

What Does the Sky Tell Us about the Camera?

Jean-François Lalonde, Srinivasa G. Narasimhan, and Alexei A. Efros

School of Computer Science, Carnegie Mellon University

<http://graphics.cs.cmu.edu/projects/sky>

Abstract. As the main observed illuminant outdoors, the sky is a rich source of information about the scene. However, it is yet to be fully explored in computer vision because its appearance depends on the sun position, weather conditions, photometric and geometric parameters of the camera, and the location of capture. In this paper, we propose the use of a physically-based sky model to analyze the information available within the visible portion of the sky, observed over time. By fitting this model to an image sequence, we show how to extract camera parameters such as the focal length, and the zenith and azimuth angles. In short, the sky serves as a geometric calibration target. Once the camera parameters are recovered, we show how to use the same model in two applications: 1) segmentation of the sky and cloud layers, and 2) data-driven sky matching across different image sequences based on a novel similarity measure defined on sky parameters. This measure, combined with a rich appearance database, allows us to model a wide range of sky conditions.

1 Introduction

When presented with an outdoor photograph (such as images on Fig. 1), an average person is able to infer a good deal of information just by looking at the sky. Is it morning or afternoon? Do I need to wear a sunhat? Is it likely to rain? A professional, such as a sailor or a pilot, might be able to tell even more: time of day, temperature, wind conditions, likelihood of a storm developing, etc. As the main observed illuminant in an outdoor image, the sky is a rich source of information about the scene. However it is yet to be fully explored in computer vision. The main obstacle is that the problem is woefully under-constrained. The appearance of the sky depends on a host of factors such as the position of the sun, weather conditions, photometric and geometric parameters of the camera, and location and direction of observation. Unfortunately, most of these factors remain unobserved in a single photograph; the sun is rarely visible in the picture, the camera parameters and location are usually unknown, and worse yet, only a small fraction of the full hemisphere of sky is actually seen.

However, if we were to observe the same small portion of the sky *over time*, we would see the changes in sky appearance due to the sun and weather that are not present within a single image. In short, this is exactly the type of problem that might benefit from observing a time-lapse image sequence. Such a sequence is typically acquired by a static camera looking at the same scene over a period of



Fig. 1. The sky appearance is a rich source of information about the scene illumination

time. When the scene is mostly static, the resulting sequence of images contains a wealth of information that has been exploited in several different ways, the most commonly known being background subtraction, but also shadow detection and removal [1], video factorization and compression [2], radiometric calibration [3], camera geo-location [4], temporal variation analysis [5] and color constancy [6]. The main contribution of this paper is to show what information about the camera is available in the visible portion of the sky in a time-lapse image sequence, and how to extract this information to calibrate the camera.

The sky appearance has long been studied by physicists. One of the most popular physically-based sky model was introduced by Perez et al [7]. This model has been used in graphics for relighting [8] and rendering [9]. Surprisingly however, very little work has been done on extracting information from the visible sky. One notable exception is the work of Jacobs et al [10] where they use the sky to infer the camera azimuth by using a correlation-based approach. In our work, we address a broader question: what does the sky tell us about the camera? We show how we can recover the viewing geometry using an optimization-based approach. Specifically, we estimate the camera focal length, its zenith angle (with respect to vertical), and its azimuth angle (with respect to North). We will assume that a static camera is observing the same scene over time, with no roll angle (i.e. the horizon line is parallel to the image horizontal axis). Its location (GPS coordinates) and the times of image acquisition are also known. We also assume that the sky region has been segmented, either manually or automatically [5].

Once the camera parameters are recovered, we then show how we can use our sky model in two applications. First, we present a novel sky-cloud segmentation algorithm that identifies cloud regions within an image. Second, we show how we can use the resulting sky-cloud segmentation in order to find matching skies across different cameras. To do so, we introduce a novel bi-layered sky model which captures both the physically-based sky parameters and cloud appearance, and determine a similarity measure between two images. This distance can then be used for finding images with similar skies, even if they are captured by different cameras at different locations. We show qualitative cloud segmentation and sky matching results that demonstrate the usefulness of our approach.

In order to thoroughly test our algorithms, we require a set of time-lapse image sequences which exhibit a wide range of skies and cameras. For this, we use the AMOS (Archive of Many Outdoor Scenes) database [5], which contains image sequences taken by static webcams over more than a year.

