# Deep Photo: Model-Based Photograph Enhancement and Viewing

Johannes Kopf, University of Konstanz
Boris Neubert, University of Konstanz
Billy Chen, Microsoft
Michael Cohen, Microsoft Research
Daniel Cohen-Or, Tel Aviv University
Oliver Deussen, University of Konstanz
Matt Uyttendaele, Microsoft Research
Dani Lischinski, The Hebrew University


Original Dehazed Relighted Annotated

**Figure 1:** *Some of the applications of the Deep Photo system.*

## Abstract

In this paper, we introduce a novel system for browsing, enhancing, and manipulating casual outdoor photographs by combining them with already existing georeferenced digital terrain and urban models. A simple interactive registration process is used to align a photograph with such a model. Once the photograph and the model have been registered, an abundance of information, such as depth, texture, and GIS data, becomes immediately available to our system. This information, in turn, enables a variety of operations, ranging from dehazing and relighting the photograph, to novel view synthesis, and overlaying with geographic information. We describe the implementation of a number of these applications and discuss possible extensions. Our results show that augmenting photographs with already available 3D models of the world supports a wide variety of new ways for us to experience and interact with our everyday snapshots.

**Keywords:** image-based modeling, image-based rendering, image completion, dehazing, relighting, photo browsing

**ACM Reference Format**
Kopf, J., Neubert, B., Chen, B., Cohen, M., Cohen-Or, D., Deussen, O., Uyttendaele, M., Lischinski, D. 2008. Deep Photo: Model-Based Photograph Enhancement and Viewing. *ACM Trans. Graph. 27*, 5, Article 116 (December 2008), 10 pages. DOI = 10.1145/1409060.1409069 http://doi.acm.org/10.1145/1409060.1409069.



## 1 Introduction

Despite the increasing ubiquity of digital photography, the metaphors we use to browse and interact with our photographs have not changed much. With few exceptions, we still treat them as 2D entities, whether they are displayed on a computer monitor or printed as a hard copy. It is well understood that augmenting a photograph with depth can open the way for a variety of new exciting manipulations. However, inferring the depth information from a single image that was captured with an ordinary camera is still a long-standing unsolved problem in computer vision. Luckily, we are witnessing a great increase in the number and the accuracy of geometric models of the world, including terrain and buildings. By registering photographs to these models, depth becomes available at each pixel. The Deep Photo system described in this paper, consists of a number of applications afforded by these newfound depth values, as well as the many other types of information that are typically associated with such models.

Deep Photo is motivated by several recent trends now reaching critical mass. The first trend is that of geo-tagged photos. Many photo sharing web sites now enable users to manually add location information to photos. Some digital cameras, such as the RICOH Caplio 500SE and the Nokia N95, feature a built-in GPS, allowing automatic location tagging. Also, a number of manufacturers offer small GPS units that allow photos to be easily geo-tagged by software that synchronizes the GPS log with the photos. In addition, location tags can be enhanced by digital compasses that are able to measure the orientation (tilt and heading) of the camera. It is expected that, in the future, more cameras will have such functionality, and that most photographs will be geo-tagged.

The second trend is the widespread availability of accurate digital terrain models, as well as detailed urban models. Thanks to commercial projects, such as Google Earth and Microsoft's Virtual Earth, both the quantity and the quality of such models is rapidly increasing. In the public domain, NASA provides detailed satellite imagery (e.g., Landsat [NASA 2008a]) and elevation models (e.g., Shuttle Radar Topography Mission [NASA 2008b]). Also, a number of cities around the world are creating detailed 3D models of their cityscape (e.g., Berlin 3D).

The combination of geo-tagging and the availability of fairly accurate 3D models allows many photographs to be precisely *geo-registered*. We envision that in the near future automatic geo-registration will be available as an online service. Thus, although we briefly describe the simple interactive geo-registration technique that we currently employ, the emphasis of this paper is on the applications that it enables, including:

- dehazing (or adding haze to) images,
- approximating changes in lighting,
- novel view synthesis,
- expanding the field of view,
- adding new objects into the image,
- integration of GIS data into the photo browser.

Our goal in this work has been to enable these applications for single outdoor images, taken in a casual manner without requiring any special equipment or any particular setup. Thus, our system is applicable to a large body of existing outdoor photographs, so long as we know the rough location where each photograph was taken. We chose New York City and Yosemite National Park as two of the many locations around the world, for which detailed textured models are already available[1]. We demonstrate our approach by combining a number of photographs (obtained from flickr™) with these models.

It should be noted that while the models that we use are fairly detailed, they are still a far cry from the degree of accuracy and the level of detail one would need in order to use these models directly to render photographic images. Thus, one of our challenges in this work has been to understand how to best leverage the 3D information afforded by the use of these models, while at the same time preserving the photographic qualities of the original image.

In addition to exploring the applications listed above, this paper also makes a number of specific technical contributions. The two main ones are a new data-driven stable dehazing procedure, and a new model-guided layered depth image completion technique for novel view synthesis.

Before continuing, we should note some of the limitations of Deep Photo in its current form. The examples we show are of outdoor scenes. We count on the available models to describe the distant static geometry of the scene, but we cannot expect to have access to the geometry of nearby (and possibly dynamic) foreground objects, such as people, cars, trees, etc. In our current implementation such foreground objects are matted out before combining the rest of the photograph with a model, and may be composited back onto the photograph at a later stage. So, for some images, the user must spend some time on interactive matting, and the fidelity of some of our manipulations in the foreground may be reduced. That said, we expect the kinds of applications we demonstrate will scale to include any improvements in automatic computer vision algorithms and depth acquisition technologies.

[1]For Yosemite, we use elevation data from the Shuttle Radar Topography Mission [NASA 2008b] with Landsat imagery [NASA 2008a]. Such data is available for the entire Earth. Models similar to that of NYC are currently available for dozens of cities.

## 2 Related Work

Our system touches upon quite a few distinct topics in computer vision and computer graphics; thus, a comprehensive review of all related work is not feasible due to space constraints. Below, we attempt to provide some representative references, and discuss in detail only the ones most closely related to our goals and techniques.

