Shadow Detection and Sun Direction in Photo Collections

Scott Wehrwein, Noah Snavely, and Kavita Bala
{swehrwein, snavely, kb}@cs.cornell.edu

http://www.cs.cornell.edu/projects/shadows

Motivation

- Appearance modeling in outdoor photo collections is challenging because many factors affect image intensity.
- Existing approaches tend to model many elements at once, using complex algorithms and unstable nonlinear optimization.
- **Shadow detection** is a simpler, more tractable problem that can nonetheless reveal a lot about the illumination in a scene.
- **Sun direction** is an important illumination property linked to capture time and applicable to more detailed lighting estimation, relighting, and other applications.

Contributions

- Introduction and analysis of the *Illumination Ratio*, a quantity that captures the relative illumination of two scene points in an image, invariant to albedo and camera exposure.
- An algorithm using the illumination ratio to estimate binary shadow labels for points in a large Internet photo collection.
- A method for using sparse shadow labels to estimate the direction of the sun.

The Illumination Ratio

Image Formation Model:

$$I_i(x) = \rho_x E_i[C_{x,i} L_d \cos(\phi_{x,i}) + L_a]$$

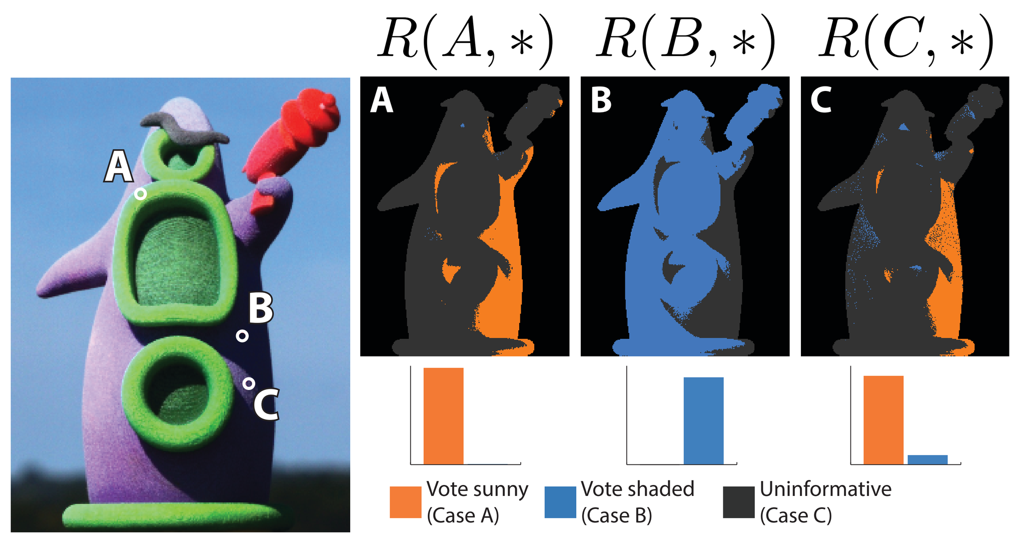
The Illumination Ratio:

$$R(x,y) = \frac{\frac{I_i(x)}{\mathcal{E}_i[I_i(x)]}}{\frac{I_i(y)}{\mathcal{E}_i[I_i(y)]}} = \frac{\frac{\textit{Intensity of x}}{\textit{Average intensity of x}}}{\frac{\textit{Intensity of y}}{\textit{Average intensity of y}}}$$

