Color Matching and Illumination Estimation for Urban Scenes

Mingxuan Sun, Grant Schindler, Greg Turk, and Frank Dellaert
Georgia Institute of Technology
{cynthia,schindler,turk,dellaert}@cc.gatech.edu

Abstract

Photographs taken of the same scene often look very different, due to various conditions such as the time of day, the camera characteristics, and subsequent processing of the image. Prime examples are the countless photographs of urban centers taken throughout history. In this paper we present an approach to match the appearance between photographs that removes effects such as different camera settings, illumination, fading ink and paper discoloration over time, and digitization artifacts. Global histogram matching techniques are inadequate for appearance matching of complex scenes where background, light, and shadow can vary drastically, making correspondence a difficult problem. We alleviate this correspondence problem by registering photographs to 3D models of the scene. In addition, by estimating the calendar date and time of day, we can additionally remove the effect of drastic lighting and shadow differences between the photographs. We present results for the case of urban scenes, and show that our method allows for realistic visualizations by blending information from multiple photographs without color-matching artifacts.

Figure 1. We want to visualize urban scenes evolving over time, given a collection of images taken by different cameras from different viewpoints at different times. In the above example scene, not only have the images been captured under different conditions, but the geometry has also changed over the years 1966 to 1978.

1. Introduction

We are interested in color-matching images of urban scenes for the purpose of visualizing them across space and time. In the urban context, where there is a richness of historical images, we are specifically interested in the visualization of both large and small temporal changes in a city from any chosen viewpoint. Examples include observing the changes of the evolving cityscape over many decades, and observing a cityscape changing from sunrise to sunset on an arbitrary day.

Modeling and interactively visualizing large-scale 3D urban scenes from images has been a very successful approach, as evidenced by commercial programs such as Google Earth and Microsoft's Virtual Earth, which speak to the public imagination. The Facade project by Debevec et al. [4] introduced view-dependent texture mapping for virtual view synthesis, while Photo Tourism [14] successfully applied structure from motion to recover camera poses and a 3D point cloud from 2D images correspondences. Most recently, Schindler et al. [13] expanded the scope towards modeling *time-varying* 3D structures from historical photographs in the 4D Cities project.

Unfortunately, visualizing an urban scene using images taken with different cameras at different times of day, at different times of the year, and in different historical eras is challenging due to many photometric inconsistencies between images. Figure 1 shows an example of five such images[1], illustrating the difficulties. In traditional image-based rendering, the scene is relatively well controlled and the image collections are structured, which means the scene is captured densely by a set of images that are uniformly distributed around the scene in a relatively short period of time [3, 18, 17, 6]. Most previous work assumes constant scene brightness and small geometry changes over a short time, which can hardly be applied to large-scale dynamic scenes like cities over a hundred years.

[1] Historic images are from atlanta history center.

In order to visualize a scene at any time from an arbitrary viewpoint, we would like to recover the reflectance properties (e.g. albedo) of structures in the scene. This task is made difficult by changes in illumination, shadowing, 3D structure, and camera color properties. When dealing with a limited number of historical images captured by dramatically different cameras at unknown dates and times, it is especially hard to infer the photometry of the scene. *Our goal is to recover a uniformly-shaded, shadow-free and photometrically-consistent scene albedo from all the images*. To achieve our goal, the underlying geometry and sun position for each image are needed to efficiently extract and combine textures from images.

In this paper, to visualize a scene from an entire collection of images, we select one photo as the target color tone, and perform color matching of all the other photos to this target, using scene geometry to find corresponding facades across multiple images for local color matching, and using of an estimated sun position for each photo to correct for the varying amounts of sunlight that fall on each facade.

We first address the color matching problem in section 2, where we find that global color matching methods are insufficient. In our urban scene application, where sky appears in most of the outdoor images, *global color mapping over the entire image can not transfer the local object colors consistently between images. Instead we perform local color mapping by identifying facades of the same building that are similarly lit in both images*. Thus, we introduce the geometry model in section 3, which we use in our local color matching technique.

Based on the scene geometry, we propose a novel method to estimate sun position from images with unknown time stamps in section 4. Furthermore, we can estimate the date and time of an image if we know the approximate geographical position of the scene in the image. And then in section 5, with the estimated sun position for each image, we can identify the shading and shadow information in the images and generate a consistent set of object albedos. This information also allows us to match object reflectances between photos, which lets us perform better color transfer between images. Finally in section 6, by adding the shading and shadow information from a virtual sun, we can visualize the scene from an arbitrary view at an arbitrary time.

