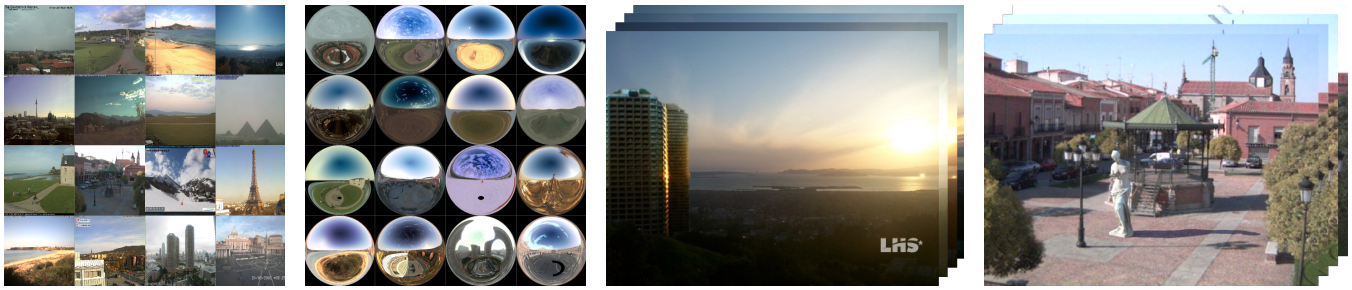
Webcam Clip Art: Appearance and Illuminant Transfer from Time-lapse Sequences

Jean-François Lalonde, Alexei A. Efros, and Srinivasa G. Narasimhan
School of Computer Science, Carnegie Mellon University



(a) Webcam database (b) Environment maps (c) 2D object insertion (building) (d) 3D object relighting (statue)

Figure 1: *We introduce a new, high-quality dataset of calibrated time-lapse sequences (a). Illumination conditions are estimated in a physically-consistent way and HDR environment maps are generated for each image (b). This database can be used as a "clip art" library for applications like inserting lighting-consistent 2D objects, such as the buildings in (c), or relighting 3D objects, such as the statue in (d).*

Abstract

Webcams placed all over the world observe and record the visual appearance of a variety of outdoor scenes over long periods of time. The recorded time-lapse image sequences cover a wide range of illumination and weather conditions – a vast untapped resource for creating visual realism. In this work, we propose to use a large repository of webcams as a "clip art" library from which users may transfer scene appearance (objects, scene backdrops, outdoor illumination) into their own time-lapse sequences or even single photographs. The goal is to combine the recent ideas from data-driven appearance transfer techniques with a general and theoretically-grounded physically-based illumination model. To accomplish this, the paper presents three main research contributions: 1) a new, high-quality outdoor webcam database that has been calibrated radiometrically and geometrically; 2) a novel approach for matching illuminations across different scenes based on the estimation of the properties of natural illuminants (sun, sky, weather and clouds), the camera geometry, and illumination-dependent scene features; 3) a new algorithm for generating physically plausible high dynamic range environment maps for each frame in a webcam sequence.

Keywords: image databases, object insertion, time-lapse video, image-based lighting, HDR, computer vision

1 Motivation

Despite the recent advances, visual realism in complex, real-world scenes remains largely elusive for traditional computer graphics. The main culprit appears to be the sheer complexity of our visual world – a realistic scene requires much more detailed geometry and lighting than could possibly be provided by hand. Of course, if one happens to have access to the physical site to be modelled (e.g. a monument or a movie set), then image-based modeling and rendering techniques can be employed *in situ*, producing excellent results (e.g. [Arnold et al. 2003]). However, often it is not possible to travel "on location" in order to capture the needed geometric and photometric data in person. Fortunately, over the last few years, the Internet has developed into a gargantuan depository of visual data (photos, videos, webcams, annotations, etc) captured by people all over the globe. A pressing research question is how this data could be useful in graphics as a way of "crowd-sourcing" visual realism.

Recently, the concept of *appearance transfer* has been proposed as a way of employing large datasets for synthesizing novel imagery. The basic idea is to: *match* various aspects of the given scene to one or more previously seen examples from the dataset; and *transfer* the appearance information associated with the examples onto the scene. Compared to more traditional techniques, the main difference is that the algorithmic burden is shifted from a synthesis task to a (hopefully easier) matching task. Some surprisingly successful applications of this simple idea have been demonstrated for several types of data including unordered photo collections [Hays and Efros 2007; Bitouk et al. 2008], and community-labelled object databases [Lalonde et al. 2007]. Closely related are efforts that perform "appearance mining" of photo collections to recover visual content like virtual walk-throughs [Snavely et al. 2008], skies [Tao et al. 2009] and even 3D models [Goesele et al. 2007].

One intriguing source of visual data that has yet to be fully explored are *Internet webcams*. A webcam is a stationary camera that continuously captures a time-lapse sequence, typically at one frame every 5-10 minutes. There are tens of thousands of such webcams all over the world, providing a rich round-the-clock visual record of some of the most interesting, as well as some of the most mundane, places on the planet. The big advantage of webcam data is that, by observing *the same scene under many different lighting and weather conditions*, it contains much more information than can be extracted from either a single image or an unstructured video. As a result, working with this kind of data allows for the use of more principled physics-based methods to comple-

ACM Reference Format
Lalonde, J., Efros, A., Narasimhan, S. 2009. Webcam Clip Art: Appearance and Illuminant Transfer from Time-lapse Sequences. *ACM Trans. Graph. 28*, 5, Article 131 (December 2009), 10 pages. DOI = 10.1145/1618452.1618477 http://doi.acm.org/10.1145/1618452.1618477.