**Image-based modeling.** In recent years, much work has been done on image-based modeling techniques, which create high quality 3D models from photographs. One example is the pioneering Façade system [Debevec et al. 1996], designed for interactive modeling of buildings from collections of photographs. Other systems use panoramic mosaics [Shum et al. 1998], combine images with range data [Stamos and Allen 2000], or merge ground and aerial views [Früh and Zakhor 2003], to name a few.

Any of these approaches may be used to create the kinds of textured 3D models that we use in our system; however, in this work we are not concerned with the creation of such models, but rather with the ways in which their combination with a single photograph may be useful for the casual digital photographer. One might say that rather than attempting to automatically or manually reconstruct the model from a single photo, we exploit the availability of digital terrain and urban models, effectively replacing the difficult 3D reconstruction/modeling process by a much simpler registration process.

Recent research has shown that various challenging tasks, such as image completion and insertion of objects into photographs [Hays and Efros 2007; Lalonde et al. 2007] can greatly benefit from the availability of the enormous amounts of photographs that had *already been captured*. The philosophy behind our work is somewhat similar: we attempt to leverage the large amount of textured geometric models that have *already been created*. But unlike image databases, which consist mostly of unrelated items, the geometric models we use are all anchored to the world that surrounds us.

**Dehazing.** Weather and other atmospheric phenomena, such as haze, greatly reduce the visibility of distant regions in images of outdoor scenes. Removing the effect of haze, or *dehazing*, is a challenging problem, because the degree of this effect at each pixel depends on the depth of the corresponding scene point.

Some haze removal techniques make use of multiple images; e.g., images taken under different weather conditions [Narasimhan and Nayar 2003a], or with different polarizer orientations [Schechner et al. 2003]. Since we are interested in dehazing single images, taken without any special equipment, such methods are not suitable for our needs.

There are several works that attempt to remove the effects of haze, fog, etc., from a single image using some form of depth information. For example, Oakley and Satherley [1998] dehaze aerial imagery using estimated terrain models. However, their method involves estimating a large number of parameters, and the quality of the reported results is unlikely to satisfy today's digital photography enthusiasts. Narasimhan and Nayar [2003b] dehaze single images based on a rough depth approximation provided by the user, or derived from satellite orthophotos. The very latest dehazing methods [Fattal 2008; Tan 2008] are able to dehaze single images by making various assumptions about the colors in the scene.

Our work differs from these previous single image dehazing methods in that it leverages the availability of more accurate 3D models,

and uses a novel data-driven dehazing procedure. As a result, our method is capable of effective, stable high-quality contrast restoration even of extremely distant regions.

**Novel view synthesis.** It has been long recognized that adding depth information to photographs provides the means to alter the viewpoint. The classic "Tour Into the Picture" system [Horry et al. 1997] demonstrates that fitting a simple mesh to the scene is sometimes enough to enable a compelling 3D navigation experience. Subsequent papers, Kang [1998], Criminisi *et al.* [2000], Oh *et al.* [2001], Zhang *et al.* [2002], extend this by providing more sophisticated, user-guided 3D modelling techniques. More recently Hoiem *et al.* [2005] use machine learning techniques in order to construct a simple "pop-up" 3D model, completely automatically from a single photograph. In these systems, despite the simplicity of the models, the 3D experience can be quite compelling.

In this work, we use already available 3D models in order to add depth to photographs. We present a new model-guided image completion technique that enables us to expand the field of view and to perform high-quality novel view synthesis.

**Relighting.** A number of sophisticated relighting systems have been proposed by various researchers over the years (e.g., [Yu and Malik 1998; Yu et al. 1999; Loscos et al. 2000; Debevec et al. 2000]). Typically, such systems make use of a highly accurate geometric model, and/or a collection of photographs, often taken under different lighting conditions. Given this input they are often able to predict the appearance of a scene under novel lighting conditions with a very high degree of accuracy and realism. Another alternative to use a time-lapse video sequence [Sunkavalli et al. 2007]. In our case, we assume the availability of a geometric model, but have just one photograph to work with. Furthermore, although the model might be detailed, it is typically quite far from a perfect match to the photograph. For example, a tree casting a shadow on a nearby building will typically be absent from our model. Thus, we cannot hope to correctly recover the reflectance at each pixel of the photograph, which is necessary in order to perform physically accurate relighting. Therefore, in this work we propose a very simple relighting approximation, which is nevertheless able to produce fairly compelling results.

**Photo browsing.** Also related is the "Photo Tourism" system [Snavely et al. 2006], which enables browsing and exploring large collections of photographs of a certain location using a 3D interface. But, the browsing experience that we provide is very different. Moreover, in contrast to "Photo Tourism", our system requires only a single geo-tagged photograph, making it applicable even to locations without many available photos.

The "Photo Tourism" system also demonstrates the transfer of annotations from one registered photograph to another. In Deep Photo, photographs are registered to a model of the *world*, making it possible to tap into a much richer source of information.

**Working with geo-referenced images.** Once a photo is registered to geo-referenced data such as maps and 3D models, a plethora of information becomes available. For example, Cho [Cho 2007] notes that absolute geo-locations can be assigned to individual pixels and that GIS annotations, such as building and street names, may be projected onto the image plane. Deep Photo supports similar labeling, as well as several additional visualizations, but in contrast to Cho's system, it does so dynamically, in the context of an interactive photo browsing application. Furthermore, as discussed earlier, it also enables a variety of other applications.

In addition to enhancing photos, location is also useful in organizing and visualizing photo collections. The system developed by Toyama *et al.* [2003] enables a user to browse large collections of geo-referenced photos on a 2D map. The map serves as both a visualization device, as well as a way to specify spatial queries, i.e., all photos within a region. In contrast, DeepPhoto focuses on enhancing and browsing of a single photograph; the two systems are actually complementary, one focusing on organizing large photo collections, and the other on enhancing and viewing single photographs.