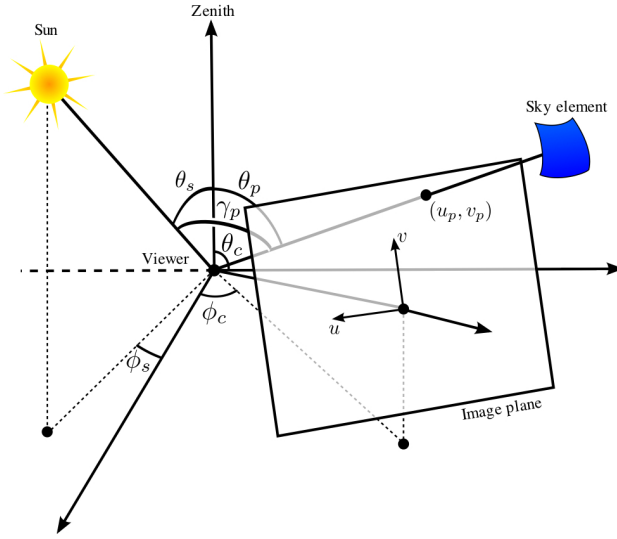


Fig. 2. Geometry of the problem, when a camera is viewing a sky element (blue patch in the upper-right). The sky element is imaged at pixel (u_p, v_p) in the image, and the camera is rotated by angles (θ_c, ϕ_c) . The camera focal length f_c , not shown here, is the distance between the origin (center of projection), and the image center. The sun direction is given by (θ_s, ϕ_s) , and the angle between the sun and the sky element is γ_p . Here (u_p, v_p) are known because the sky is segmented.

2 Physically-Based Model of the Sky

First, we introduce the physically-based model of the sky that lies at the foundation of our approach. We will first present the model in its general form, then in a useful simplified form, and finally demonstrate how it can be written as a function of camera parameters. We will consider clear skies only, and address the more complicated case of clouds at a later point in the paper.

2.1 All-Weather Perez Sky Model

The Perez sky model [7] describes the luminance of any arbitrary sky element as a function of its elevation, and its relative orientation with respect to the sun. It is a generalization of the CIE standard clear sky formula [11], and it has been found to be more accurate for a wider range of atmospheric conditions [12]. Consider the illustration in Fig. 2. The *relative* luminance l_p of a sky element is a function of its zenith angle θ_p and the angle γ_p with the sun:

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

where the 5 constants (a, b, c, d, e) specify the current atmospheric conditions. As suggested in [9], those constants can also be expressed as a linear function of a

single parameter, the turbidity t . Intuitively, the turbidity encodes the amount of scattering in the atmosphere, so the lower t , the clearer the sky. For clear skies, the constants take on the following values: $a = -1$, $b = -0.32$, $c = 10$, $d = -3$, $e = 0.45$, which corresponds approximately to $t = 2.17$.

The model expresses the *absolute* luminance L_p of a sky element as a function of another arbitrary reference sky element. For instance, if the zenith luminance L_z is known, then

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

where θ_s is the zenith angle of the sun.

2.2 Clear-Weather Azimuth-Independent Sky Model

By running synthetic experiments, we were able to determine that the influence of the second factor in (1) becomes negligible when the sun is more than 100° away from a particular sky element. In this case, the sky appearance can be modeled by using only the first term from (1):

$$l'_p = f'(\theta_p) = 1 + a \exp(b / \cos \theta_p) . \tag{3}$$

This equation effectively models the sky *gradient*, which varies from light to dark from horizon to zenith on a clear day. L'_p is obtained in a similar fashion as in (2):

$$L'_p = L_z \frac{f'(\theta_p)}{f'(0)} . \tag{4}$$

2.3 Expressing the Sky Model as a Function of Camera Parameters

Now suppose a camera is looking at the sky, as in Fig. 2. We can express the general (1) and azimuth-independent (3) models as functions of camera parameters. Let us start with the simpler azimuth-independent model.

If we assume that the camera zenith angle θ_c is independent of its azimuth angle ϕ_c , then $\theta_p \approx \theta_c - \arctan\left(\frac{v_p}{f}\right)$. This can be substituted into (3):

$$l'_p = g'(v_p, \theta_c, f_c) = 1 + a \exp\left(\frac{b}{\cos(\theta_c - \arctan(v_p/f_c))}\right) , \tag{5}$$

where, v_p is the v -coordinate of the sky element in the image, and f_c is the camera focal length.

In the general sky model case, deriving the equation involves expressing γ_p as a function of camera parameters:

$$\gamma_p = \arccos(\cos \theta_s \cos \theta_p + \sin \theta_s \sin \theta_p \cos \Delta\phi_p) , \tag{6}$$

where $\Delta\phi_p \approx \phi_c - \phi_s - \arctan\left(\frac{u_p}{f}\right)$, and u_p is the sky element u -coordinate in the image. We substitute (6) into (1) to obtain the final equation. For succinctness, we omit writing it in its entirety, but do present its general form:

$$l_p = g(u_p, v_p, \theta_c, \phi_c, f_c, \theta_s, \phi_s) , \tag{7}$$

where θ_c, ϕ_c (θ_s, ϕ_s) are the camera (sun) zenith and azimuth angles.

