Shadow Removal for Aerial Imagery by Information Theoretic Intrinsic Image Analysis

Vivek Kwatra
kwatra@google.com

Mei Han
meihan@google.com

Shengyang Dai
sydai@google.com

Google Inc., Mountain View, CA

Abstract

We present a novel technique for shadow removal based on an information theoretic approach to intrinsic image analysis. Our key observation is that any illumination change in the scene tends to increase the entropy of observed texture intensities. Similarly, the presence of texture in the scene increases the entropy of the illumination function. Consequently, we formulate the separation of an image into texture and illumination components as minimization of entropies of each component. We employ a non-parametric kernel-based quadratic entropy formulation, and present an efficient multi-scale iterative optimization algorithm for minimization of the resulting energy functional. Our technique may be employed either fully automatically, using a proposed learning based method for automatic initialization, or alternatively with small amount of user interaction. As we demonstrate, our method is particularly suitable for aerial images, which consist of either distinctive texture patterns, e.g. building facades, or soft shadows with large diffuse regions, e.g. cloud shadows.

1. Introduction

Natural images can typically be described as the interaction of scene illumination with the geometry and reflectance of underlying objects. Separation of an image into these components has been a well studied problem, generally referred to as *intrinsic image analysis*, where each component characterizes an intrinsic property of the scene. Such decomposition is useful for many computer vision problems; *e.g.* tasks such as segmentation and recognition require illumination invariance and therefore benefit from removal of the illumination component.

A commonly observed phenomenon attributable to illumination changes is (cast) shadow formation, caused by occlusion of the light source. We present a shadow removal approach based on an information theoretic model for the underlying texture and illumination of the scene. Our key observation is that any illumination change in the scene tends to increase the diversity of observed texture intensities. An information theoretic interpretation of this effect is that the entropy of texture appearance is increased. Similarly, the presence of texture in the scene increases the entropy of the illumination function, which is otherwise mostly smooth except at shadow boundaries. Hence, shadow removal can be cast as separation of texture from illumination such that the entropy of both entities is reduced. It is important to consider *both* texture and illumination simultaneously, as minimizing one quantity alone would simply transfer the entire energy to the other quantity. The constraint on illumination entropy serves as a regularization, which imposes a robust smoothness prior on it.

Our method can be employed either fully automatically or alternatively with small amount of user interaction, where a loosely specified region of pixels completely outside the shadow may be provided. Our formulation of texture-illumination separation as entropy minimization of the two components is novel. We employ non-parametric estimates for texture and illumination densities and a kernel-based quadratic entropy measure for diversity. To minimize the resulting energy functional, we have devised an efficient multi-scale iterative optimization technique, which may be applicable to other kernel-based methods as well. We also present a technique for automatically bootstrapping our algorithm, resulting in a fully automatic system.

Our approach is particularly suitable for soft shadows, where an explicit shadow boundary may not be present, as well as for images with distinctive textures, where texture entropy can be easily analyzed. We have found these characteristics to be typical of aerial images, which often need to be processed for shadow removal before use in modern mapping systems. For example, raw long-distance shots used for *satellite views* in maps (*e.g.* Google Maps), can have soft shadows cast by clouds with diffuse penumbra regions. On the other hand, photos used for texturing building facades in 3D virtual environments (*e.g.* Google Earth),

usually have structured texture patterns, allowing for a simplification of texture entropy analysis. We have applied our technique on datasets from production mapping systems, including fairly high resolution images, to demonstrate its effectiveness. We also present sample results on natural images to show that our work is more generally applicable.

2. Related Work

For shadow detection and removal, two categories of algorithms have been widely explored in the literature: model based and learning based. Model based approaches characterize the image generation process and model the process physically [5, 12, 13]. On the other hand, with the rapid development of image feature extraction and machine learning techniques, learning based approaches are getting popular. Color and gradient information is used in [15], and shadow-variant and shadow-invariant cues from illumination, textural and odd order derivative characteristics are explored in [18] to recognize shadows in monochromatic images.

Due to the under-determined nature of the problem, researchers have studied restrictions to specific domains. In [10], ground regions in outdoor scenes are specifically targeted. In [9], textured surfaces are considered through correlations between local mean luminance and local luminance amplitude. More information may be integrated as extra cues, such as multiple images [6], time-elapse image sequences [16] or user interactions [3, 14, 17]. Several algorithms rely on explicit classification of the shadow region before removal. One issue with such approaches is that they rely on the shadows being mostly uniform with sharp boundaries [7] or narrow penumbra regions [1, 11]. These assumptions fail in the presence of soft non-uniform shadows with large penumbra regions, which can occur in the presence of multiple light sources or when the occluder is only partially opaque (*e.g.* clouds). [4] provides a survey of prior work in shadow removal from satellite/aerial imagery. Most of these techniques are either highly specialized, or require multispectral input.

Entropy minimization has been used for image filtering in [2]. They employ gradient descent for optimization, while we present a more efficient iterative least squares approach. An entropy based approach is used in [5, 7] as well. However, they optimize over a different quantity: color space projection basis. Recently, classifiers on paired regions [8] have been used to detect shadows. They compute a discrete solution followed by matting for removal, whereas we use a continuous formulation, better suited for diffuse shadows. However, their paired formulation is related to our texture entropy formulation and both can possibly be adapted to work together.