2. Color Matching Using Color Statistics

We are interested in performing color matching between images for the purpose of visualizing a 4D urban scene from historical images. Color transfer is a widely used method applied to images or videos to make the color tone of one image look like the other. Some techniques [12] transfer *global* color image statistics (mean and variance) between two images, while others [15] segment the images into several color clusters and then transfer color independently across each corresponding region. A third approach [10], used in the video conferencing domain, learns a color mapping function for human faces and then applies this mapping to the whole image. Some other approaches [2, 7, 16] learn an consistent color from multiple images or multiple parts of an image for 3D scenes. In addition, the problem of learning a good color mapping function from a large image database [9] has been studied outside the 4D scene visualization context.

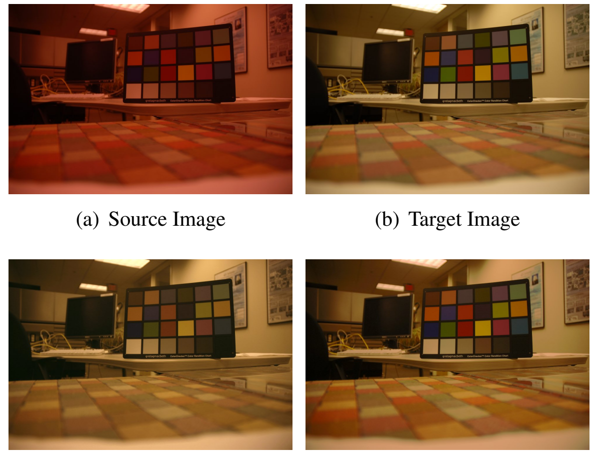
The simplest method of performing color transfer is histogram matching using global image statistics in each color channel. This method matches the mean and variance of the source image S to those of the target image D in each color channel c. For each channel, let (m_S, σ_S) denote the mean and variance of the source image S and (m_D, σ_D) for the target image D. The color value is normalize ranging from 0 to 1. The color transferring function $f(x)$, which map the pixel value x of the source image S towards target image D in each channel c should satisfy these constraints:

$$\begin{aligned} f(m_S) &= f(m_D) \\ f'(m_S) &= \sigma_D/\sigma_S \end{aligned} \quad (1)$$

The simplest such mapping is the linear mapping [12]:

$$f(x) = \frac{\sigma_D}{\sigma_S}(x - m_S) + m_D \quad (2)$$

When portions of the scene are very dark or overly bright, the linear mapping will lead to saturation at lower and upper intensities. To avoid this, we fit a piecewise cubic spline function that satisfy the constrains in 1. In addition, we limit $f(0) = 0, f(1) = 1$ and constrain the derivatives of the



(a) Source Image

(b) Target Image

(c) Global Statistics

(d) Local Statistics

Figure 2. First Row: Left: source image (through red filter), Right: target image (no filter). Bottom row: Left: color matching based on global color statistics, which does not transfer local color correctly. Right: color matching based on local color checker correspondences.