## 3 Registration and Matting

We assume that the photograph has been captured by a simple pinhole camera, whose parameters consist of position, pose, and focal length (seven parameters in total). To register such a photograph to a 3D geometric model of the scene, it suffices to specify four or more corresponding pairs of points [Gruen and Huang 2001]. Assuming that the rough position from which the photograph was taken is available (either from a geotag, or provided by the user), we are able to render the model from roughly the correct position, let the user specify sufficiently many correspondences, and recover the parameters by solving a nonlinear system of equations [Nister and Stewenius 2007]. The details and user interface of our registration system are described in a technical report [Chen et al. 2008].

For images that depict foreground objects not contained in the model, we ask the user matte out the foreground. For the applications demonstrated in this paper the matte does not have to be too accurate, so long as it is conservative (i.e., all the foreground pixels are contained). We created mattes with the Soft Scissors system [Wang et al. 2007]. The process took about 1-2 minutes per photo. For every result produced using a matte we show the matte next to the input photograph.

## 4 Image Enhancement

Many of the typical images we take are of a spectacular, often well known, landscape or cityscape. Unfortunately in many cases the lighting conditions or the weather are not optimal when the photographs are taken, and the results may be dull or hazy. Having a sufficiently accurate match between a photograph and a geometric model offers new possibilities for enhancing such photographs. We are able to easily remove haze and unwanted color shifts and to experiment with alternative lighting conditions.

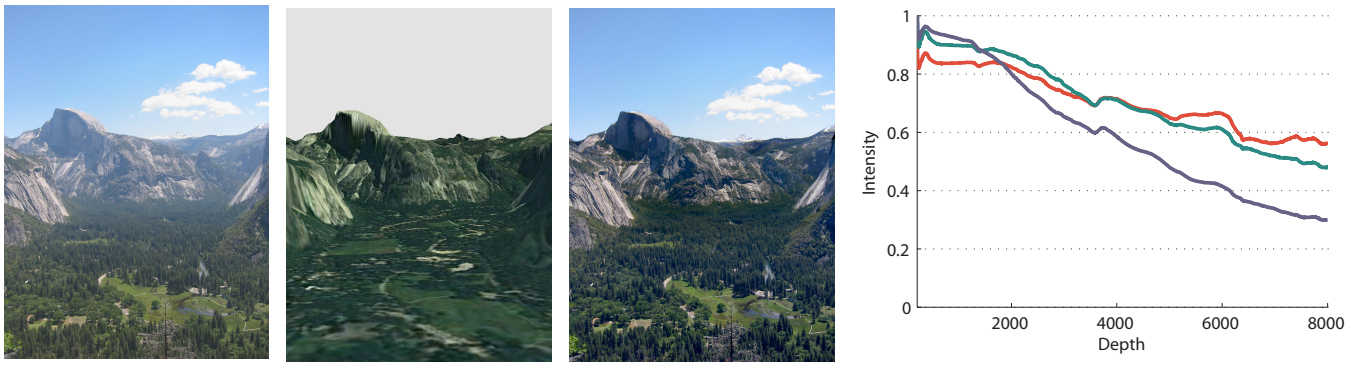
### 4.1 Dehazing

Atmospheric phenomena, such as haze and fog can reduce the visibility of distant regions in images of outdoor scenes. Due to atmospheric absorption and scattering, only part of the light reflected from distant objects reaches the camera. Furthermore, this light is mixed with *airlight* (scattered ambient light between the object and camera). Thus, distant objects in the scene typically appear considerably lighter and featureless, compared to nearby ones.

If the depth at each image pixel is known, in theory it should be easy to remove the effects of haze by fitting an analytical model (e.g., [McCartney 1976; Nayar and Narasimhan 1999]):

$$I_h = I_o f(z) + A\,(1 - f(z))\,. \tag{1}$$

Here $I_h$ is the observed hazy intensity at a pixel, $I_o$ is the original intensity reflected towards the camera from the corresponding scene point, $A$ is the airlight, and $f(z) = \exp(-\beta z)$ is the attenuation in intensity as a function of distance due to outscattering. Thus, after




Input | Model textures | Final dehazed result | Estimated haze curves $f(z)$

**Figure 2:** *Dehazing. Note the artifacts in the model texture, and the significant deviation of the estimated haze curves from exponential shape.*

Input | Dehazed | Input | Dehazed

**Figure 3:** *More dehazing examples.*

estimating the parameters $A$ and $\beta$ the original intensity may be recovered by inverting the model:

$$I_o = A + (I_h - A)\frac{1}{f(z)}. \tag{2}$$

As pointed out by Narasimhan and Nayar [2003a], this model assumes single-scattering and a homogeneous athmosphere. Thus, it is more suitable for short ranges of distance and might fail to correctly approximate the attenuation of scene points that are more than a few kilometers away. Furthermore, since the exponential attenuation goes quickly down to zero, noise might be severely amplified in the distant areas. Both of these artifacts may be observed in the "inversion result" of Figure 4.

While reducing the degree of dehazing [Schechner et al. 2003] and regularization [Schechner and Averbuch 2007; Kaftory et al. 2007] may be used to alleviate these problems, our approach is to estimate stable values for the haze curve $f(z)$ directly from the relationship between the colors in the photograph and those of the model textures. More specifically, we compute a curve $f(z)$ and an airlight $A$, such that eq. (2) would map averages of colors in the photograph to the corresponding averages of (color-corrected) model texture colors. Note that although our $f(z)$ has the same physical interprertation as in the previous approaches, due to our estimation process it is not subject to the constraints of a physicially-based model. Since we estimate a single curve to represent the possibly spatially varying haze it can also contain non-monotonicities. All of the parameters are estimated completely automatically.

For robustness, we operate on averages of colors over depth ranges. For each value of $z$, we compute the average model texture color $\hat{I}_m(z)$ for all pixels whose depth is in $[z-\delta, z+\delta]$, as well as the average hazy image color $\hat{I}_h(z)$ for the same pixels. In our implementation, the depth interval parameter $\delta$ is set to 500 meters, for all images we experimented with. The averaging makes our approach less sensitive to model texture artifacts, such as registration and stitching errors, bad pixels, or contained shadows and clouds.