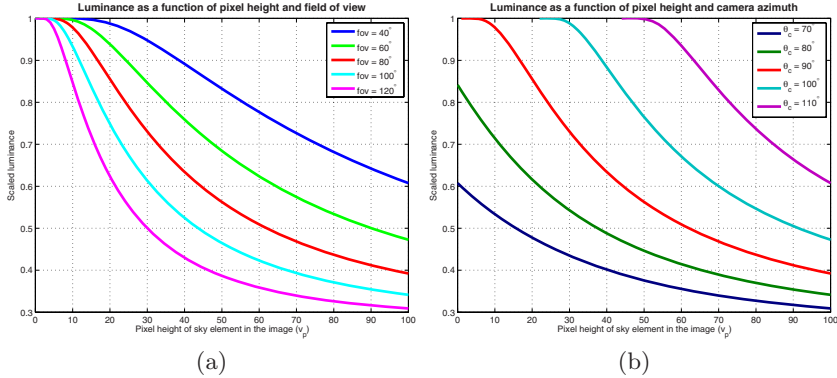


Fig. 3. Luminance profiles predicted by the azimuth-independent model (5). For clear skies, intensity diminishes as pixel height (x -axis) increases. (a) The camera zenith angle is kept constant at $\theta_c = 90^\circ$, while the field of view is varied. (b) The field of view is kept constant at 80° , while the camera zenith angle is varied. Both parameters have a strong influence on the shape and offset of the predicted sky gradient.

Before we present how we use the models presented above, recall that we are dealing with *ratios* of sky luminances, and that a reference element is needed. Earlier, we used the zenith luminance L_z as a reference in (2) and (4), which unfortunately is not always visible in images. Instead, we can treat this as an additional unknown in the equations. Since the denominators in (2) and (4) do not depend on camera parameters, we can combine them with L_z into a single unknown scale factor k .

3 Using the Clear Sky as a Calibration Target

In the previous section, we presented a physically-based model of the clear sky that can be expressed as a function of camera parameters. Now if we are given a set of images taken from a static camera, can we use the clear sky as a calibration target and *recover* the camera parameters, from the sky appearance only?

3.1 Recovering Focal Length and Zenith Angle

Let us first consider the simple azimuth-independent model (5). If we plot the predicted luminance profile for different focal lengths as in Fig. 3-(a) (or, equivalently, for different fields of view), we can see that there is a strong dependence between the focal length f_c and the shape of the luminance gradient. Similarly, the camera azimuth θ_c dictates the vertical offset, as in Fig. 3-(b). From this intuition, we devise a method of recovering the focal length and zenith angle of a camera from a set of images where the sun is far away from its field of view (i.e. at least 100° away). Suppose we are given a set \mathcal{I} of such images, in which the sky is visible at pixels in set \mathcal{P} , also given. We seek to find the camera parameters (θ_c, f_c) that minimize

$$\min_{\theta_c, f_c, k^{(i)}} \sum_{i \in \mathcal{I}} \sum_{p \in \mathcal{P}} \left(y_p^{(i)} - k^{(i)} g'(v_p, \theta_c, f_c) \right)^2, \quad (8)$$

where $y_p^{(i)}$ is the observed intensity of pixel p in image i , and $k^{(i)}$ are unknown scale factors (Sect. 2.3), one per image. This non-linear least-squares minimization can be solved iteratively using standard optimization techniques such as Levenberg-Marquadt, or `fminsearch` in MATLAB. f_c is initialized to a value corresponding to a 35° field of view, and θ_c is set such that the horizon line is aligned with the lowest visible sky pixel. All $k^{(i)}$'s are initialized to 1.

3.2 Recovering Azimuth Angle

From the azimuth-independent model (5) and images where the sun is far from the camera field of view, we were able to estimate the camera focal length f_c and its zenith angle θ_c . Now if we consider the general model (7) that depends on the sun position, we can also estimate the camera azimuth angle using the same framework as before.

Suppose we are given a set of images \mathcal{J} where the sky is clear, but where the sun is now closer to the camera field of view. Similarly to (8), we seek to find the camera azimuth angle which minimizes

$$\min_{\phi_c, k^{(j)}} \sum_{j \in \mathcal{J}} \sum_{p \in \mathcal{P}} \left(y_p^{(j)} - k^{(j)} g(u_p, v_p, \theta_c, \phi_c, f_c, \theta_s, \phi_s) \right)^2. \quad (9)$$

We already know the values of f_c and θ_c , so we do not need to optimize over them. Additionally, if the GPS coordinates of the camera and the time of capture of each image are known, the sun zenith and azimuth (θ_s, ϕ_s) can be computed using [13]. Therefore, the only unknowns are $k^{(j)}$ (one per image), and ϕ_c . Since this equation is highly non-linear, we have found that initializing ϕ_c to several values over the $[-\pi, \pi]$ interval and keeping the result that minimizes (9) works the best.