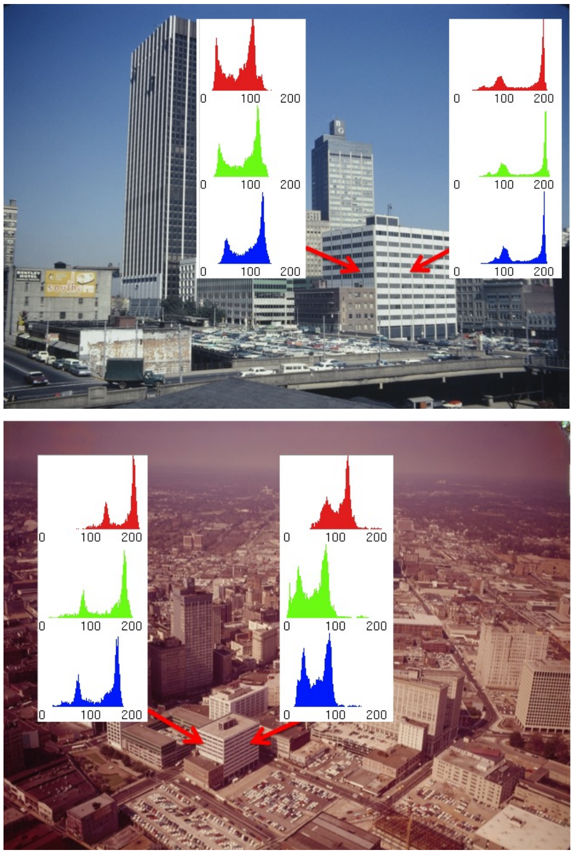


Figure 3. Facades of the same albedo that are lit in both images share histograms of similar shape, the color statistics of which can be used to form the color mapping function. Top image: Left: facade in shadow, Right: facade lit. Bottom image: Left: facade lit, Right: facade in shadow. Each histogram shows red, green, and blue channel information (from top to bottom) for the individual facade regions.

two ends in a way similarly to [10]:

$$f^{'}(0) = 0.5 * \frac{m_D}{m_S}, f^{'}(1) = 0.5 * \frac{1-m_D}{1-m_S} \tag{3}$$

However, using global image statistics cannot transfer local color correctly when background and lighting vary drastically between images. To illustrate the problem, we captured a scene containing a color checker with a camera using different color filters, and performed histogram matching between the images. Figure 2 (lower left) shows the results of performing global histogram matching for the color checker datasets.

Instead of matching the color statistics of the whole image, we use color matching with local correspondences to transfer color between images. The mapping function in equation 2 can be learned by using the mean/variance of local correspondences (e.g. the color checker) in both images, and this mapping can then be applied to every pixel in the whole image. For example, we use the color statistics of the checker in the image to obtain the mapping function and then map the whole image, the results in the lower right of Figure 2 shows that color matching by using local correspondences works better.

Figure 4. Time-Dependent Geometry. Scene geometry in both 1930 (left) and 1972 (right) is overlaid on two photos from their recovered 3D viewpoints. Observe that the geometry changes dramatically over the years. Our illumination estimation and color matching relies on such a time-varying scene description to accurately model shadows and establish corresponding regions across images.

In the context of outdoor urban scene photos, in addition to the geometry correspondence between images, we should also take shading and shadow information into account because the color statistics of the same facade will change dramatically from when it is being lit to when it is in shadow. For example, figure 3 shows the histogram of two facades of one building in two different images. The correspondence of the same facade no longer describes the overall color changes for most of the scene. Only the facades of the same albedo that are lit in both images indicate the overall color statistics changes between the two images.

Based on this observation, we color match between images by using the lit facades that are present in both images. To building up the spatial correspondences between facades in different images, we need the underlying geometry which will be introduced in section 3. Given the geometry model, we can further estimate the illumination in the outdoor scene in section 4 and take the scene illumination into account for color matching.

3. Time-Dependent Geometry and Motion

In order to perform color matching, we require scene geometry for two reasons: (1) to estimate scene illumination from shadows cast by the geometry, and (2) to establish correspondences between image regions in two distinct photographs of the same scene that are taken from different viewpoints. Traditionally, a 3D model of the scene geometry could be used for both the illumination and correspondence tasks (e.g. [8], which registers images to static 3D building models in order to remove haze). However, because we are matching photographs across decades of time during which new buildings are constructed and demolished, we must make use of time-varying 3D geometry which we refer to as a *4D model*.

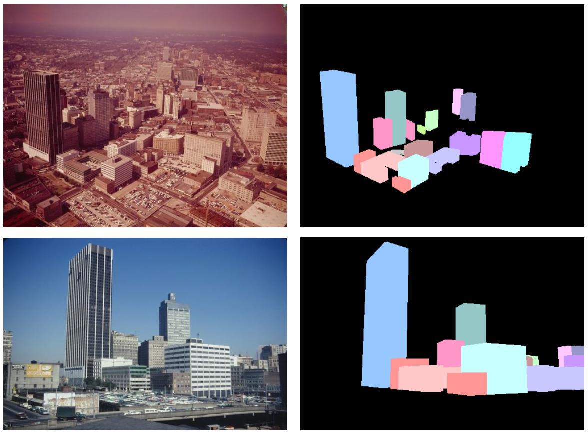


Figure 5. For each photograph (left), a corresponding building segmentation (right) is derived from time-varying geometry and recovered camera motion. Color coding shows the identity of each building and demonstrates how we establish corresponding regions across the two images, despite being captured years apart and from different viewpoints. Though not shown above, individual facade correspondences are established in the same manner.

