, C., OLIVER, N., CURLESS, B., AND SALESIN, D. 2001. Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM, 327–340.

IRONY, R., COHEN-OR, D., AND LISCHINSKI, D. 2005. Colorization by example. In *Proceedings of the Sixteenth Eurographics conference on Rendering Techniques*, Eurographics Association, 201–210.

JACOBS, N., ROMAN, N., AND PLESS, R. 2007. Consistent temporal variations in many outdoor scenes. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, IEEE, 1–6.

KOPF, J., NEUBERT, B., CHEN, B., COHEN, M., COHEN-OR, D., DEUSSEN, O., UYTTENDAELE, M., AND LISCHINSKI, D. 2008. Deep photo: Model-based photograph enhancement and viewing. In *ACM Transactions on Graphics (TOG)*, vol. 27, ACM, 116.

LAFFONT, P.-Y., BOUSSEAU, A., PARIS, S., DURAND, F., DRETTAKIS, G., ET AL. 2012. Coherent intrinsic images from photo collections. *ACM Transactions on Graphics 31*, 6.

LALONDE, J., EFROS, A., AND NARASIMHAN, S. 2009. Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. In *ACM Transactions on Graphics (TOG)*, vol. 28, ACM, 131.

LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2004. Colorization using optimization. In *ACM Transactions on Graphics (TOG)*, vol. 23, ACM, 689–694.

LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2006. A closed form solution to natural image matting. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, IEEE, 61–68.

LIU, C., YUEN, J., TORRALBA, A., SIVIC, J., AND FREEMAN, W. 2008. Sift flow: Dense correspondence across different scenes. *Computer Vision–ECCV 2008*, 28–42.

PETSCHNIGG, G., SZELISKI, R., AGRAWALA, M., COHEN, M., HOPPE, H., AND TOYAMA, K. 2004. Digital photography with flash and no-flash image pairs. In *ACM Transactions on Graphics (TOG)*, vol. 23, ACM, 664–672.

PITIE, F., KOKARAM, A., AND DAHYOT, R. 2005. N-dimensional probability density function transfer and its application to color transfer. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2, IEEE, 1434–1439.

POULI, T., AND REINHARD, E. 2011. Progressive color transfer for images of arbitrary dynamic range. *Computers & Graphics 35*, 1, 67–80.

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

ROWELL, G. 2012. In *Mountain Light*, Sierra Club Books.

XIAO, J., HAYS, J., EHINGER, K., OLIVA, A., AND TORRALBA, A. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, IEEE, 3485–3492.

YEDIDIA, J. S., FREEMAN, W. T., WEISS, Y., ET AL. 2000. Generalized belief propagation. In *NIPS*, vol. 13, 689–695.