4 Evaluation of Camera Parameters Estimation

In order to thoroughly evaluate our model, we have performed extensive tests on synthetic data generated under a very wide range of operating conditions. We also evaluated our model on real image sequences to demonstrate its usefulness in practice.

4.1 Synthetic Data

We tested our model and fitting technique on a very diverse set of scenarios using data synthetically generated by using the original Perez sky model in (1). During these experiments, the following parameters were varied: the camera focal length f_c , the camera zenith and azimuth angles (θ_c, ϕ_c), the number of

Table 1. Camera calibration from the sky on 3 real image sequences taken from the AMOS database [5]. Error in focal length, zenith and azimuth angle estimation is shown for each sequence. The error is computed with respect to values obtained by using the sun position to estimate the same parameters [14].

Sequence name	Focal length error (%)	Zenith angle error ($^{\circ}$)	Azimuth angle error ($^{\circ}$)
257	1.1	< 0.1	2.6
414	3.1	< 0.1	2
466	2.5	< 0.1	4.5

input images used in the optimization, the number of visible sky pixels, and the camera latitude (which effects the maximum sun height). In all our experiments, 1000 pixels are randomly selected from each input image, and each experiment is repeated for 15 random selections.

The focal length can be recovered with at most 4% error even in challenging conditions: 30% visibility, over a wide range of field of view ($[13^{\circ}, 93^{\circ}]$ interval), zenith angles ($[45^{\circ}, 135^{\circ}]$), azimuth angles ($[-180^{\circ}, 180^{\circ}]$), and sun positions (entire hemisphere). We note a degradation in performance at wider fields of view ($> 100^{\circ}$), because the assumption of independent zenith and azimuth angles starts to break down (Sect. 2.3). Less than 0.1° error for both zenith and azimuth angles is obtained in similar operating conditions.

4.2 Real Data

Although experiments on synthetic data are important, real image sequences present additional challenges, such as non-linear camera response functions, non-gaussian noise, slight variations in atmospheric conditions, etc. We now evaluate our method on real image sequences and show that our approach is robust to these noise sources and can be used in practice.

First, the camera response function may be non-linear, so we need to radiometrically calibrate the camera. Although newer techniques [3] might be more suitable for image sequences, we rely on [15] which estimates the inverse response function by using color edges gathered from a single image. For additional robustness, we detect edges across several frames. Recall that the optimization procedures in (8) and (9) requires clear sky image sets \mathcal{I} and \mathcal{J} , where the sun is far or close to the camera respectively. We approximate (5) by a vertical quadratic in image space, and automatically build set \mathcal{I} by keeping images with low residual fitting error. Similarly, set \mathcal{J} is populated by finding images with a good fit to horizontal quadratic. It is important that the effect of the moving sun be visible in the selected images \mathcal{J} .

We present results from applying our algorithm on three image sequences taken from the AMOS database [5]. Since ground truth is not available on those sequences, we compare our results with those obtained with the method described in [14], which uses hand-labelled sun positions to obtain high-accuracy estimates. Numerical results are presented in Table 1, and Fig. 4 shows a

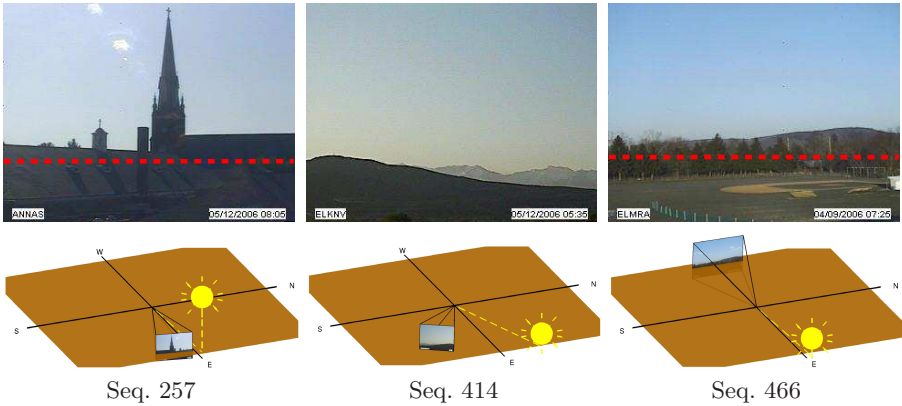


Fig. 4. Illustration of estimated camera parameters. *First row:* Example image for the three sequences in Table 1. The horizon line is drawn in red. Note that the horizon line in sequence 414 is found to be just below the image. *Second row:* Graphical illustration of all three estimated parameters: focal length, zenith and azimuth angles. The sun is drawn at the position corresponding to the image in the first row.

visualization of the recovered camera parameters. The results are consistent with image data: for instance, sun flares are visible in the first image (Seq. 257), which indicate that the sun must be above the camera, slightly to its left. This matches the visualization below the image.