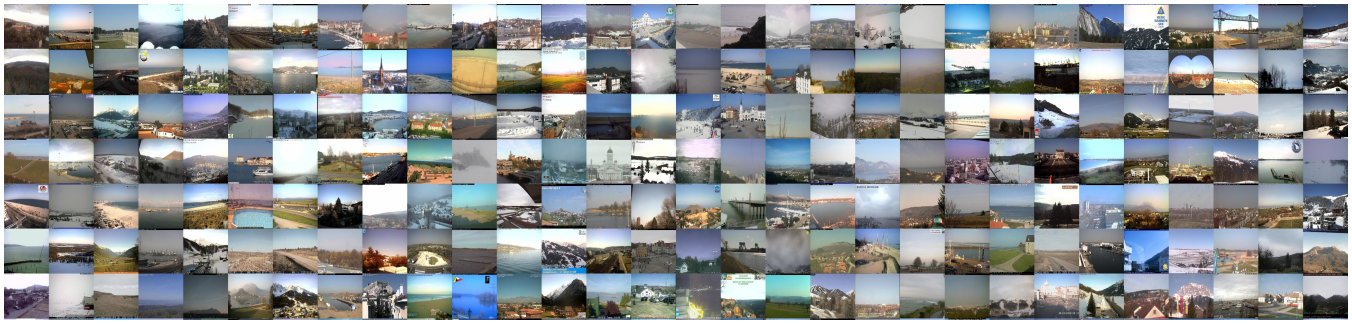


Figure 2: *180 images randomly selected from our dataset of 1.2 million images from 1350 webcams.*

ment the mostly heuristic data-driven approaches. Several research groups have explored this source of information, recovering various scene properties including scene geometry, surface reflectance, shadows, illumination, camera parameters, and even camera geolocation [Koppal and Narasimhan 2006; Sunkavalli et al. 2007; Weiss 2001; Sunkavalli et al. 2008; Kim et al. 2008; Lalonde et al. 2009; Jacobs et al. 2007b]. These efforts have prepared the ground for the next step: actually using the wealth of webcam data as a source of visual realism for computer graphics applications.

2 Overview

In this paper, we propose a combined data-driven / physics-based approach for using a library of calibrated outdoor webcams as a "Webcam clip art" – to inject varied, realistic appearance into the user's own scenes. These scenes could themselves be time-lapse sequences, as well as 3D models, 2D Light Stage objects, or, in some cases, even single photographs. Our aim is to transfer the right data from the appropriate source images while avoiding, as much as possible, synthesizing new appearance and illumination. This broadly follows the philosophy of Photo Clip Art [Lalonde et al. 2007], which proposes a complete system for outdoor object insertion from an image library by matching geometry, scale, illumination context, and local context of an object to be inserted with the background image. However, in this work, we focus on a single aspect of appearance – illumination – where time-lapse data can provide much more information than a single image, allowing for a more physics-based approach as well as a much richer set of applications. We divide these applications into two broad categories:

Illumination-consistent appearance transfer: Webcams see and record the visual life of many interesting things – buildings, monuments, mountains, beautiful skies, etc. We would like to be able to insert an object from our Webcam Clip Art library directly into a user's own time-lapsed sequence in such a way that the lighting of the object is consistent with the overall illumination of the scene *at each time instance*. For instance, using a Paris webcam, one could create a time-lapse Eiffel Tower clip art object which, when inserted into a new scene, will automatically become correctly lit.

Illuminant transfer: A webcam library could also act as a rich source of realistic, time-varying natural illumination for relighting the user's own virtual objects. By estimating the sun and sky light from the webcam, we can create a library of sky probe sequences, to be used for relighting scenes under many different illumination and weather/cloud conditions. Moreover, by hypothesizing full environment maps for a given sequence, virtual objects can be seamlessly inserted into real scenes. For instance, an architect or a sculptor wishing to see how their proposed creations will harmonize with the existing surroundings could simply install a webcam at the site (or several alternative sites), and visualize their model *in situ*, lit by a wide range of real-world illumination conditions.

These are demanding goals. To succeed will require addressing three major challenges: **1) Building a high-quality, calibrated webcam database.** Current webcam collections are either much too small [Lalonde et al. 2009] or too low-resolution [Jacobs et al. 2007a] to be suitable for graphics applications. Building a truly large dataset will require that Internet crawling and camera calibration all be done completely automatically. **2) Matching illumination conditions across different webcams.** This is hard because the webcams can depict completely different scenes in terms of geometric structure, color, and materials and weather, making a purely data-driven approach unlikely to succeed. **3) Estimating sky probes from a webcam.** The small field of view of a typical camera means that webcams show only a tiny part of the scene, from which the algorithm must recreate its global lighting environment.

Our approach to addressing these challenges combines the use of data-driven appearance transfer paradigm with a physics-based model of outdoor illumination. We will assume that all our webcams have only natural (daylight) illumination, that they are filmed by static cameras, and that their GPS locations are known.

While illumination has been analyzed extensively in laboratory settings, little has been done on outdoor illumination in unconstrained scenes observed in the real world by off-the-shelf cameras. [Lalonde et al. 2007] also model illumination within a data-driven framework, but they do it in a purely heuristic way – via color histogram of the sky region. This coarse representation loses spatial information and can only match the global "feel" of the scene (e.g. sunny vs. overcast). A similar behavior is obtained by the sky matching approach of [Tao et al. 2009]. A physics-based illumination model has been proposed by [Bitouk et al. 2008] but only for the specific domain of faces (which are assumed to be Lambertian) and does not generalize to arbitrary outdoor scenes. [Haber et al. 2009] apply inverse rendering on several images of the same object to estimate reflectance and illumination, but require a priori knowledge of object shape. Most relevant to our work, [Lalonde et al. 2009] focus on using the sky as "calibration target" to recover the parameters of the camera. While we use their technique to pre-process our webcams, we go much further. For instance, sky alone cannot reliably estimate sun visibility. We demonstrate how the scene can be used in conjunction with the sky to get a complete representation of outdoor illumination. We also show how the model can be used to extrapolate the sky appearance everywhere, even the parts of the sky hemisphere the camera does not see. Finally, we show how we can effectively match frames by by considering all the illumination parameters: sky and weather coefficients, cloud cover, sun position and visibility.

This paper is organized as follows. We first discuss how we collected our webcam library (Sec. 3), then present our data-driven model of natural illumination (Sec. 4). The remainder is devoted to deriving algorithms for our two main applications, appearance transfer (Sec. 5) and illuminant transfer (Sec. 6).