We define the time-dependent geometry or 4D model S^{4D} as a set of 3D models with associated temporal information. Each 3D model S^{3D} is represented as a set of 3D vertices $\{v_j | j \in 1..N\}$ and a set of polygons $\{f_k | k \in 1..K\}$ connecting these vertices. We do not attempt to model dynamically changing objects such as cars, or planes, etc. Instead, we aim to represent large-scale discrete changes such as buildings appearing and then disappearing at discrete times. Hence, the 4D model S^{4D} we propose is simply a set of 3D models S_r^{3D}, each having an associated time interval t_r, i.e., $S^{4D} \triangleq \{(S_r^{3D}, t_r) | r \in 1..R\}$, with R the number of 3D models in the scene.

We build the 4D model and recover camera motion M from images $\mathbf{I}$ via structure from motion (SFM). We perform SFM with manually matched correspondences to recover 3D positions of building corners which we then manually connect into solid polygonal building models S_r^{3D}. For each photograph we know the year in which it was captured Y, and for each building we know the years of construction and demolition t_r from historical records. Note that such 4D models can also be constructed using the more automated approach presented in [13].

As an example, in the context of an urban scene consisting of buildings, our reconstructed 4D model is shown in Figure 4. Using this 4D model, each individual building and facade can be identified in each image. Figure 5 shows a building-based segmentation of the images, accounting for visibility and occlusion. The whole model (S^{4D}, M, Y) that is obtained serves as a strong geometry proxy for sun direction estimation and virtual view visualization.

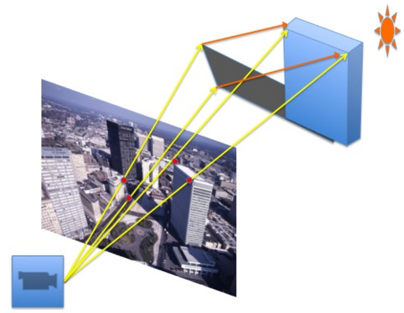


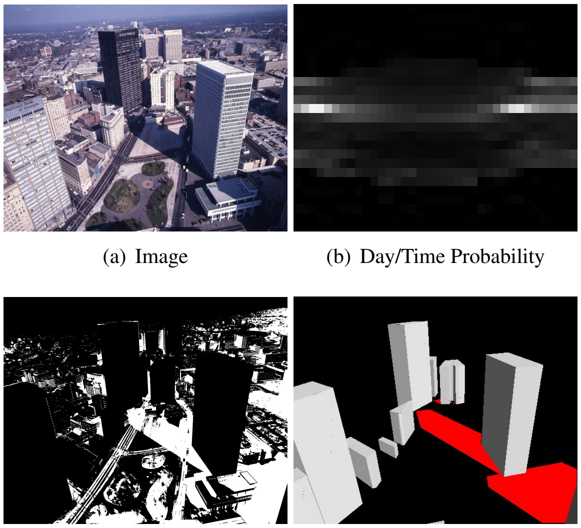
Figure 6. 3D cast shadows of buildings help identify the sun direction, where the sun is modeled as a distant light source. Different sun positions will change the shadow areas dramatically, and we will observe different projected 2D shadows in the image accordingly.

4. Estimating Sun Direction from Images

To account for strong illumination changes in outdoor urban scenes, we take an extra step and attempt to recover the dominant lighting in the scene by estimating the direction of the sun. Most of our historic images do not have exact time stamps, but the buildings do show shading contrast and shadows. We make use of the shadows cast by buildings in an image to identify the sun direction when the image was captured. Figure 6 illustrates the underlying geometry which lets us infer the light direction from the 2D projection of the shadows in the 3D scene. Estimating the sun direction from images is important since it is the most dominant light source. Thus sun position provides considerable information for image-based modeling and rendering applications. Applications such as [19] use time-stamped images collected intensively through out a day from a calibrated camera to compute the sun position and radiance changes, used to recover the reflectance properties in the scene. Similarly, Jacobs et. al [5] infer the geographical location of a static camera by examining how image intensity varies throughout the day. There are methods for estimating general lighting sources from cast shadows such as [1].