5 Application: Separation of Sky and Cloud Layers

Now that we have recovered camera parameters, we demonstrate how to use the same physically-based model for two applications. Until now, we have only dealt with clear skies, but alas, this is not always true! In this section, we present a novel cloud segmentation algorithm, which will later be used for sky matching.

Clouds exhibit a wide range of textures, colors, shapes, and even transparencies. Segmenting the clouds from the sky cannot be achieved with simple heuristics such as color-based thresholding as they are easily confounded by the variation in their appearances. On the other hand, our physically-based model predicts the sky appearance, so any pixel that differs from it is an outlier and is likely to correspond to a cloud. Using this intuition, we now consider two ways of fitting our model to skies that may contain clouds. Note that we perform all processing in the xyY color space as recommended in [9].

5.1 Least-Squares Fitting

The first idea is to follow a similar approach as we did previously and fit the model (5) in a non-linear least-squares fashion, by adjusting the coefficients (a, b, c, d, e) and the unknown scale factor k independently in each color channel, and treating the outliers as clouds. To reduce the number of variables,

we follow [9] and express the five weather coefficients as a linear function of a single value, the turbidity t . Strictly speaking, this means minimizing over $\mathbf{x} = [t \ k^{(1)} \ k^{(2)} \ k^{(3)}]$:

$$\min_{\mathbf{x}} \sum_{i=1}^3 \sum_{p \in \mathcal{P}} \left(y_p^{(i)} - k^{(i)} g(u_p, v_p, \theta_s, \phi_s, \tau^{(i)}(t)) \right)^2, \quad (10)$$

where i indexes the color channel. Here the camera parameters are fixed, so we omit them for clarity. The vector $\tau^{(i)}(t)$ represents the coefficients (a, \dots, e) obtained by multiplying the turbidity t with the linear transformation $M^{(i)}$: $\tau^{(i)}(t) = M^{(i)} [t \ 1]^T$. The entries of $M^{(i)}$ for the xyY space are given in the appendix in [9]. The $k^{(i)}$ are initialized to 1, and t to 2 (low turbidity).

Unfortunately, solving this simplified minimization problem does not yield satisfying results because the L2-norm is not robust to outliers, so even a small amount of clouds will bias the results.

5.2 Regularized Fitting

In order to increase robustness to outliers, we compute a data-driven prior model of clear skies \mathbf{x}_c , which we use to add 2 terms to (10): 1) we assign more weight to pixels we believe are part of the sky; and 2) we penalize parameters that differ from the prior in an L2 sense. Equation (10) becomes

$$\min_{\mathbf{x}} \sum_{i=1}^3 \sum_{p \in \mathcal{P}} w_p \left(y_p^{(i)} - k^{(i)} g(u_p, v_p, \theta_s, \phi_s, \tau^{(i)}(t)) \right)^2 + \beta \|\mathbf{x} - \mathbf{x}_c\|^2, \quad (11)$$

where, $w_p \in [0, 1]$ is a weight given to each pixel, and $\beta = 0.05$ controls the importance of the prior term in the optimization. We initialize \mathbf{x} to the prior \mathbf{x}_c .

Let us now look at how \mathbf{x}_c is obtained. We make the following observation: clear skies should have low turbidities, and they should be smooth (i.e. no patchy clouds). Using this insight, if minimizing (10) on a given image yields low residual error and turbidity, then the sky must be clear. We compute a database of clear skies by keeping all images with turbidity less than a threshold (we use 2.5), and keep the best 200 images, sorted by residual error. Given an image, we compute \mathbf{x}_c by taking the mean over the K nearest neighbors in the clear sky database, using the angular deviation between sun positions as distance measure (we use $K = 2$). This allows us to obtain a prior model of what the clear sky should look like at the current sun position. Note that we simply could have used the values for (a, \dots, e) from Sect. 2 and fit only the scale factors $k^{(i)}$, but this tends to over-constrain, so we fit t as well to remain as faithful to the data as possible.