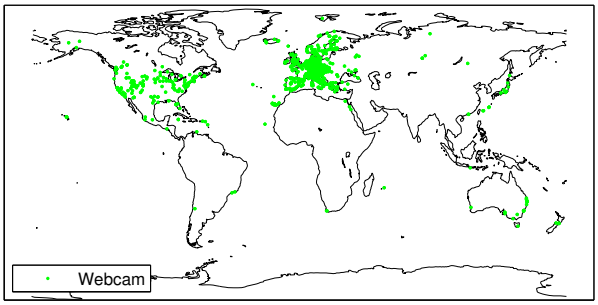


Figure 3: *World map showing the location of all 1350 webcams in the dataset. The main concentration is in Europe and North America, but there are also several in Asia and Oceania.*

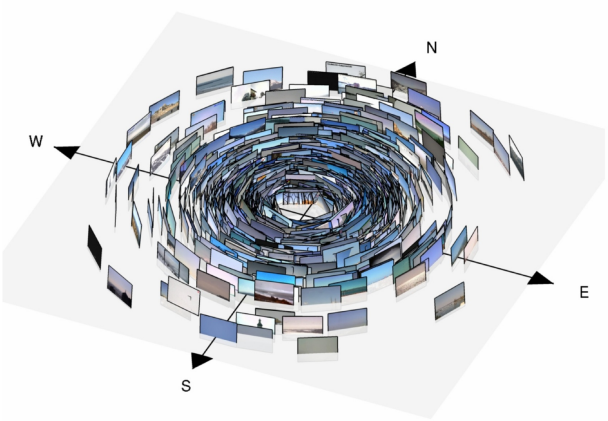


Figure 4: *Our calibrated webcam dataset. It comprises 1350 webcams downloaded from the Internet, and calibrated using the technique of [Lalonde et al. 2009]. The display depicts their fields of view and orientations.*

3 Building the Webcam Clip Art database

Creating a webcam database suitable for our needs poses several challenges. The database must be large enough to contain a wide range of appearances and illumination conditions as to be useful for graphics applications. The webcams must be of high resolution and stationary (e.g. no excessive shaking due to wind). Furthermore, all cameras within the database must be calibrated, both geometrically and radiometrically. Geometric calibration is crucial for determining the illumination direction with respect to the camera. Radiometric calibration is needed to linearize camera responses in order to fit the sky appearance model. In this section, we will briefly describe our approach for automatically collecting and calibrating our webcam database, the first of its kind in computer graphics[1].

3.1 Data collection

While the number of Internet webcams is vast, many are not suitable as sources of interesting appearance data (e.g. TrafficCams, OfficeCams, SportsCams). Additionally, many (especially older) webcams use cheap cameras and offer poor resolution and image quality. For example, we initially experimented with Archive of Many Outdoor Scenes (AMOS) [Jacobs et al. 2007a], the only existing dataset containing over 1,000 webcams, but rejected it due to low image resolution (320×240 for most sequences). Luckily, webcam quality and resolution have been steadily going up in recent times. For our task, we have been able to identify a source of high-quality webcams – the "webcams.travel" website [Opperman et al. 2009], which contains a set of live links to 11,500 geo-tagged webcams worldwide at the time of writing.

We have developed a system that crawls the entire website and automatically finds webcam sequences where 1) the resolution is sufficient, 2) the sky is visible, and 3) the camera is static. To determine if the sky is visible, we use the geometric context algorithm [Hoiem et al. 2005] and average the segmentations over several images to yield a robust sky mask. To determine if the camera is static, we first extract features (using SIFT [Lowe 2004]) from each image of a webcam. We select only those webcams with no translation between the feature locations across frames, as in [Jacobs et al. 2008].

After the crawling, we ended up with 1350 suitable webcams, for a total of over 1.2 million images (so far). The images were downloaded every 15 minutes from sunrise to sunset over at least two months, and more than one year in some cases (we continue the downloads, aiming to have at least 6 months of data from every webcam). For all webcams the location information (GPS) is provided, and every image is tagged with time-of-capture. All of our sequences have at least 640×480 pixel spatial resolution, with 10% of them with greater than 1024×768 resolution.

Fig. 2 shows a random sample of 180 sequences from our dataset. It illustrates the wide variety of scenes that are being captured: mountains, cities, landmarks, buildings, public spaces, parks, beaches, fields, airfields, streets, etc. The webcams are distributed throughout a wide range of geographic locations, as illustrated in Fig. 3. While the main concentrations are in Europe and North America, all five continents (except Antarctica) are represented.

3.2 Radiometric and geometric calibration

In addition to the raw image data, we also calibrated the cameras radiometrically and geometrically. For the radiometric calibration step, we have evaluated several existing approaches. The method of [Kuthirummal et al. 2008] cannot be used in our context since they assume a random sampling of scenes. The technique proposed by [Kim et al. 2008] also cannot be applied on all sequences since it relies on lambertian surfaces. Instead, we use the method of [Lin et al. 2004] which was designed to estimate the inverse camera response function by using color edges gathered from a single image. For robustness, we detect edges across several images, and run the optimization on all of them.

Given the GPS and time-of-capture information, we employ the method of [Lalonde et al. 2009] to geometrically calibrate the webcams; let us summarize it here for completeness. The method works by fitting a physically-based model of sky appearance [Perez et al. 1993] to clear sky images. This model expresses the *relative* luminance l_p of a sky element as a function of its zenith angle θ_p and angle γ_p with respect to the sun:

$$l_p = f(\theta_p, \gamma_p) = [1 + a \exp(b/\cos\theta_p)] \times \left[1 + c\exp(d\gamma_p) + e\cos^2\gamma_p\right] \quad . \tag{1}$$

Here, the weather coefficients a, b, c, d, e quantify the atmospheric conditions, which can range from clear to overcast. The *absolute* luminance is obtained by dividing by the predicted zenith luminance:

$$L_p = L_z \frac{f(\theta_p, \gamma_p)}{f(0, \theta_s)} \quad , \tag{2}$$

where L_z is the actual zenith luminance, and θ_s is the sun zenith angle.

[1] See http://graphics.cs.cmu.edu/projects/webcamclipart

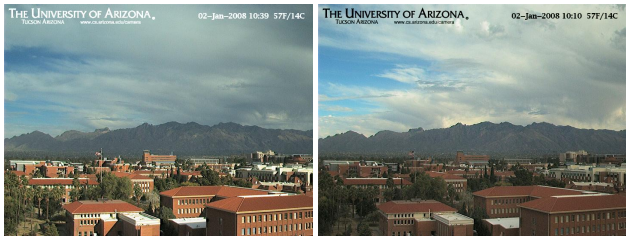


Figure 5: *An example of similar sky appearance, but very different scene illumination. We rely on scene-based features to determine whether the sun is visible or not.*

When the sky is clear, the weather coefficients take on known numerical values, and thus (2) depends only on angles. The authors then show how (2) can be reparameterized as a function of three camera parameters: the focal length f_c and its zenith and azimuth angles θ_c and ϕ_c respectively. The exact expressions can be found in [Lalonde et al. 2009].