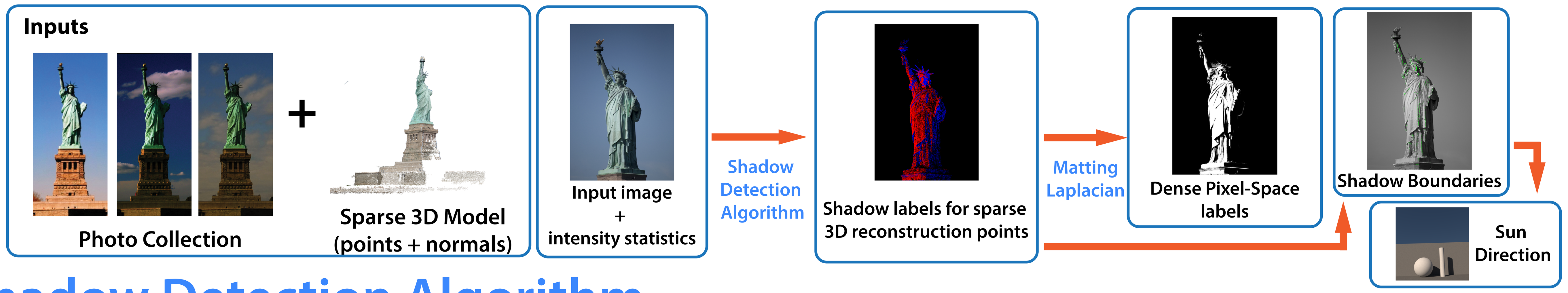
$$= \frac{\frac{\rho_x E_i L_{i,x}}{\rho_x \mathcal{E}_i[E_i L_{i,x}]}}{\frac{\rho_y E_i L_{i,y}}{\rho_y \mathcal{E}_i[E_i L_{i,y}]}} = \boxed{\frac{C_x \cos(\phi_x) + f}{C_y \cos(\phi_y) + f}} \qquad f = \frac{L_a}{L_d}$$

Key Properties:

- Invariant to albedo and exposure
- Captures relative illumination of x and y.



System Overview



Shadow Detection Algorithm

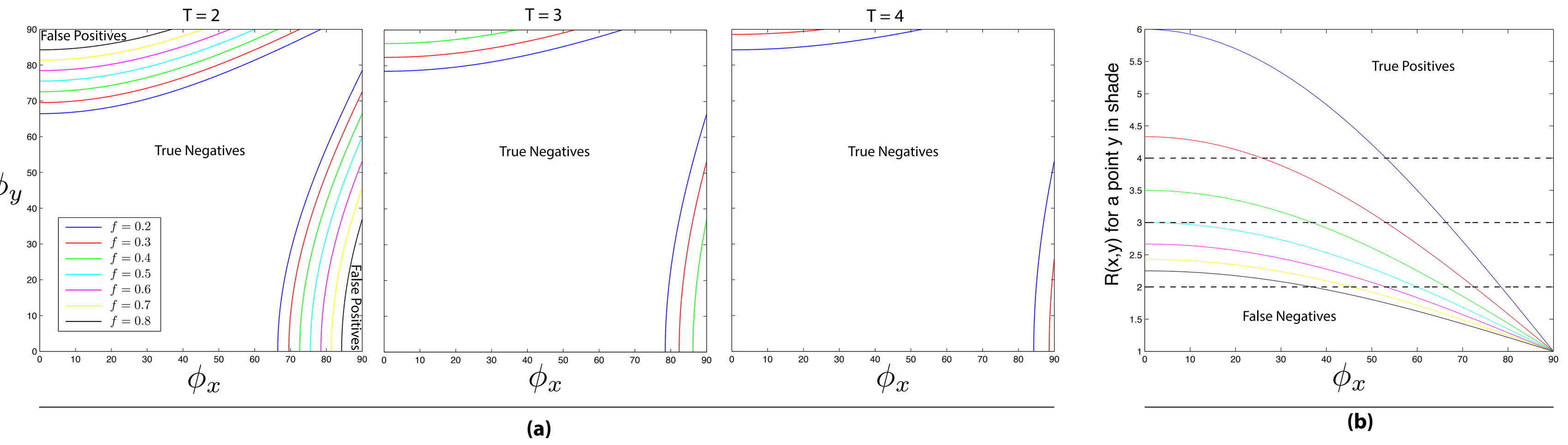
Key Idea: Compute illumination ratio among many pairs of points, and aggregate information by voting.