To obtain the weights w_p in (11), the color distance λ between each pixel and the prior model is computed and mapped to the $[0, 1]$ interval with an inverse exponential: $w_p = \exp\{-\lambda^2/\sigma^2\}$ (we use $\sigma^2 = 0.01$ throughout this paper). After the optimization is over, we re-estimate w_p based on the new parameters \mathbf{x} , and repeat the process until convergence, or until a maximum number of iterations

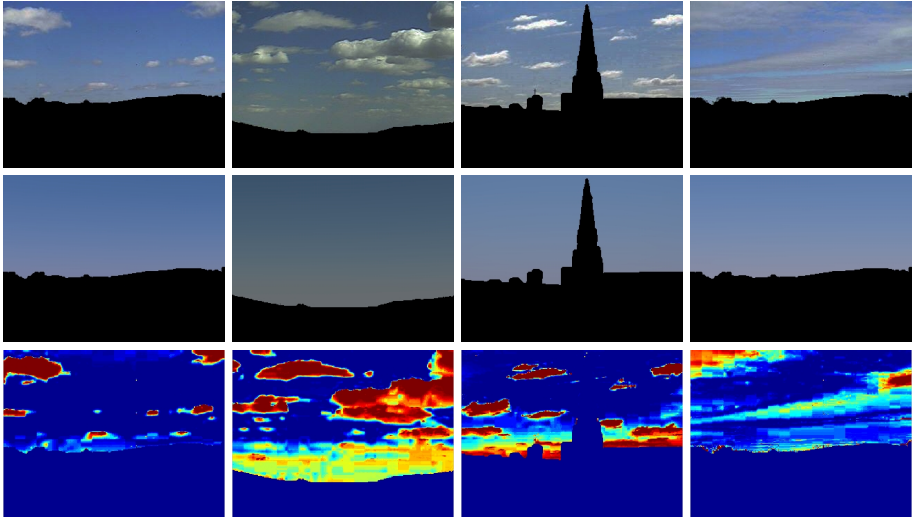


Fig. 5. Sky-cloud separation example results. *First row:* input images (radiometrically corrected). *Second row:* sky layer. *Third row:* cloud segmentation. The clouds are color-coded by weight: 0 (blue) to 1 (red). Our fitting algorithm is able to faithfully extract the two layers in all these cases.

is reached. The process typically converges in 3 iterations, and the final value for w_p is used as the cloud segmentation. Cloud coverage is then computed as $\frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} w_p$.

5.3 Segmentation Results

Figure 5 shows typical results of cloud layers extracted using our approach. Note that unweighted least-squares (10) fails on all these examples because the clouds occupy a large portion of the sky, and the optimization tries to fit them as much as possible, since the quadratic loss function is not robust to outliers. A robust loss function behaves poorly because it treats the sky pixels as outliers in the case of highly-covered skies, such as the examples shown in the first two columns of Fig. 6. Our approach injects domain knowledge into the optimization by using a data-driven sky prior, forcing it to fit the visible sky. Unfortunately, since we do not model sunlight, the estimation does not converge to a correct segmentation when the sun is very close to the camera, as illustrated in the last two columns of Fig. 6.

6 Application: Matching Skies across Image Sequences

After obtaining a sky-cloud segmentation, we consider the problem of finding matching skies between images taken by different cameras. Clearly, appearance-based matching algorithms such as cross-correlation would not work if the cameras have different parameters. Instead, we use our sky model along with cloud

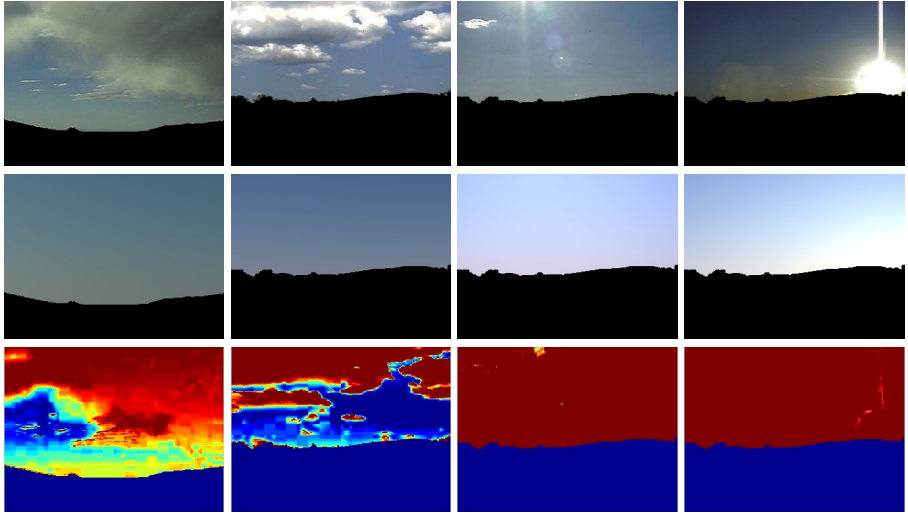


Fig. 6. More challenging cases for the sky-cloud separation, and failure cases. *First row:* input images (radiometrically corrected). *Second row:* sky layer. *Third row:* cloud layer. The clouds are color-coded by weight: 0 (blue) to 1 (red). Even though the sky is more than 50% occluded in the input images, our algorithm is able to recover a good estimate of both layers. The last two columns illustrate a failure case: the sun (either when very close or in the camera field of view) significantly alters the appearance of the pixels such that they are labeled as clouds.

statistics in order to find skies that have similar properties. We first present our novel bi-layered representation for sky and clouds, which we then use to define a similarity measure between two images. We then present qualitative matching results on real image sequences.