Before explaining the details of our method, we would like to point out that the model textures typically have a global color bias. For example, Landsat uses seven sensors whose spectral responses differ from the typical RGB camera sensors. Thus, the colors in the resulting textures are only an approximation to ones that would have been captured by a camera (see Figure 2). We correct this color bias by measuring the ratio between the photo and the texture colors in the foreground (in each channel), and using these ratios to correct the colors of the entire texture. More precisely, we compute a global multiplicative correction vector $C$ as

$$C = \frac{F_h}{\mathrm{lum}(F_h)} \Big/ \frac{F_m}{\mathrm{lum}(F_m)}, \tag{3}$$


**Figure 4:** *Comparison with other dehazing methods. The second row shows full-resolution zooms of the region indicated with a red rectangle in the input photo. See the supplementary materials for more comparison images.*

where $F_h$ is the average of $\hat{I}_h(z)$ with $z < z_F$, and $F_m$ is a similarly computed average of the model texture. $\text{lum}(c)$ denotes the luminance of a color $c$. We set $z_F$ to 1600 meters for all our images.

Now we are ready to explain how to compute the haze curve $f(z)$. Ignoring for the moment the physical interpretation of $A$ and $f(z)$, note that eq. (2) simply stretches the intensities of the image around $A$, using the scale coefficient $f(z)^{-1}$. Our goal is to find $A$ and $f(z)$ that would map the hazy photo colors $\hat{I}_h(z)$ to the color-corrected texture colors $C\hat{I}_m(z)$. Substituting $\hat{I}_h(z)$ for $I_h$, and $C\hat{I}_m(z)$ for $I_o$, in eq. (2) we get

$$f(z) = \frac{\hat{I}_h(z) - A}{C\hat{I}_m(z) - A}. \tag{4}$$

Different choices of $A$ will result in different scaling curves $f(z)$. We set $A = 1$ since this guarantees $f(z) \geq 0$. Using $A > 1$ would result in larger values of $f(z)$, and hence less contrast in the dehazed image, and using $A < 1$ might be prone to instabilities. Figure 2 shows the $f(z)$ curve estimated as described above.

The recovered haze curve $f(z)$ allows to effectively restore the contrasts in the photo. However, the colors in the background might undergo a color shift. We compensate for this by adjusting $A$, while keeping $f(z)$ fixed, such that after the change the dehazing preserves the colors of the photo in the background.

To adjust $A$, we first compute the average background color $B_h$ of the photo as the average of $\hat{I}_h(z)$ with $z > z_B$, and a similarly computed average of the model texture $B_m$. We set $z_B$ to 5000m for all our images. The color of the background is preserved, if the ratio

$$R = \frac{A + (B_h - A) \cdot f^{-1}}{B_h}, \quad f = \frac{B_h - 1}{B_m - 1}, \tag{5}$$

has the same value for every color channel. Thus, we rewrite eq. (5) to obtain $A$ as

$$A = B_h \frac{R - f^{-1}}{1 - f^{-1}}, \tag{6}$$

and set $R = \max(B_{m,red}/B_{h,red},\ B_{m,green}/B_{h,green},\ B_{m,blue}/B_{h,blue})$. This particular choice of $R$ results in the maximum $A$ that guarantees $A \leq 1$. Finally, we use eq. (2) with the recovered $f(z)$ and the adjusted $A$ to dehaze the photograph.

Figures 2 and 3 show various images dehazed with our method. Figure 4 compares our method with other approaches. In this comparison we focused on methods that are applicable in our context of working with a single image only. Fattal's method [2008] dehazes the image nicely up to a certain distance (particularly considering that this method does not require any input in addition to the image itself), but it is unable to effectively dehaze the more distant parts, closer to the horizon. The "Inversion Result" was obtained via eq. (2) with an exponential haze curve. This is how dehazing was performed in a number of papers, e.g., [Schechner et al. 2003; Narasimhan and Nayar 2003a; Narasimhan and Nayar 2003b]. Here, we use our accurate depth map instead of using multiple images or user-provided depth approximations. The airlight color was set to the sky color near the horizon, and the optical depth $\beta$ was adjusted manually. The result suffers from amplified noise in the distance, and breaks down next to the horizon. In contrast, our result manages to remove more haze than the two other approaches, while preserving the natural colors of the input photo.

Note that in practice one might not want to remove the haze completely as we have done, because haze sometimes provides perceptually significant depth cues. Also, dehazing typically amplifies some noise in regions where little or no visible detail remain in the original image. Still, almost every image benefits from some degree of dehazing.

Having obtained a model for the haze in the photograph we can insert new objects into the scene in a more seamless fashion by applying the model to these objects as well (in accordance with the depth they are supposed to be at). This is done simply by inverting eq. (2):

$$I_h = A + (I_o - A) f(z). \tag{7}$$

This is demonstrated in the companion video.

### 4.2 Relighting

One cannot underestimate the importance of the role that lighting plays in the creation of an interesting photograph. In particular, in landscape photography, the vast majority of breathtaking photographs are taken during the "golden hour", after sunrise, or before sunset [Reichmann 2001]. Unfortunately most of our outdoor snapshots are taken under rather boring lighting. With Deep Photo