By fitting this camera-dependent function to clear sky images and optimizing over the camera parameters, they can effectively use the sky as a calibration target. They demonstrate good accuracy of the method on over 40 real, low-quality webcam sequences. As a preprocessing step to this work, we estimated the camera orientation (azimuth ϕ_c and zenith θ_c angles) and focal length f_c for each webcam in our library.

4 A data-driven model of natural illumination

The appearance of an outdoor scene depends on a variety of factors such as sun position, camera orientation, sky color, cloud cover and weather conditions. Our approach relies on extracting illumination information from both the sky and the scene present in the images of a webcam.

4.1 Analysis of sky appearance

The visible portion of the sky is a rich source of information about outdoor illumination, especially when observed over time. For instance, we have seen in Sec. 3.2 how to recover the camera parameters from the sky only. Moreover, since GPS coordinates and time-of-capture are known, it is easy to compute the relative position of the sun with respect to the camera, even if the sun is never directly observed.

The sun is, of course, the dominant illuminant, but the sky itself acts as an area light source of spatially and temporally varying intensity and can strongly affect the scene. For instance, it is responsible for the global "feel" of the scene (sunny vs cloudy, red sky at sunsets, etc.). How can we capture its color variation when only such a small portion is visible? To capture the sky appearance, we use the same sky model (2) as suggested by [Lalonde et al. 2009]. Instead of optimizing the model over camera parameters as in Sec. 3.2, they optimize over the turbidity t (encoding the weather coefficients in a single number [Preetham et al. 1999]). While this is only an approximation, it was found to work very well in practice. To add robustness, they propose to use a data-driven prior model of the clear sky, and force the optimization to remain close to it by adding a regularization term. This enables the recovery of sky *layers*: one for the sky, described by turbidity, and the other one for the clouds, which are outliers for the sky model. For each frame of our 1350 webcams, we estimate the turbidity t of the atmosphere. Finally, for each image, we also estimate the cloud layer.

4.2 Estimating sun visibility

While the approach of [Lalonde et al. 2009] gives robust estimates of the sky and camera parameters, it is not enough to describe the illumination conditions of the *scene*. This is because the visible portion of the sky is typically a small fraction of the entire sky hemisphere, and much of it remains unobserved within the field of view. For instance, Fig. 5 shows two similarly-looking cloudy skies with scenes illuminated differently: in the first, the sun shines from the right of the camera on the scene, but not in the second. How can we then determine whether the sun is occluded by clouds or other objects when the full sky hemisphere is not observed? For this, we must rely on cues provided by the visible scene itself.

Consider the mean image of a large webcam sequence: it shows no preferred illumination direction and resembles a scene captured under overcast skies (where sun is hidden behind clouds). Only small deviations from this mean image are likely when the sun is not visible to the scene. But larger deviations (bright or dark) from the mean are more likely when the sun is visible. Based on this observation, we compute scene features that can be used as a measure of sun visibility. In order to avoid the scene-dependent hue, the features are computed in the saturation-value space. We first compute the ratios of saturation and value of the current image w.r.t the mean image, and then compute a 2-D joint histogram of these ratio images. For an image where the sun is not visible, the joint histogram will have a dominant peak near (1,1). On the other hand, when the sun is visible, the shadow regions result in multiple peaks or a wider spread in the joint histogram.

We validated the effectiveness of our representation by evaluating it on a set of 400 randomly chosen images that cover a wide range of scene and sky appearances. We manually labelled each image as to whether or not the scene appears to be sunny. We compare the performance of our features against three others, also computed over the saturation-value space — (a) 2-D joint histogram of saturation and value of a single image; (b) Histogram of gradients of an image as a measure of local contrast, computed on the strong edges only (determined by canny edge detector with high threshold of 0.1); (c) first four moments of each of the marginal histograms of the previous features. A k-NN classifier is used to test the performance of the features, where the optimal k is determined by 10-fold cross-validation. The k nearest neighbors are found by using the χ^2 distance for the histogram-based features, and L2 for the moments. The visibility classification accuracy for each of these features are shown in Fig. 6. Our features are computed relative to the mean image and outperform the others.

Together with the sky and camera parameters, the visibility of the sun is crucial for matching illumination across scenes for appearance transfer (Sec. 5). It is also required to estimate the sun brightness for sky probe synthesis (Sec. 6). For that, we will need the *partial* sun visibility, i.e. how much is the sun being occluded. We estimate the sun visibility coefficient s_{vis} by taking the ratio of the number of nearest neighbors with fully visible suns divided by k. Note that we currently cannot evaluate our partial visibility estimation method quantitatively since obtaining ground truth would require knowing the thickness of occluding clouds even if they are not visible. Qualitatively however, results are satisfying: the shadows generated under a partially-occluded sun match those of the scene (see Fig. 12(a) for example).

5 Appearance transfer across webcams

The world's webcams record the visual appearance of cities, mountains, beaches, forests and skies, under a variety of illumination and weather conditions. But while the space of all possible scenes may

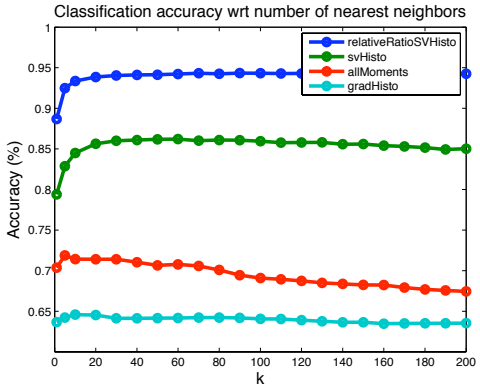


Figure 6: *Classification accuracy (in %) for each feature tested on our training set, using a k nearest neighbor algorithm. Each feature is evaluated by varying k and performing 10-fold cross validation on the training set.*

well be infinite, the space of illumination conditions they experience might not be that large. Indeed, given a large-enough database of scenes it is extremely likely that many will share similar orientation with respect to the sun and, given long-enough recording time, similar skies and weather. And if a similar scene can be found, then appearance can simply be transferred from it to the target scene, without having to explicitly model the complex physics and material properties. Thus, the collection of webcams can be treated as a "clip art" from which users can insert objects into their own time-lapse photographs in an illumination-consistent way. As with any data-driven approach, just growing the number of webcams in the clip art library will further improve performance by increasing the number of illumination matches for transfer. In this section, we describe our techniques for matching illumination and transferring objects from one webcam to another.