Our goal is to find the best sun position given the image. We model the sun as a parallel light source defined by elevation θ and azimuth α (clockwise from north). In order for the sun to properly interact with our geometry, it is important that we align the local coordinate system in which we reconstructed the 3D model with a global coordinate system on the Earth's surface; knowing the true physical location of any three points in the scene produces the required rotation, translation, and scaling from local to global coordinates.

The main idea is to use the geometry as a sundial: we find cast shadows in the scene, and then search for the sun



(a) Image (b) Day/Time Probability

(c) Measured Shadows (d) Predicted Shadows

Figure 7. Illumination Estimation. For an image with unknown time and date (a), we measure a shadow by thresholding the ground region in the image (c) and search over times of day and days of the year for a sun position that predicts the same shadowing pattern. The shadow regions predicted by the estimated most likely sun position are shown in (d). By comparing measured and predicted shadows, we generate a probability distribution (b) over days of the year (horizontal axis from left to right is Jan. 1 to Dec. 31) and times of day (vertical axis from top to bottom is 0 to 23 hours). We predict that the image is taken at Jan. 23, 10 am.

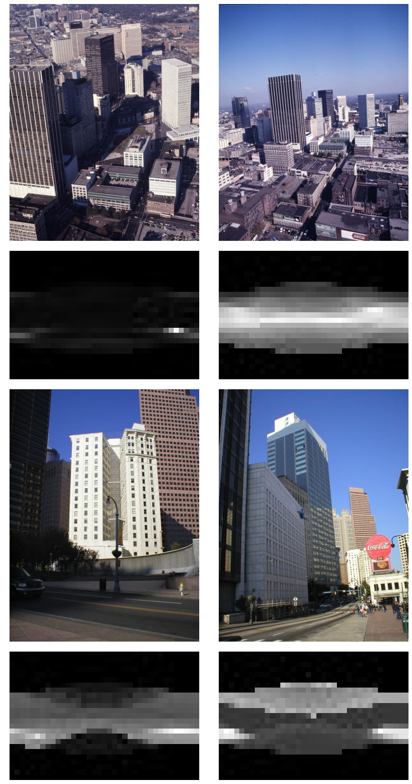


Figure 8. Top: two historic images with unknown dates and times. Bottom: two photos with exif tags, both taken at around 3pm Nov. 9, 2003. We predict the time of day at 3pm and day of year with most probability at two peaks (Nov. 6 and Feb. 9 of the year).

position that best predicts the cast shadows. The shadow in the image can be measured using intensity thresholding in regions of interests. For our initial experiments, an approximate ground shadow map is obtained by thresholding the ground region in the image given the building segmentation. One can further manually mark a region of interest with better accuracy.

To bias the search towards likely sun positions, we do not directly search the azimuth-elevation space but instead search over calendar date and time of day. For a specific geographical location, not all positions of the sun (θ, α) are possible due to the Earth's movement relative to sun. The date within a year and the time of day uniquely determine a position of the sun with respect to a given location [11], so we use data/time instead of angles.

For every date/time combination, we evaluate the corresponding sun direction (θ, α) by a simple sum of square differences (SSD) criterion. We denote the *predicted* shadow image as I_p and the real, *measured* shadow region in the images as I_m. Both I_p and I_m are binary images where the value at each pixel indicates whether the pixel is lit ($I_p(x,y) = 1$) or shadowed ($I_p(x,y) = 0$) in the original photograph. Given the underlying scene geometry, for each sun direction (θ, α), we can easily predict the shadow map I_p. Given the geometry and camera pose, each pixel in the image can be identified as lit by sun or not by the following ray-tracing procedure: first find a pixel's 3D position by backprojecting the pixel onto the aligned 3D geometry, and then shoot a ray from this 3D point along the sun direction. If the ray is not occluded by other geometry, then the pixel is lit, otherwise it is in shadow. We then compute the sum of squared differences (SSD) between pixels in I_p and I_m.