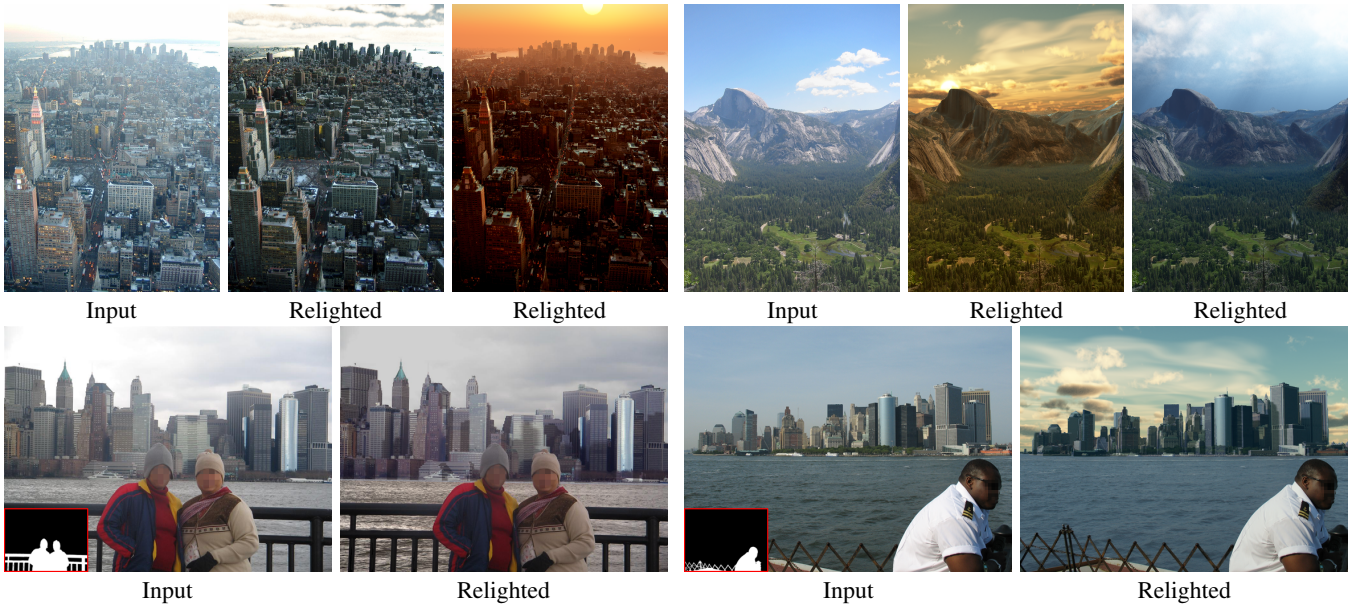



**Figure 5:** *Relighting results produced with our system.*


**Figure 6:** *A comparison between the original photo, its relighted version, and a rendering of the underlying model under the same illumination.*

it is possible to modify the lighting of a photograph, approximating what the scene might look like at another time of day.

As explained earlier, our goal is to work on single images, augmented with a detailed, yet not completely accurate geometric model of the scene. This setup does not allow us to correctly recover the reflectance at each pixel. Thus, we use the following simple workflow, which only approximates the appearance of lighting changes in the scene. We begin by dehazing the image, as described in the previous section, and modulate the colors using a lightmap computed for the novel lighting. The original sky is replaced by a new one simulating the desired time of day (we use Vue 6 Infinite [E-on Software 2008] to synthesize the new sky). Finally, we add haze back in using Eq. (7), after multiplying the haze curves $f(z)$ by a global color mood transfer coefficient.

The global color mood transfer coefficient $L_G$ is computed for each color channel. Two sky domes are computed, one corresponding to the actual (known or estimated) time of day the photograph was taken, and the other corresponding to the desired sun position. Let $I_{\text{ref}}$ and $I_{\text{new}}$ be the average colors of the two sky domes. The color mood transfer coefficients are then given by $L_G = I_{\text{new}}/I_{\text{ref}}$.

The lightmap may be computed in a variety of ways. Our current implementation offers the user a set of controls for various aspects of the lighting, including atmosphere parameters, diffuse and ambient colors, etc. We then compute the lightmap with a simple local shading model and scale it by the color mood coefficient:

$$L = L_G \cdot L_S \cdot (L_A + L_D \cdot (\mathbf{n} \cdot \mathbf{l})), \tag{8}$$

where $L_S \in [I_{shadow}, 1]$ is the shadow coefficient that indicates the amount of light attenuation due to shadows, $L_A$ is the ambient coefficient, $L_D$ is the diffuse coefficient, $\mathbf{n}$ the point normal, and $\mathbf{l}$ the direction to the sun. The final result is obtained simply by multiplying the image by $L$.

Note that we do not attempt to remove the existing illumination before applying the new one. However, we found even this basic procedure yields convincing changes in the lighting (see Figure 5, and the dynamic relighting sequences in the video). Figure 6 demonstrates that relighting a geo-registered photo generates a completely different (and more realistic) effect than simply rendering the underlying geometric model under the desired lighting.

## 5 Novel View Synthesis

One of the compelling features of Deep Photo is the ability to modify the viewpoint from which the original photograph was taken. Bringing the static photo to life in this manner significantly enhances the photo browsing experience, as shown in the companion video.

Assuming that the photograph has been registered with a sufficiently accurate geometric model of the scene, the challenge in changing the viewpoint is reduced to completing the missing texture in areas that are either occluded, or are simply outside the original view frustum. We use image completion [Efros and Leung 1999; Drori et al. 2003] to fill the missing areas with texture from

other parts of the photograph. Our image completion process is similar to texture-by-numbers [Hertzmann et al. 2001], where instead of a hand-painted label map we use a *guidance map* derived from the textures of the 3D model. In rural areas these are typically aerial images of the terrain, while in urban models these are the texture maps of the buildings.

The texture is synthesized over a cylindrical layered depth image (LDI) [Shade et al. 1998], centered around the original camera position. The LDI image stores, for each pixel, the depths and normals of scene points intersected by the corresponding ray from the viewpoint. We use this data structure, since it is able to represent both the visible and the occluded parts of the scene (in our examples we used a LDI with four depth layers per pixel). The colors of the frontmost layer in each pixel are taken from the original photograph provided that they are inside the original view frustum, while the remaining colors are synthesized by our guided texture transfer.

We begin the texture transfer process by computing the guiding value for all of the layers at each pixel. The guiding value is a vector $(U,V,D)$, where $U$ and $V$ are the chrominance values of the corresponding point in the model texture, and $D$ is the distance to the corresponding scene point from the location of the camera. In our experiments, we tried various other features, including terrain normal, slope, height, and combinations thereof. We achieved the best results, however, with the realtively simple feature vector above. Including the distance $D$ in the feature vector biases the synthesis towards generating textures at the correct scale. $D$ is normalized so that distances from 0 to 5000 meters map to $[0,1]$. We only include chrominance information in the feature vector (and not luminance) to alleviate problems associated with existing transient features such as shading and shadows in the model textures.