5.1 Matching illumination across scenes

Matching illuminations of completely different scenes is a challenging problem. One scene may be a natural landscape and the other may consist of skyscrapers. In this case, simple image-based heuristics will not suffice because they do not take into account either the viewing and illumination geometry, the type of weather, or the sun visibility. However, this is exactly the type of information we can estimate from webcam sequences, as described in Sec. 4.

Consider two images captured by different webcams. In order to test whether the appearance of an object in one can be transferred to another, we must match (a) the camera azimuth angle relative to the sun $\Delta\phi = (\phi_c - \phi_s)$, which ensures, for example, whether the sun is either to the left or right or behind the cameras; (b) the camera zenith angles θ_c; (c) the sun zenith angle θ_s which effects sky color especially during sunrise and sunset; (d) the visibility of the sun; (e) turbidity t in the atmosphere; and (f) the cloud coverage denoted by the norm $||C||$ of the cloud layer. Since we do not expect different webcams to exactly match all of these five quantities, we define an aggregate matching function M between the skies of source and destination images i and j, as the weighted linear combination:

$$M = w_1 \angle(\Delta\phi^i, \Delta\phi^j) + w_2 \angle(\theta_c^i, \theta_c^j) + w_3 w_z \angle(\theta_s^i, \theta_s^j) + w_4 \chi^2(V^i, V^j) + w_5 D(C^i, C^j) + w_6 D(t^i, t^j)\,, \quad (3)$$

where, $\angle(\cdot,\cdot)$ is the angular difference, $D(\cdot,\cdot)$ the L2 distance, and $\chi^2(\cdot,\cdot)$ the chi-square distance between two histograms. We denote by V the sun visibility features from Sec. 4.2.

(a) Input image (b) [Lalonde et al. 2007] (c) Ours

Figure 7: *Sky matching comparison against [Lalonde et al. 2007]. The input image from the Vatican sequence (a) is matched against frames from the Visby sequence using Photo Clip Art (b) and our technique (c). While Photo Clip Art can capture the overall "feel" of the scene (sunny, in this case), note that the shadow direction is completely wrong.*

The weights $w_1 = 2$, $w_2 = 1.5$, $w_3 = 1.5$, $w_4 = 1.5$, $w_5 = 1$, and $w_6 = 1$ are used in all the examples shown in the paper. Note that errors in sun and camera angles or sun visibility are more easily perceived that turbidity or cloud cover, hence their larger relative weights. Furthermore, matching the colors of the sky during sunrise or sunset is critical for perception. We set an additional weight $w_z = \sin(\theta_s^i)$ that increases the importance of the term when the sun is close to the horizon ($\theta_s^i = \pi/2$). Each of the terms in the above functions are equalized by normalizing them to a range of [0-1]. A nearest neighbor algorithm is used to match the function M of one webcam against the library of source webcams.

Fig. 7 compares our new illumination matching function M with the one proposed in [Lalonde et al. 2007]. While their approach is good at representing the general scene mood (e.g. sunny vs. cloudy), it does not accurately capture the sun position, so it cannot be used to transfer the appearance of objects and scenes across webcams in a physically-consistent way.

5.2 2-D appearance transfer across webcams

Once the illuminations of the source and destination webcams are matched, the object of interest can be matted from the source webcam and composited into the destination. Fig. 8 shows three different appearance transfer results: 1) skyscrapers from Tokyo are placed into an image sequence in Berkeley, CA; 2) the Eiffel Tower in Paris is composited into the Berlin skyline; and 3) the sky (it's just another object!) can likewise be transferred, here from Berkeley to Portugal. To compensate for color differences that might exist between cameras, we post-process the object by re-coloring according to the scene colors, following [Reinhard et al. 2001]. When the sky is transferred, pixels are automatically warped according to the field of view of the cameras. Notice the beautiful sunset effect where the reflection of the sun, the colors of the buildings and the sky appearance are all consistent. The consistency between the illumination conditions of the transferred object and those of the destination scene is what makes the results appear so realistic.

Illumination can be matched for each frame of the source / destination webcam. However, in order to ensure smooth transitions between frames adjacent in time, we add a constraint which penalizes large changes in appearance in the resulting video. We first compute the mean histogram of scene intensity differences over all pairs of adjacent frames in the sequence containing the source object. We then penalize images whose histogram of object intensity differences with respect to the previous frame differs from the mean histogram (using the χ^2 distance measure). We ensure global consistency using dynamic programming as in [Schöld et al. 2000].

The success of appearance transfer depends on the quality of matches recovered from the database. Rare illumination conditions such as storms and dense fog for example, are harder to find and

may result in erroneous relighting results. Since not every webcam can be matched to every other webcam (e.g. sun positions at the earth's equator and pole do not overlap), we assume that the webcams are distributed across the entire globe. As with other data-driven approaches, many current limitations will be addressed by simply adding more data. In addition, while this type of compositing maintains consistency in the appearance of the object, shadows cast by the object onto others (or vice versa) are hard to simulate without knowing 3D geometry of the scene. While shadow detection and transfer solutions have been proposed [Weiss 2001; Lalonde et al. 2007], they may not work in arbitrarily complex scenes with occlusions, moving objects, etc. While this topic is part of our future investigations, in the following section we present a different way to accurately handle object shadows – by using 3D virtual object models.

6 Illuminant transfer

Webcam sequences can be used to transfer pixel data across scenes, as we demonstrated earlier, but we would also like to move beyond pixels, and capture as much of the illumination information as we can. This will allow us to insert synthetic 3D objects seamlessly into a webcam sequence, and also to use the natural light from a webcam to relight a novel 3D scene. In this case, knowledge of geometry will result in physically-consistent shadows.

The way that the relighting problem is typically addressed in graphics is by capturing natural light from the real scene and using it as an *environment map* to light virtual objects [Debevec 1998]. An environment map is a sample of the plenoptic function at a single point in space capturing the full sphere of light rays incident at that point. It is typically captured by either taking a high dynamic range (HDR) panoramic photograph from the point of view of the object, or by placing a mirrored ball at the desired location and photographing it. Such an environment map can then be used as an extended light source for rendering synthetic objects.