For each point x:

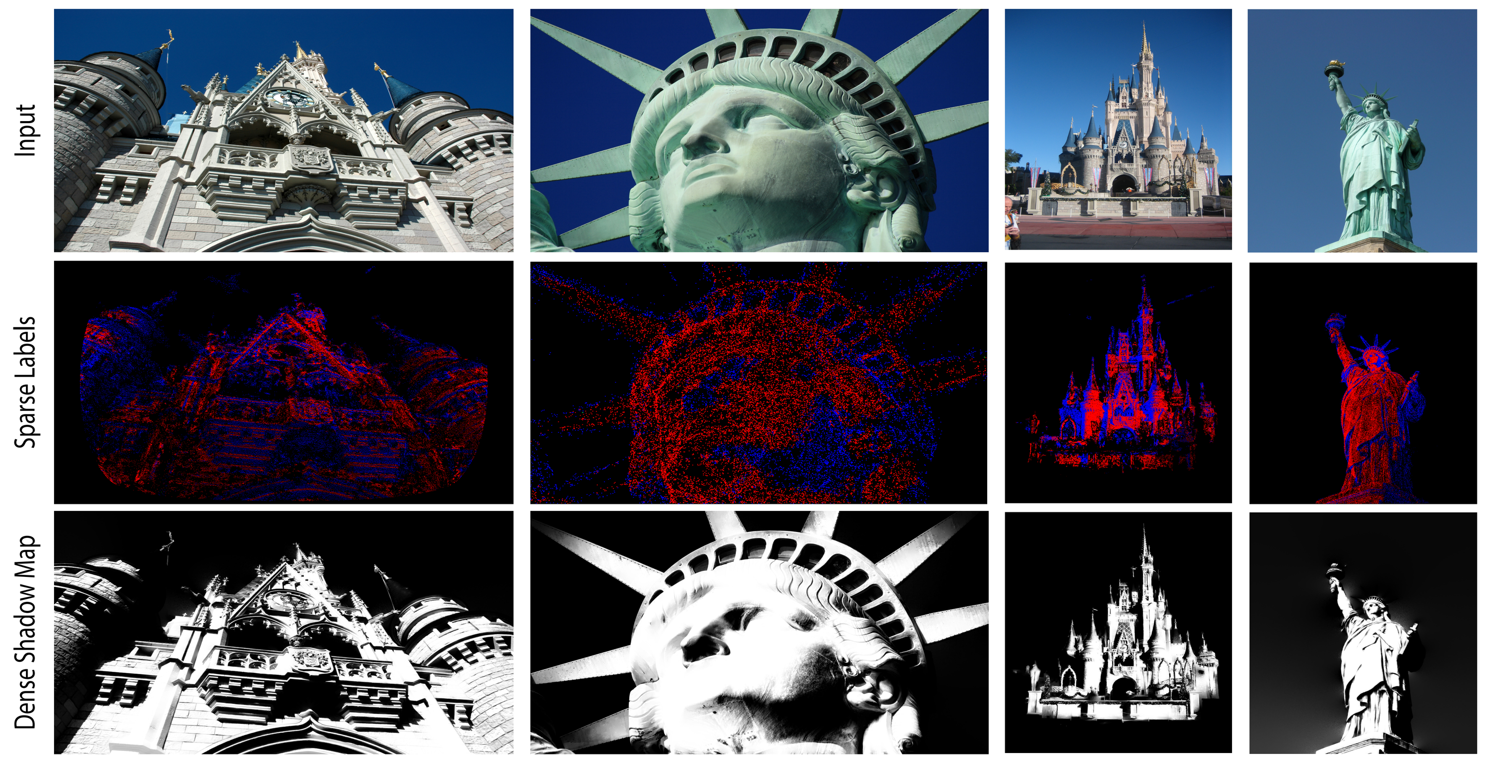
1. Pick K other points $y_1 \ldots y_K$
2. For each y_j:
   - If $R(x, y_j) > T$: cast a vote for x being sunlit.
   - If $R(x, y_j) < 1/T$: cast a vote for x being shaded.
   - Otherwise: cast no vote
3. Assign x's label according to majority vote.
4. Use the Matting Laplacian [Levin et al. 2006] to estimate dense pixel-space labels from projected 3D point labels

How do we choose T?