Texture synthesis is carried out in a multi-resolution manner. The first (coarsest) level is synthesized by growing the texture outwards from the known regions. For each unknown pixel we examine a square neighborhood around it, and exhaustively search for the best matching neighborhood from the known region (using the $L_2$ norm). Since our neighborhoods contain missing pixels we cannot apply PCA compression and other speed-up structures in a straight forward way. However, the first level is sufficiently coarse and its synthesis is rather fast. To synthesize each next level we upsample the result of the previous level and perform a small number of $k$-coherence synthesis passes [Ashikhmin 2001] to refine the result. Here we use a $5\times 5$ look-ahead region and $k=4$. The total synthesis time is about 5 minutes per image. The total texture size is typically on the order of $4800\times 1600$ pixels, times four layers.

It should be noted that when working with LDIs the concept of a pixel's neighborhood must be adjusted to account for the existence of multiple depth layers at each pixel. We define the neighborhood in the following way: On each depth layer, a pixel has up to 8 pixels surrounding it. If the neighboring pixel has multiple depth layers, the pixel on the layer with the closest depth value is assigned as the immediate neighbor.

To render images from novel viewpoints, we use a shader to project the LDI image onto the geometric model by computing the distance of the model to the camera and using the pixel color from the depth layer closest to this distance. Significant changes in the viewpoint eventually cause texture distortions if one keeps using the texture from the photograph. To alleviate this problem, we blend the photograph's texture into the model's texture as the new virtual camera gets farther away from the original viewpoint. We found this to significantly improve the 3D viewing experience, even for drastic view changes, such as going to bird's eye view.

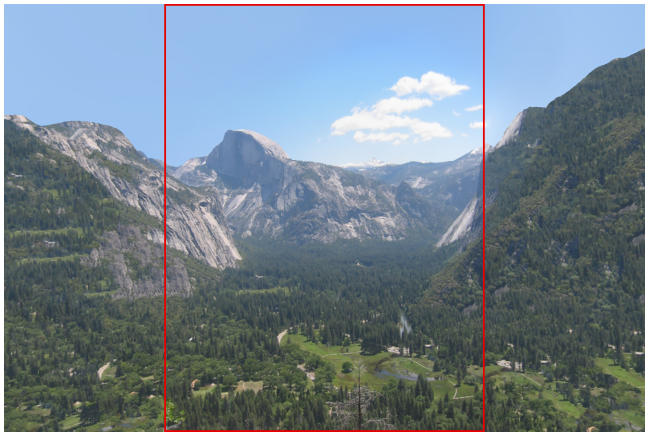


**Figure 7:** *Extending the field of view. The red rectangle indicates the boundaries of the original photograph. The companion video demonstrates changing the viewpoint.*

Thus, the texture color $T$ at each terrain point $\mathbf{x}$ is given by

$$T(\mathbf{x}) = g(\mathbf{x})\,T_{\text{photo}}(\mathbf{x}) + (1-g(\mathbf{x}))\,T_{\text{model}}(\mathbf{x}), \tag{9}$$

where the blending factor $g(\mathbf{x})$ is determined with respect to the current view, according to the following principles: (i) pixels in the original photograph which correspond to surfaces facing camera are considered more reliable than those on oblique surfaces; and, (ii) pixels in the original photograph are also preferred whenever the corresponding scene point is viewed from the same direction in the current view, as it was in the original one.

Specifically, let $\mathbf{n}(\mathbf{x})$ denote the surface normal, $C_0$ the original camera position from which the photograph was taken, and $C_{new}$ the current camera position. Next, let $\mathbf{v}_0 = (C_0-\mathbf{x})/\|C_0-\mathbf{x}\|$ denote the normalized vector from the scene point to the original camera position, and similarly $\mathbf{v}_{new} = (C_{new}-\mathbf{x})/\|C_{new}-\mathbf{x}\|$. Then

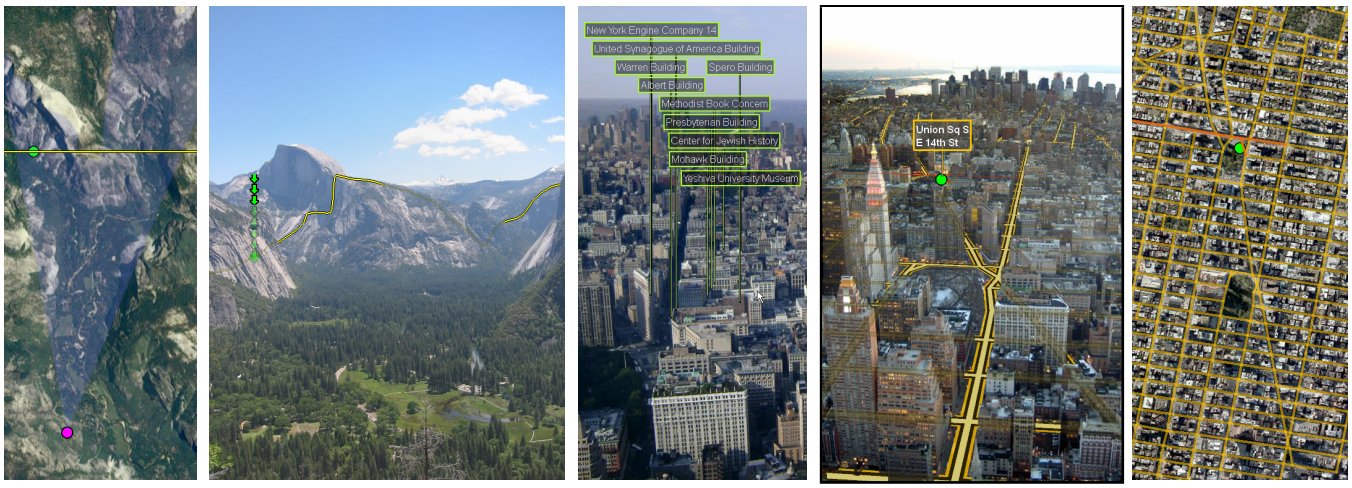
$$g(\mathbf{x}) = \max\left(\mathbf{n}(\mathbf{x})\cdot\mathbf{v}_0, \mathbf{v}_{new}\cdot\mathbf{v}_0\right). \tag{10}$$

In other words, $g$ is defined as the greater among the cosine of the angle between the normal and the original view direction, and the cosine of the angle between the two view directions.