Given only an image or a webcam sequence however, it is generally impossible to recover the true environment map of the scene, since an image will only contain a small visible portion of the full map (and from the wrong viewpoint besides). However, there is psychophysical evidence suggesting that an approximation to the environment map which is consistent with the target image is typically enough to create believable lighting effects [Dror et al. 2004]. Consequently, a simple technique for environment map estimation from a single image has been proposed [Khan et al. 2006] that work reasonably well for its intended setting. However, we have found it inadequate for our task mainly due to its low dynamic range and inability to estimate even approximate illumination direction. Both problems stem from the fact that the sun and the sky – the main illuminants in an outdoor environment – are not being explicitly modelled. Coincidentally, this is exactly the information that we can reliably recover from webcam sequences (Sec. 4). Here we propose to use our model of natural illumination to estimate high dynamic range environment maps at each frame. Since we are dealing with outdoor images, we can divide the environment map into two parts: the *sky probe* and the *scene probe*. We now detail the process of building a realistic approximation to the real environment map for these two parts.

6.1 Creating a sky probe

Our sky probe is composed of the sky layer, the sun layer, and the cloud layer. The sky layer is generated in a straightforward way from the sky parameters that were computed in Sec. 4.1. The estimated turbidity and camera parameters allow us to use (2) and extrapolate the sky appearance on the entire hemisphere, even though only a small portion of it was originally visible to the camera.

Color channel	α	β
R	1.4243×10^{10}	0.2187
G	1.4463×10^{10}	0.2399
B	1.3×10^{10}	0.2889

Table 1: *Recovered parameters after fitting our parametric model to the sun data from the HDR sky database of [Stumpfel et al. 2004].*

The camera calibration procedure also yields the relative position of the sun with respect to the camera. For the sun layer, we can use this to position a synthetic sun at the desired location. But how should it look, and how bright should it be? We rely on the HDR sky dataset of Stumpfel et al. [Stumpfel et al. 2004] to estimate sun appearance and brightness. In all frames of the dataset where the sun is not occluded by clouds, we can automatically detect the location of its center by fitting a 2-D Gaussian in log-intensity space. We compute the sun appearance by rotating the sky probe so that all the sun locations are at the origin, and averaging them.

We also compute the sun brightness in each color channel across the full database as a function of its height and fit a parametric model of the form $\alpha \exp(-\beta m(\theta_s))$, where $m(\theta_s)$ is relative optical path length through Earth's atmosphere [Kasten and Young 1989]:

$$m(\theta_s) = \frac{1}{\cos(\theta_s) + 0.50572(96.07995 - \frac{180}{\pi}\theta_s)^{-1.6364}} \quad . \quad (4)$$

Here, α is an arbitrary scale factor and β is a scattering coefficient. Table 1 shows the recovered values for parameters (α, β) for the RGB color channels, and Fig. 10 illustrates that the model provides a good fit to the sun data.

The final sun appearance placed in the sun layer of the sky probe is generated by taking the mean sun image and rescaling it by $s(\theta_s)$ which will take care of both sun height and sun occlusion by clouds:

$$s(\theta_s) = s_{max} s_{vis} \frac{\alpha \exp(-\beta m(\theta_s))}{\alpha \exp(-\beta)} \quad , \quad (5)$$

since $m(0) = 1$, and where s_{vis} is the partial sun visibility coefficient (defined in Sec. 4.2) and s_{max} is the maximum sun brightness value, reached when the sun is at zenith $\theta_s = 0$ ($s_{max} = 1 \times 10^5$ in all our experiments).

Finally, we must define the cloud layer. Its role is mostly decorative (since we have already taken care of sun occlusion by clouds), useful mainly for relighting mirror-like objects which can reflect the sky. We make an assumption that the cloud cover of the full sky is statistically similar to the cloud cover of the visible sky. Therefore, we can use texture synthesis to create plausible cloud cover for the missing portion of the sky. We first apply the cloud segmentation algorithm of [Lalonde et al. 2009] to extract the visible cloud layer. Next we use Image Quilting [Efros and Freeman 2001] to synthesize more of the cloud layer, but we run it in the gradient domain to preserve the color of the clear sky layer underneath. Quilting proceeds along the longitude first, and then the "north cap" of the map is filled in by planar projection at the pole. The final sky probe is the sum of the clear sky, sun, and the cloud layers. It captures the approximate illuminant of the scene in high dynamic range and can be used directly in rendering programs.

6.2 Estimating the environment map

To insert a synthetic 3D object into an image from a given webcam, we need the full spherical environment map. We have already estimated the main illuminant in the scene – the sky probe, and the

(a) Object transfer (building)

(b) Object transfer (Eiffel tower)

(c) Sky transfer

Figure 8: *Appearance transfer between webcams. The first column shows example frames from the webcams corresponding to the object (top) and destination (bottom). The other three columns show (a) buildings transferred from Tokyo to Berkeley and "re-lit" by finding an image in the Tokyo sequence with illumination that matches that of Berkeley; (b) the Eiffel Tower transferred in Berlin, notice how the shading effects are consistent across the scene; (c) the sky transferred from Berkeley to Portugal, in which case the scene can be automatically "re-lit" in an illumination-consistent way, observe how the reflections on the water are consistent with the sun position.*

Figure 9: *Rain or shine, the knight stands guard at the castle. Our technique can be used to render objects captured in a Light Stage [Debevec et al. 2002].*

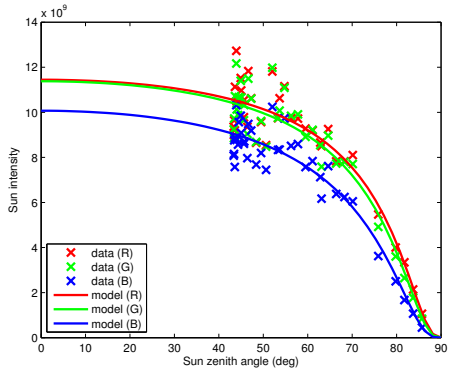


Figure 10: *Sun intensity as a function of sun zenith angle. The data points are obtained from the HDR sky database [Stumpfel et al. 2004], and the solid lines indicate the prediction by our low-dimensional parametric model (Eqn. 5).*