6.1 Bi-layered Representation for Sky and Clouds

Because clouds can appear so differently due to weather conditions, a generative model such as the one we are using for the sky is likely to have a large number of parameters, and thus be difficult to fit to image data. Instead, we propose a hybrid model: our physically-based sky model parameterized by the turbidity t for the sky appearance, and a non-parametric representation for the clouds.

Taking inspiration from Lalonde et al [16], we represent the cloud layer by a joint color histogram in the xyY space over all pixels which belong to the cloud regions. While they have had success with color histograms only, we have found this to be insufficient on our richer dataset, so we also augment the representation with a texton histogram computed over the same regions. A 1000-word texton dictionary is built from a set of skies taken from training images different than the ones used for testing. In our implementation, we choose 21^3 bins for the color histograms.

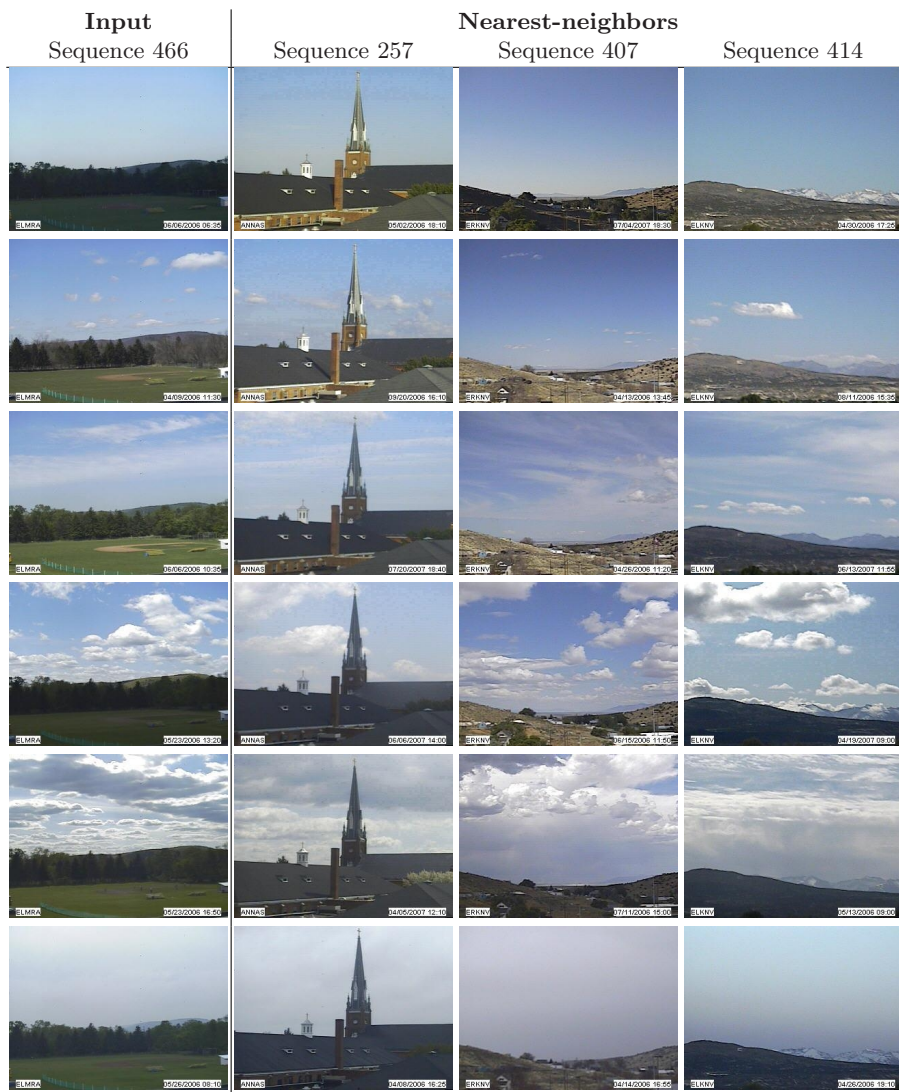


Fig. 7. Sky matching results across different cameras. The left-most column shows several images taken from different days of sequence 466 in the AMOS database. The three other columns are the nearest-neighbor matches in sequences 257, 407 and 414 respectively, obtained using our distance measure. Sky conditions are well-matched, even though cameras have different parameters.