Finally, we also apply *re*-hazing on-the-fly. First, we remove haze from the texture completely as described in Section 4.1. Then, we add haze back in, this time using the distances from the current camera position. The results may be seen in Figure 7 and in the video.

## 6 Information Visualization

Having registered a photograph with a model that has GIS data associated with it allows displaying various information about the scene, while browsing the photograph. We have implemented a simple application that demonstrates several types of information visualization. In this application, the photograph is shown side-by-side with a top view of the model, referred to as the *map view*. The view frustum corresponding to the photograph is displayed in the map view, and is updated dynamically whenever the view is changed (as described in Section 5). Moving the cursor in either of the two views highlights the corresponding location in the other view. In the map view, the user is able to switch between a street map, an orthographic photo, a combination thereof, etc. In addition




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

**Figure 8:** *Different information visualization modes in our system. (a-b) Coupled map and photo views. As the user moves the mouse over one of the views, the corresponding location is shown in the other view as well. The profile of a horizontal scanline in the map view (a) is shown superimposed over the terrain in the photo view (b). Since the location of the mouse cursor is occluded by a mountain in the photo, its location in the photo view is indicated using semi-transparent arrows. (c) Names of landmarks are automatically superimposed on the photo. (d-e) Coupled photo and map views with superimposed street network. The streets under the mouse cursor are highlighted in both views.*

to text labels it is also possible to superimpose graphical map elements, such as roads, directly onto the photo view. These abilities are demonstrated in Figures 1 and 8 and in the companion video.

There are various databases with geo-tagged media available on the web. We are able to highlight these locations in both views (photo and map). Of particular interest are geo-tagged Wikipedia articles about various landmarks. We display a small Wikipedia icon at such locations, which opens a browser window with the corresponding article, when clicked. This is also demonstrated in the companion video.

Another nice visualization feature of our system is the ability to highlight the object under the mouse in the photo view. This can be useful, for example, when viewing night time photographs: in an urban scene shot at night, the building under the cursor may be shown using daylight textures from the underlying model.

## 7 Discussion and Conclusions

We presented Deep Photo, a novel system for editing and browsing outdoor photographs. It leverages the high quality 3D models of the earth that are now becoming widely available. We have demonstrated that once a simple geo-registration of a photo is performed, the models can be used for many interesting photo manipulations that range from de- and rehazing and relighting to integrating GIS information.

The applications we show are varied. Haze removal is a challenging problem due to the fact that haze is a function of depth. We have shown that now that depth is available in a geo-registered photograph, excellent "haze editing" can be achieved. Similarly, having an underlying geometric model makes it possible to generate convincing relighted photographs, and dynamically change the view. Finally, we demonstrate that the enormous wealth of information available online can now be used to annotate and help browse photographs.

Within our framework we used models obtained from Virtual Earth. The manual registration is done within a minute, matting out the foreground is also an easy task using state-of-the-art techniques such as Soft Scissors [Wang et al. 2007]. All other operations such as dehazing and relighting run at interactive speeds; however, computing very detailed shadow maps for the relighting can be time consuming.

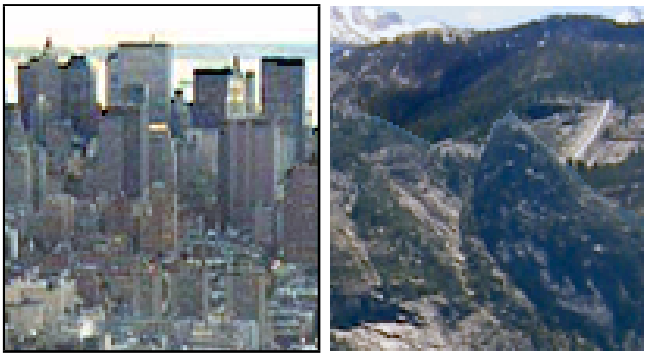


**Figure 9:** *Failure cases: some of the described applications produce artifacts for badly registered (left) and/or insuffienctly accurate models (right). In this case the dehazing application generated halos around misaligned depth edges because it used wrong depth values there. The same artifacts can be observed by zooming into the full images in Figures 2 and 3.*

As can be expected, there are always some differences and misalignments between the photograph and the model. The may arise due to insufficiently accurate models, and also due to the fact that the photographs were not captured with an ideal pinhole camera. Although they can lead to some artifacts (see Figure 9), we found that in many cases these differences are less problematic than one might fear. However, automatically resolving such differences is certainly a challenging and interesting topic for future work.

We believe that the applications presented here represent just a small fraction of possible geo-photo editing operations. Many of the existing digital photography products could be greatly enhanced

with the use of geo information. Operations could encompass noise-reduction and image sharpening with 3D model priors, post-capture refocussing, object recovery in under or over-exposed areas as well as illumination transfer between photographs.

GIS databases contain a wealth of information, of which we have just used a small amount. Water, grass, pavement, building materials, etc, can all potentially be automatically labeled and used to improve photo tone adjustment. Labels can be transferred automatically from one image to others. Again, having a single consistent 3D model for our photographs provides much more than just a depth value per pixel.

In this paper we mostly dealt with single images. Most of the applications that we demonstrated become even stronger when combining multiple input photos. A particularly interesting direction might be to combine Deep Photo with the Photo Tourism system. Once a Photo Tour is geo-registered, the coarse 3D information generated by Photo Tourism could be used to enhance online 3D data and vice-versa. The information visualization and novel view synthesis applications we demonstrate here could be combined with the Photo Tourism viewer. This idea of fusing multiple images could even be extended to video that could be registered to the models.

### Acknowledgements

This research was supported in parts by grants from the the following funding agencies: the Lion foundation, the GIF foundation, the Israel Science Foundation, and by DFG Graduiertenkolleg/1042 "Explorative Analysis and Visualization of Large Information Spaces" at University of Konstanz, Germany.

## References

ASHIKHMIN, M. 2001. Synthesizing natural textures. *Proceedings of the 2001 symposium on Interactive 3D graphics (I3D)*, 217–226.