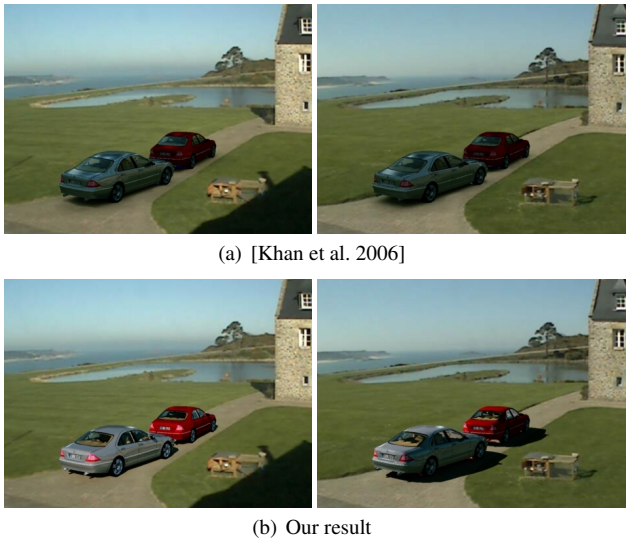
rest will be filled-in from the input image directly. First, we use the approach of [Khan et al. 2006] to map the pixels below the horizon line in the image onto the bottom hemisphere. This way, we create a map with the sky in the top hemisphere and the mirrored spherical projection of the ground in the bottom hemisphere. However, we have not yet accounted for objects that protrude above the horizon line and occlude part of the sky from view. For this we use another simple approximation – find all non-sky objects above the horizon (using the sky mask defined previously) and project them onto a cylinder around the equator of the environment map.

6.3 Relighting virtual objects

We can now insert virtual objects into a real scene under various illuminations. First, we show how a light-field object captured in a Light Stage apparatus [Debevec et al. 2002] can easily be inserted with the correct lighting. The object is synthesized by summing images of the object taken under 256 different light directions, and weighting them according to the light intensities specified in our estimated environment maps. Fig. 9 shows the result of inserting a light-field knight into a castle scene. To insert 3-D objects, we follow the image-based lighting method of [Debevec 1998]. This assumes knowledge of the geometry of the scene, which can be acquired in several ways, using photometric stereo [Sunkavalli et al. 2007], or manual intervention [Debevec et al. 1996; Horry et al. 1997] for example. We approximate the scene with a ground plane surrounding the object. Since geometry is known, complex shadow effects can be generated by most off-the-shelf rendering packages.

We now demonstrate practical examples of our approach. Fig. 12 shows a scenario where architects wish to know how their newest design will look in three different settings: on a beachfront property, on a farm, or in the Swiss mountains. Our large database provides a wide variety of such environments, which can be used to preview how an object would appear at different times of day and under different weather conditions. Alternatively, one can also visualize how an object will harmonize with existing surroundings by installing a webcam at that site. For example, Fig. 13 illustrates what would happen if the Madrid city planners chose to insert a statue of Venus de Milo in a public square. The statue can be visualized *in situ*, lit by real-world illumination conditions specific to that location.

We compare our approach to that of [Khan et al. 2006] in Fig. 11. The dynamic range and sun orientation captured by our method make the inserted object appear much more realistic.



(a) [Khan et al. 2006]

(b) Our result

Figure 11: *Comparison between objects relit by environment maps obtained with (a) the method of [Khan et al. 2006], and (b) ours. The dynamic range and sun orientation captured by our method make the inserted object appear much more realistic.*

6.4 Relighting in a single image

Finally, we demonstrate an application of illuminant transfer for relighting a scene from a *single image* (provided it was taken on overcast day). We start by modeling the geometry of the scene. For simplicity, we used a slightly modified "Tour into the Picture" approach [Horry et al. 1997] in which there is no ceiling, but any other single view modeling approach may be used. The geometry allows us to compute a sky visibility map for each pixel. Since there are no shadows present in the overcast image, the albedo of every scene point can be estimated using the known illumination and scene geometry, based on the Lambertian reflectance model. We transfer a new sky probe as described above into the scene. Using a simple ray tracing algorithm we are then able to relight the scene with the new sky probe, as shown in Fig. 14.

7 Conclusion

We have shown how to exploit the abundance of Internet webcam data that is available to us for relighting applications. Achieving visual realism by synthesizing appearance is a hard problem. Instead, for the first time, we are able to transfer appearances (sky, objects) from one scene into another and achieve a wide range of relighting effects using webcam data. The large repository of webcams serves as a new form of "clip art" from which objects can be inserted into a user's time-lapse sequence or even a single photograph in an illumination-consistent way. We have shown several applications of relighting and appearance transfer between two webcams, and webcams and single images. We believe this work presents a first step in a promising new direction – using the webcams of the world as a viable source of computer graphics content.

Acknowledgements

This work has been partially supported by NSF grants CCF-0541230, IIS-0546547, IIS-0643628 and ONR grant N00014-08-1-0330. A. Efros is grateful to the WILLOW team at ENS Paris for their hospitality. We thank Danielle Millett for her help with the single image relighting example.

(a) (b) (c)

Figure 12: *An architect wants to show what his architectural model will look like in different settings: (a) on the beach, (b) on a farm, and (c) in the mountains; and under different illumination conditions. It is easy to do so using our rich webcam dataset and our automatic environment map extraction algorithm. The corresponding environment maps are shown as insets.*

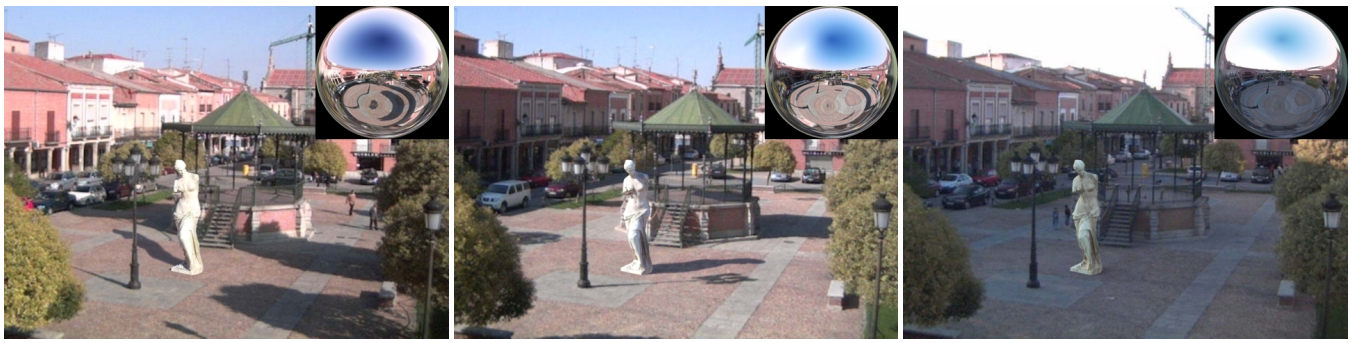


Figure 13: *The Madrid city planners are considering adding a* Venus de Milo *statue to this square. Using our approach, it is easy to visualize what it will look like in that environment. Notice how the cast shadows on the ground follow the shadows of the other objects in the scene, and how the sun visibility is estimated properly. Shadows can be inserted behind objects by manual layering (as in the left-most image). The corresponding environment maps are shown as insets.*

Figure 14: *Single image relighting. The overcast input image (left) is relit by a clear Berkeley sky and shown here in the morning, noon, and evening.*

References

ARNOLD, D., CHALMERS, A., NICCOLUCCI, F., STUMPFEL, J., TCHOU, C., YUN, N., HAWKINS, T., JONES, A., EMERSON, B., AND DEBEVEC, P. 2003. Digital reunification of the parthenon and its sculptures. In *4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*.