No ideal value, because $\cos(\phi)$ and L_a/L_d both vary and are unknown *a priori*. Because of voting, the threshold needs to be correct a majority of the time. **T=3** works empirically and is supported by our analysis based on possible values of $\cos(\phi)$ and L_a/L_d.

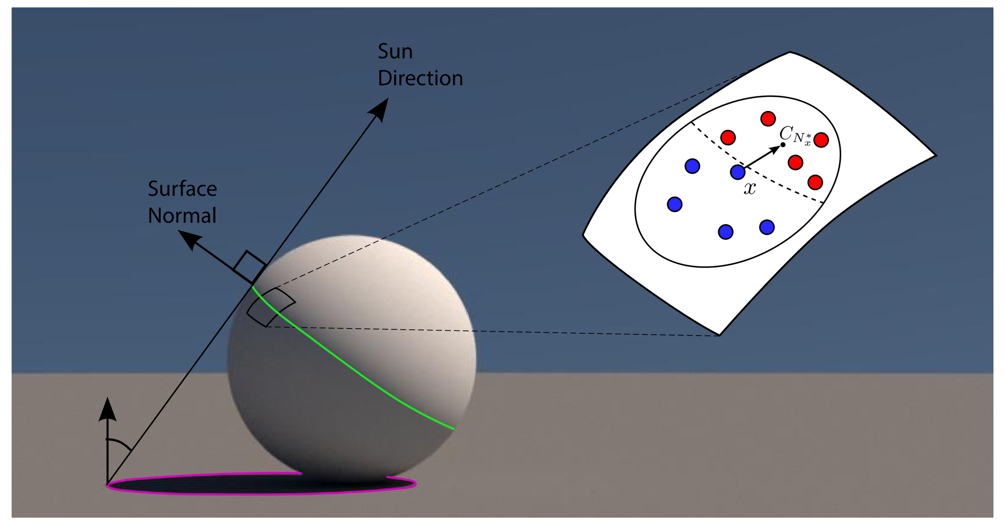


Results: Shadow Detection



Application: Sun Direction Estimation

Key Idea: Surface normals at attached shadow boundaries are orthogonal to the sun direction.



Algorithm:

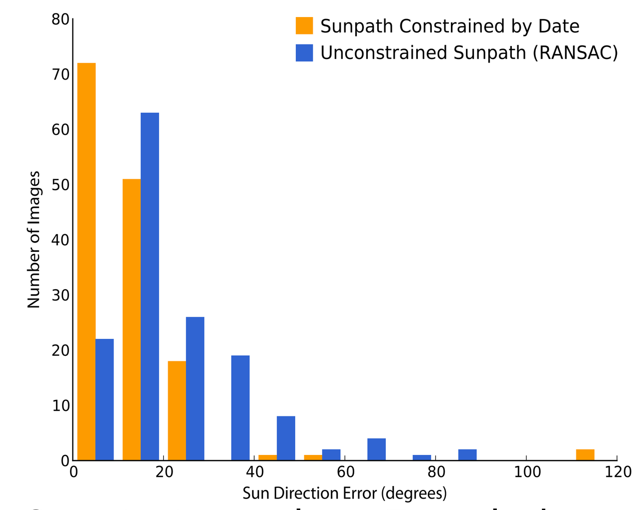
1. Find shadow boundaries: $B_i(x) = c_{sign} \frac{|N_x^*|}{|N_x|}(C_{N_x^*} - x)$
2. Use RANSAC to find consensus sun direction, discarding cast shadow boundaries and outliers.

OR

2. If scene is georegistered and date is known, test only hypotheses along the 1D sun path for that place and date.

Results: Sun Direction

Input image Shadow labels Shadow boundaries (inliers in green)



Quantitative results on Tentacle dataset