Figure 7 shows the results for estimating sun directions by sampling the date and time of day for each image. In these experiments, we sampled once every 10 days out of a possible 365, and we sample time once per hour in 24 hours. We observe that when the sampling over day and time, the sun probability map is symmetric with respect to June 22, because the summer and winter solstices divide every year into two symmetric halves. Every day between June 22 and December 22 will have a corresponding day in the first half of the year when the sun takes the same apparent path through the sky. So in the date and time estimation, there is always an ambiguity between the two dates which correspond to the same elevation θ and azimuth α of the sun. As such, we can speed up the sun position estimation process by evaluating only half a year's sun positions.

We verify this method on photographs with ground-truth dates and times as shown in Figure 8. For each image, given the estimated sun position at a date and time, we can create a shading/shadow map for the scene at that time. We show how to make use of these results during color matching and virtual view synthesis in the following two sections.

5. Color Matching with Geometry and Shading Correspondences

As discussed in section 2, we perform color matching between images using the statistics of lit facades that are in correspondence across images. The spatial correspondences of facades are identified using the underlying geometry. The sun position allows us to identify lit facades and to normalize the amounts of lighting falling on the facades in each image. This helps us to extract the uniformly lit facades from each image that can be used as textures for scene visualization.

As shown in Figure 9, for each photo, we first select the facades that are lit by the sun. Then assuming the facade is Lambertian, we can determine the object albedo by dividing the per-pixel intensities on a given facade by the cosine between the facade's normal and the direction to the sun. Then the mean and variance of the extracted shadow-free and shading corrected textures will be the constraints in 1 to form the color mapping function.

We then perform color transfer between the images based the color statistics of the extracted object albedos in both images. The mapping results are shown in Figure 10 and 11. We can see that the facades are better matched between images when we use the local correspondences.

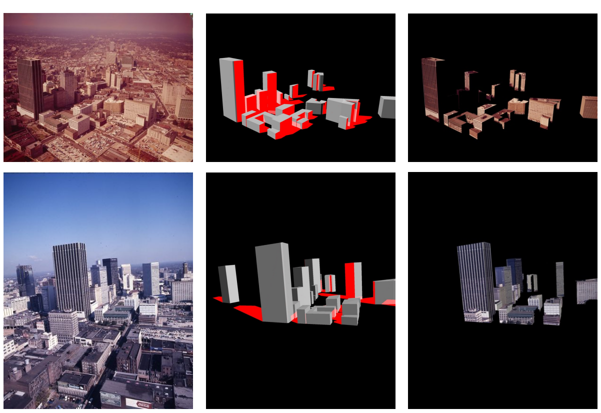


Figure 9. For each photograph (left), a corresponding building segmentation with shading and shadow information is derived from time-varying geometry, recovered camera motion and sun directions (middle). Shadow-free, shading corrected albedo texture are extracted (right) for color matching.

(a) Source Image (b) Target Image

(c) Global Color Match (d) Local + Lit Color Match

Figure 10. When adjusting the color of the source image (a) to match the target (b), our local, lit facade correspondence technique (d) produces more accurate results than the traditional global color matching method (c).

(a) Source Image (b) Target Image

(c) Global Color Match (d) Local + Lit Color Match

Figure 11. When adjusting the color of the source image (a) to match the target (b), our local, lit facade correspondence technique (d) produces more accurate results than the traditional global color matching method (c).

6. Virtual View Synthesis at a Virtual Time and Results

One of our main goals is to visualize the reconstructed urban scene from different viewpoints and lighting conditions. In order to construct high quality object albedos (textures) for the various building facades, we select the texture detail for a given facade from the image that best shows that particular building face. This means that different facades come from different photos, making it necessary to match colors across these photos. To generate object albedos for the entire collection of images, we select one photo as the target color tone, and perform color matching of all the other photos to this target. We then make use of the known sun position for each photo to correct for the varying amounts of sunlight that fall on each facade, making a Lambertian assumption about the facade reflectance. Note that we only select facades that are lit by the sun in order to generate facade albedos. Given the source images in Figure 1, Figure 12 (top) shows how poorly mismatched the facades are if color matching and illumination are ignored. Figure 12 (bottom) shows the color matched and illumination corrected versions of the facade albedos. Note that the models in this scene are not lit by any virtual lights, so that the image only shows the per-object albedos.