Once this layered sky representation is computed, similar images can be retrieved by comparing their turbidities and cloud statistics (we use χ^2 distance for histogram comparison). A combined distance is obtained by taking the sum of cloud and turbidity distance, with the relative importance between the two determined by the cloud coverage.

6.2 Qualitative Evaluation

The above algorithm was tested on four sequences from the AMOS database. Since we do not have ground truth to evaluate sky matching performance, we provide qualitative results in Fig. 7. Observe that sky conditions are matched correctly, even though cameras have different horizons, focal lengths, and camera response functions. A wide range of sky conditions can be matched successfully, including clear, various amounts of clouds, and overcast conditions. We provide additional segmentation and matching results on our project website.

7 Summary

In this paper, we explore the following question: what information about the camera is available in the visible sky? We show that, even if a very small portion of the hemisphere is visible, we can reliably estimate three important camera parameters by observing the sky *over time*. We do so by expressing a well-known physically-based sky model in terms of the camera parameters, and by fitting it to clear sky images using standard minimization techniques. We then demonstrate the accuracy of our approach on synthetic and real data. Once the camera parameters are estimated, we show how we can use the same model to segment out clouds from sky and build a novel bi-layered representation, which can then be used to find similar skies across different cameras.

We plan to use the proposed sky illumination model to see how it can help us predict the illumination of the scene. We expect that no parametric model will be able to capture this information well enough, so data-driven methods will become even more important.

Acknowledgements

This research is supported in parts by an ONR grant N00014-08-1-0330 and NSF grants IIS-0643628, CCF-0541307 and CCF-0541230. A. Efros is grateful to the WILLOW team at ENS Paris for their hospitality.

References

1. Weiss, Y.: Deriving intrinsic images from image sequences. In: IEEE International Conference on Computer Vision (2001)
2. Sunkavalli, K., Matusik, W., Pfister, H., Rusinkiewicz, S.: Factored time-lapse video. ACM Transactions on Graphics (SIGGRAPH 2007) 26(3) (August 2007)
3. Kim, S.J., Frahm, J.M., Pollefeys, M.: Radiometric calibration with illumination change for outdoor scene analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
4. Jacobs, N., Satkin, S., Roman, N., Speyer, R., Pless, R.: Geolocating static cameras. In: IEEE International Conference on Computer Vision (2007)
5. Jacobs, N., Roman, N., Pless, R.: Consistent temporal variations in many outdoor scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)

6. Sunkavalli, K., Romeiro, F., Matusik, W., Zickler, T., Pfister, H.: What do color changes reveal about an outdoor scene? In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
7. Perez, R., Seals, R., Michalsky, J.: All-weather model for sky luminance distribution – preliminary configuration and validation. *Solar Energy* 50(3), 235–245 (1993)
8. Yu, Y., Malik, J.: Recovering photometric properties of architectural scenes from photographs. *Proceedings of ACM SIGGRAPH 1998* (July 1998)
9. Preetham, A.J., Shirley, P., Smits, B.: A practical analytic model for daylight. *Proceedings of ACM SIGGRAPH 1999* (August 1999)
10. Jacobs, N., Roman, N., Pless, R.: Toward fully automatic geo-location and geo-orientation of static outdoor cameras. In: *Workshop on applications of computer vision* (2008)
11. Committee, C.T.: Spatial distribution of daylight – luminance distributions of various reference skies. Technical Report CIE-110-1994, International Commission on Illumination (1994)
12. Ineichen, P., Molineaux, B., Perez, R.: Sky luminance data validation: comparison of seven models with four data banks. *Solar Energy* 52(4), 337–346 (1994)
13. Reda, I., Andreas, A.: Solar position algorithm for solar radiation applications. Technical Report NREL/TP-560-34302, National Renewable Energy Laboratory (November 2005)
14. Lalonde, J.F., Narasimhan, S.G., Efros, A.A.: Camera parameters estimation from hand-labelled sun positions in image sequences. Technical Report CMU-RI-TR-08-32, Robotics Institute. Carnegie Mellon University (July 2008)
15. Lin, S., Gu, J., Yamazaki, S., Shum, H.Y.: Radiometric calibration from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition (2004)
16. Lalonde, J.F., Hoiem, D., Efros, A.A., Rother, C., Winn, J., Criminisi, A.: Photo clip art. *ACM Transactions on Graphics (SIGGRAPH 2007)* 26(3) (August 2007)