CHEN, B., RAMOS, G., OFEK, E., COHEN, M., DRUCKER, S., AND NISTER, D. 2008. Interactive techniques for registering images to digital terrain and building models. *Microsoft Research Technical Report MSR-TR-2008-115*.

CHO, P. L. 2007. 3D organization of 2D urban imagery. *Proceedings of the 36th Applied Imagery Pattern Recognition Workshop*, 3–8.

CRIMINISI, A., REID, I. D., AND ZISSERMAN, A. 2000. Single view metrology. *International Journal of Computer Vision 40*, 2, 123–148.

DEBEVEC, P. E., TAYLOR, C. J., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Proceedings of SIGGRAPH '96*, 11–20.

DEBEVEC, P., HAWKINS, T., TCHOU, C., DUIKER, H.-P., SAROKIN, W., AND SAGAR, M. 2000. Acquiring the reflectance field of a human face. *Proceedings of SIGGRAPH 2000*, 145–156.

DRORI, I., COHEN-OR, D., AND YESHURUN, H. 2003. Fragment-based image completion. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2003) 22*, 3, 303–312.

E-ON SOFTWARE, 2008. Vue 6 Infinite. `http://www.e-onsoftware.com/products/vue/vue_6_infinite`.

EFROS, A. A., AND LEUNG, T. K. 1999. Texture synthesis by non-parametric sampling. *Proceedings of IEEE International Conference on Computer Vision (ICCV) '99 2*, 1033–1038.

FATTAL, R. 2008. Single image dehazing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2008) 27*, 3, 73.

FRÜH, C., AND ZAKHOR, A. 2003. Constructing 3D city models by merging aerial and ground views. *IEEE Computer Graphics and Applications 23*, 6, 52–61.

GRUEN, A., AND HUANG, T. S. 2001. *Calibration and Orientation of Cameras in Computer Vision*. Springer-Verlag, Secaucus, NJ, USA.

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

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. H. 2001. Image analogies. *Proceedings of SIGGRAPH 2001*, 327–340.

HOIEM, D., EFROS, A. A., AND HEBERT, M. 2005. Automatic photo pop-up. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2005) 24*, 3, 577–584.

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. *Proceedings of SIGGRAPH '97*, 225–232.

KAFTORY, R., SCHECHNER, Y. Y., AND ZEEVI, Y. Y. 2007. Variational distance-dependent image restoration. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2007*, 1–8.

KANG, S. B. 1998. Depth painting for image-based rendering applications. Tech. rep., Compaq Cambridge Research Lab.

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

LOSCOS, C., DRETTAKIS, G., AND ROBERT, L. 2000. Interactive virtual relighting of real scenes. *IEEE Transactions on Visualization and Computer Graphics 6*, 4, 289–305.

MCCARTNEY, E. J. 1976. *Optics of the Atmosphere: Scattering by Molecules and Particles*. John Wiley and Sons, New York, NY, USA.

NARASIMHAN, S. G., AND NAYAR, S. K. 2003. Contrast restoration of weather degraded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 6, 713–724.

NARASIMHAN, S. G., AND NAYAR, S. K. 2003. Interactive (de)weathering of an image using physical models. *IEEE Workshop on Color and Photometric Methods in Computer Vision*.

NASA, 2008. The landsat program. `http://landsat.gsfc.nasa.gov/`.

NASA, 2008. Shuttle radar topography mission. `http://www2.jpl.nasa.gov/srtm/`.

NAYAR, S. K., AND NARASIMHAN, S. G. 1999. Vision in bad weather. *Proceedings of IEEE International Conference on Computer Vision (ICCV) '99*, 820–827.

NISTER, D., AND STEWENIUS, H. 2007. A minimal solution to the generalised 3-point pose problem. *Journal of Mathematical Imaging and Vision 27*, 1, 67–79.

OAKLEY, J. P., AND SATHERLEY, B. L. 1998. Improving image quality in poor visibility conditions using a physical model for contrast degradation. *IEEE Transactions on Image Processing 7*, 2, 167–179.

OH, B. M., CHEN, M., DORSEY, J., AND DURAND, F. 2001. Image-based modeling and photo editing. *Proceedings of ACM SIGGRAPH 2001*, 433–442.

REICHMANN, M., 2001. The art of photography. `http://www.luminous-landscape.com/essays/theartof.shtml`.

SCHECHNER, Y. Y., AND AVERBUCH, Y. 2007. Regularized image recovery in scattering media. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 9, 1655–1660.

SCHECHNER, Y. Y., NARASIMHAN, S. G., AND NAYAR, S. K. 2003. Polarization-based vision through haze. *Applied Optics 42*, 3, 511–525.

SHADE, J., GORTLER, S., HE, L.-W., AND SZELISKI, R. 1998. Layered depth images. *Proceedings of SIGGRAPH '98*, 231–242.

SHUM, H.-Y., HAN, M., AND SZELISKI, R. 1998. Interactive construction of 3-d models from panoramic mosaics. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 1998*, 427–433.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: exploring photo collections in 3d. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006) 25*, 3, 835–846.

STAMOS, I., AND ALLEN, P. K. 2000. 3-D model construction using range and image data. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 1998*, 531–536.

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

TAN, R. T. 2008. Visibility in bad weather from a single image. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2008*, to appear.

TOYAMA, K., LOGAN, R., AND ROSEWAY, A. 2003. Geographic location tags on digital images. *Proceedings of the 11th ACM international conference on Multimedia*, 156–166.

WANG, J., AGRAWALA, M., AND COHEN, M. F. 2007. Soft scissors: an interactive tool for realtime high quality matting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007) 26*, 3.

YU, Y., AND MALIK, J. 1998. Recovering photometric properties of architectural scenes from photographs. *Proceedings of SIGGRAPH '98*, 207–217.

YU, Y., DEBEVEC, P., MALIK, J., AND HAWKINS, T. 1999. Inverse global illumination: recovering reflectance models of real scenes from photographs. *Proceedings of SIGGRAPH '99*, 215–224.

ZHANG, L., DUGAS-PHOCION, G., SAMSON, J.-S., AND SEITZ, S. M. 2002. Single-view modelling of free-form scenes. *The Journal of Visualization and Computer Animation 13*, 4, 225–235.