With the color matched and illumination corrected facade albedos, we can now create new views of the reconstructed urban scene. These views may be from any angle, with arbitrary illumination, and at any historic time that is represented in our 4D urban model. To synthesize the scene on a given date and time, we first compute the sun position at that time and then use this information to correctly illuminate each object. To demonstrate this, we have created a synthetic time-lapse video of an urban scene from sunrise to sunset on Nov. 6, and snapshots from this video are showed in Figure 13. Note that the path of the virtual sun is a close match to the true sun path for that given date.

7. Conclusion

We have demonstrated an approach that uses estimation of sun position and correspondences of facades across photographs to build and visualize 4D models of urban scenes. The illumination estimation and facade correspondences are vital for performing color matching across photographs that were taken with un-calibrated cameras in different years or decades. Our method allows us to create the facade textures for 4D models of these urban scenes, and this makes it possible to visualize such a scene from any viewpoint, date and time. Moreover, our tools also allow us to give time-of-year estimates of undated photographs and to perform color matching between dissimilar photos.

There are a few limitations to our approach that we plan to explore in the future. First, we have largely ignored the effects of sky illumination apart from the sun, and accounting for this should give improved results. Related to this, our illumination correction makes the assumption that the urban scene is directly lit by the sun, and as such does not apply to photos taken on heavily clouded days. In addition, our illumination estimation and albedo determination is not applicable to non-Lambertian surfaces, such as buildings that have glass-dominated facades. Working with scenes that contain such non-Lambertian objects is a challenging area for future research.

In conclusion, visualizing large-scale dynamic scenes from historic images is an exciting but challenging task. For a large-scale 3D scene with many occlusions, an accurate estimation of geometry is already difficult due to the accuracy of image feature matching and SFM methods. When extended to 4D, additional techniques are required to identify the changing structures and the time when each image is taken. We have shown above that as the scene changes through time, the photometry of the scene is even harder to estimate due to the unknown camera photometry model and illumination conditions. We have presented novel illumination estimation and facade-based color matching techniques that form a valuable contribution toward solving the difficult problem of 4D scene visualization.

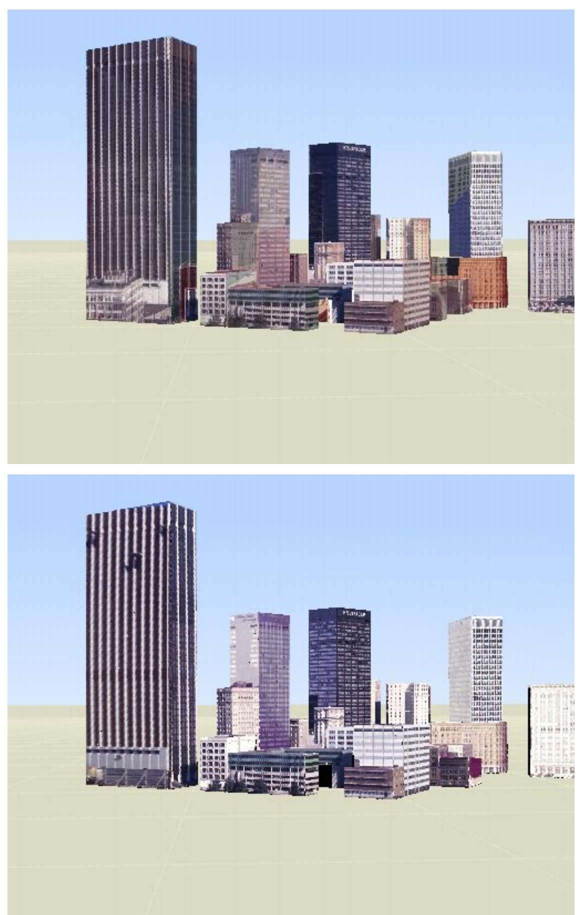


Figure 12. Top: blindly combined textures. Bottom: shadow-free, uniform-shaded, color matched object albedos.