BITOUK, D., KUMAR, N., DHILLON, S., BELHUMEUR, P., AND NAYAR, S. K. 2008. Face swapping: automatically replacing faces in photographs. *ACM Transactions on Graphics (SIGGRAPH) 27*, 3, 39.

DEBEVEC, P., TAYLOR, C. J., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *ACM SIGGRAPH*.

DEBEVEC, P., WENGER, A., TCHOU, C., GARDNER, A., WAESE, J., AND HAWKINS, T. 2002. A lighting reproduction approach to live-action compositing. *ACM Transactions on Graphics (SIGGRAPH)*.

DEBEVEC, P. 1998. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH*.

DROR, R. O., WILLSKY, A. S., AND ADELSON, E. H. 2004. Statistical characterization of real-world illumination. *Journal of Vision 4*.

EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. In *ACM SIGGRAPH*.

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

HABER, T., FUCHS, C., BEKAERT, P., SEIDEL, H.-P., GOESELE, M., AND LENSCH, H. P. A. 2009. Relighting objects from image collections. In *Conference on Computer Vision and Pattern Recognition*.

HAYS, J., AND EFROS, A. A. 2007. Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH) 26*, 3, 4.

HOIEM, D., EFROS, A. A., AND HEBERT, M. 2005. Geometric context from a single image. In *International Conference on Computer Vision*.

HORRY, Y., ANJYO, K.-I., AND ARAI, K. 1997. Tour into the picture: using a spidery mesh interface to make animation from a single image. In *ACM SIGGRAPH*.

JACOBS, N., ROMAN, N., AND PLESS, R. 2007. Consistent temporal variations in many outdoor scenes. In *Conference on Computer Vision and Pattern Recognition*.

JACOBS, N., SATKIN, S., ROMAN, N., SPEYER, R., AND PLESS, R. 2007. Geolocating static cameras. In *International Conference on Computer Vision*.

JACOBS, N., ROMAN, N., AND PLESS, R. 2008. Toward fully automatic geo-location and geo-orientation of static outdoor cameras. In *Workshop on applications of computer vision*.

KASTEN, F., AND YOUNG, A. T. 1989. Revised optical air mass tables and approximation formula. *Applied optics 28*.

KHAN, E. A., REINHARD, E., FLEMING, R., AND BÜELTHOFF, H. 2006. Image-based material editing. *ACM Transactions on Graphics (SIGGRAPH)*.

KIM, S. J., FRAHM, J.-M., AND POLLEYFEYS, M. 2008. Radiometric calibration with illumination change for outdoor scene analysis. In *Conference on Computer Vision and Pattern Recognition*.

KOPPAL, S. J., AND NARASIMHAN, S. G. 2006. Clustering appearance for scene analysis. In *International Conference on Computer Vision*.

KUTHIRUMMAL, S., AGARWALA, A., GLODMAN, D. B., AND NAYAR, S. K. 2008. Priors for large photo collections and what they reveal about cameras. In *European Conference on Computer Vision*.

LALONDE, J.-F., HOIEM, D., EFROS, A. A., ROTHER, C., WINN, J., AND CRIMINISI, A. 2007. Photo clip art. *ACM Transactions on Graphics (SIGGRAPH) 26*, 3, 3.

LALONDE, J.-F., NARASIMHAN, S. G., AND EFROS, A. A. 2009. What do the sun and the sky tell us about the camera? *International Journal on Computer Vision (to appear)*.

LIN, S., GU, J., YAMAZAKI, S., AND SHUM, H.-Y. 2004. Radiometric calibration from a single image. In *Conference on Computer Vision and Pattern Recognition*.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2.

OPPERMAN, I., FERNDRIGER, S., AND EUGSTER, J., 2009. Touristic webcams worldwide. Accessed: 09/03/09, `http://www.webcams.travel`.

PEREZ, R., SEALS, R., AND MICHALSKY, J. 1993. All-weather model for sky luminance distribution – preliminary configuration and validation. *Solar Energy 50*, 3 (March), 235–245.

PREETHAM, A. J., SHIRLEY, P., AND SMITS, B. 1999. A practical analytic model for daylight. In *ACM SIGGRAPH*.

REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Computer Graphics and Applications, special issue on Applied Perception 21*, 5 (September - October), 34–41.

SCHÖLD, A., SZELISKI, R., SALESIN, D., AND ESSA, I. 2000. Video textures. In *ACM SIGGRAPH*.

SNAVELY, N., GARG, R., SEITZ, S. M., AND SZELISKI, R. 2008. Finding paths through the world's photos. *ACM Transactions on Graphics (SIGGRAPH) 27*, 3, 15.

STUMPFEL, J., JONES, A., WENGER, A., TCHOU, C., HAWKINS, T., AND DEBEVEC, P. 2004. Direct HDR capture of the sun and sky. In *AFRIGRAPH*.

SUNKAVALLI, K., MATUSIK, W., PFISTER, H., AND RUSINKIEWICZ, S. 2007. Factored time-lapse video. *ACM Transactions on Graphics (SIGGRAPH) 26*, 3, 101.

SUNKAVALLI, K., ROMEIRO, F., MATUSIK, W., ZICKLER, T., AND PFISTER, H. 2008. What do color changes reveal about an outdoor scene? In *Conference on Computer Vision and Pattern Recognition*.

TAO, L., YUAN, L., AND SUN, J. 2009. SkyFinder: attribute-based sky image search. *ACM Transactions on Graphics (SIGGRAPH) 28*, 3, 68.

WEISS, Y. 2001. Deriving intrinsic images from image sequences. In *International Conference on Computer Vision*.