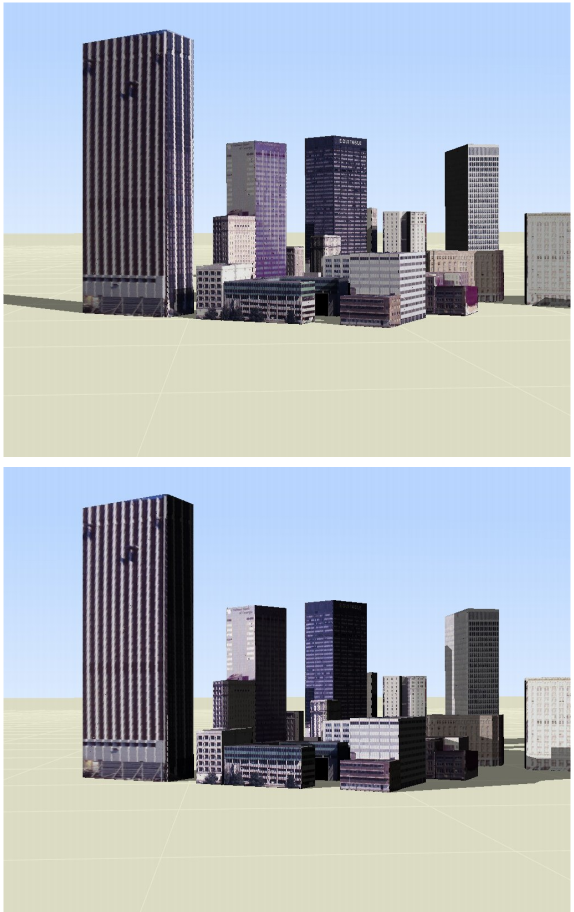


Figure 13. Virtual views at two different times of day (Nov. 6). Top: 10 am. Bottom: 5 pm. Note the changes in illumination and shadowing due to sun position.

References

[1] A. Balan, M. Black, H. Haussecker, and L. Sigal. Shining a light on human pose: On shadows, shading and the estimation of pose and shape. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1–8, 2007.

[2] N. Bannai, A. Agathos, and R. Fisher. Fusing multiple color images for texturing models. In *3D Data Processing Visualization and Transmission (3DPVT)*, pages 558–565, 2004.

[3] C.Zitnick, S.B.Kang, M.Uyttendaele, S.Winder, and R.Szeliski. High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics*, 23(3):600–608, 2004.

[4] P. Debevec, C. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *SIGGRAPH*, 30:11–20, 1996.

[5] N. Jacobs, S.Satkin, N. Roman, R. Speyer, and R. Pless. Geolocating static cameras. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1–6, 2007.

[6] J.P.Pons, R.Keriven, and O.Faugeras. Modelling dynamic scenes by registering multi-view image sequences. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 822–827, 2005.

[7] R. Kawakami, K. Ikeuchi, and R. Tan. Consistent surface color for texturing large objects in outdoor scenes. In *Intl. Conf. on Computer Vision (ICCV)*, volume 2, 2005.

[8] J. Kopf, B. Neubert, B. Chen, M. Cohen, D. Cohen-Or, O. Deussen, M. Uyttendaele, and D. Lischinski. Deep photo: Model-based photograph enhancement and viewing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2008)*, to appear(to appear):to appear, 2008.

[9] J. Lalonde and A. Efros. Using color compatibility for assessing image realism. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1–8, 2007.

[10] Z. Liu, C. . Zhang, and Z. Zhang. Learning-based perceptual image quality improvement for video conferencing. In *IEEE Intl. Conf. on Multimedia and Expo(ICME)*, pages 1035–1038, 2007.

[11] I. Reda and A. Andreas. Solar position algorithm for solar radiation application. Technical Report NREL/TP-560-34302, National Renewable Energy Laboratory, 2003.

[12] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE COMPUTER GRAPHICS AND APPLICATIONS*, pages 34–41, 2001.

[13] G. Schindler, F. Dellaert, and S. Kang. Inferring temporal order of images from 3D structure. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[14] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *SIGGRAPH*, pages 835–846, 2006.

[15] Y. Tai, J. Jia, and C. Tang. Local color transfer via probabilistic segmentation by expectation-maximization. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 747–754, 2005.

[16] A. Troccoli and P. Allen. Building illumination coherent 3d models of large-scale outdoor scenes. *Intl. J. of Computer Vision*, 78(2-3):261–280, 2008.

[17] S. Vedula, S. Baker, and T. Kanade. Image-based spatio-temporal modeling and view interpolation of dynamic events. *ACM Transactions on Graphics*, 24(2):240 – 261, April 2005.

[18] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A.Adams, M.Horowitz, and M.Levoy. High performance imaging using large camera arrays. *ACM Transactions on Graphics*, 24(3):765 – 776, 2005.

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