3. Information Theoretic Formulation

Let $\mathcal{I}(x, y)$ denote the observed intensity at pixel location (x, y) in the image. Then, it can be expressed as a function of the reflectance field (restricted to the 2D albedo map) $\mathcal{R}(x, y)$ and per-pixel illumination $\mathcal{L}(x, y)$ as:

$$\mathcal{I}(x, y) = \mathcal{L}(x, y) \cdot \mathcal{R}(x, y) \implies I(x, y) = L(x, y) + R(x, y)$$

where the latter equation re-expresses the first one in log domain. Treating these quantities as random variables, it is reasonable to assume that R and L are independent of each other, since they represent distinct intrinsic characteristics of the scene: the surface and the illuminant. The observed image I is therefore the sum of two independent random variables. It is well known that the probability density function (PDF) of the sum of independent random variables can be expressed as a convolution of their respective densities. Hence, the summation results in a smoothing of the component densities (as PDFs always form positive kernels). From an information theoretic standpoint, this is equivalent to an increase in the entropy of the system, as the PDFs become more spread out. Mathematically, it can be stated as $H(I) \geq H(R), H(L)$, where H denotes entropy.

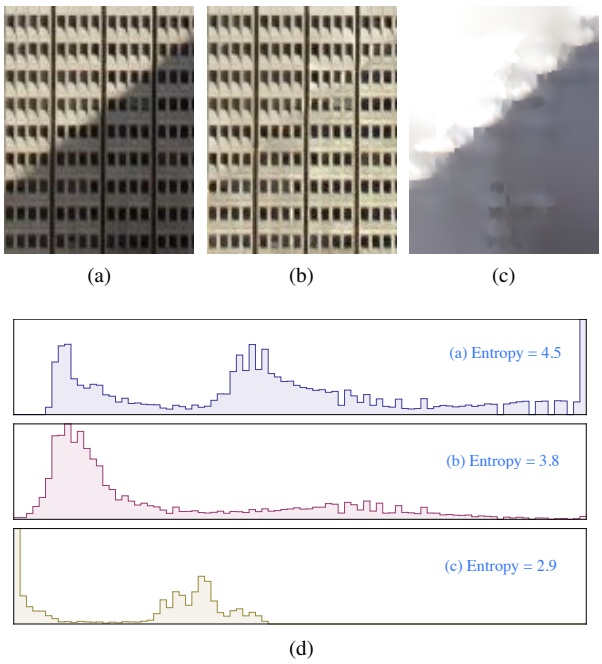


Figure 1: (a) Original shadowed image (b) Image after shadow removal (c) Illumination component (d) $-\log(intensity)$ histograms and their entropies. The original image has higher entropy than any of its components.

This concept is further demonstrated in Figure 1. It shows an image containing a shadow along with the shadow-free and illumination components (as extracted by

our system). Figure 1d shows their log domain histograms and respective entropies. Note that the histogram corresponding to the shadow image is more spread out than either the shadow-free or the illumination component, and is indeed a convolution of the components. The shadow-free image histogram is unimodal, while the illumination function is bimodal corresponding to the lit and dark regions. Their convolution results in a bimodal histogram for the shadowed image, which also has the highest entropy.

Decomposition of the image into its intrinsic components is fundamentally under-constrained and therefore some prior knowledge (or regularization) is necessary. Based on the above observations, we can impose a novel prior which requires the entropies of the components to be smaller than the original image. However, that still leaves the problem severely under-constrained. A stronger prior is to search for the hypothesis that results in the smallest possible component entropies. This formulation is consistent with the *Minimum Description Length* (MDL) principle, which postulates that the hypothesis with the most compact representation should be favored.

We note that a popular prior in the literature is to impose smoothness constraints on the illumination function. It turns out that minimization of illumination entropy is an alternate way of expressing this smoothness constraint (although it requires that the spatial location of pixels be taken into account when computing entropy). The missing piece is the regularization of the reflectance function, which is what we propose in this paper. To that end, we seek to minimize the *texture entropy* of the reflectance field.

We distinguish texture from reflectance because, unlike illumination, the reflectance function may not be (even piecewise) spatially smooth, as evident in Figure 1b. However, it may exhibit constancy of appearance over the same object/surface, which we capture using the notion of texture. We describe the texture at a given location as the appearance of the local neighborhood around that location. Texture entropy is then measured as the entropy of local neighborhoods within a region belonging to the same surface. Our objective is to obtain:

$$L^* = \arg\min_L H(L,P) + \lambda H(T,S); \quad \text{and} \quad R^* = I - L^* \tag{1}$$

where $T \equiv T(R) = T(I-L)$ denotes the texture as a function of reflectance, and λ controls the relative importance of the two entropies. To obtain spatial regularization for illumination, we couple it with the pixel location $P \equiv (x,y)$, and minimize the resulting joint entropy. For texture, we only want to minimize the entropy within the region corresponding to individual surfaces S. However, we do not have *a-priori* segmentation of the scene into distinct surfaces, making S unknown. Nevertheless, we account for it when constructing the texture entropy (see Section 4).

4. Non-parametric Entropy Estimation

In order to measure the entropy of a random variable, we first need to define its PDF. For a general scene, the forms of the PDFs for texture and illumination are unknown and therefore difficult to model parametrically. Hence, we resort to non-parametric estimation of the PDF from sampled data points. Kernel density estimation (a.k.a. Parzen windows) is a well-known technique, where the PDF of a multivariate random variable is defined as the interpolation of impulse functions situated at data points. More formally, let X be a d-dimensional random variable with an unknown PDF f. Given a sample set $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N\}$ drawn from f, the kernel density estimate of f at point $\mathbf{x}$ is:

$$\hat{f}_\Psi(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} K_\Psi(\mathbf{x}, \mathbf{a}_i) \tag{2}$$

where K is a normalized kernel function (*i.e.* integrates to one) with smoothing (or bandwidth) parameter matrix Ψ. A typical choice for K is the radially symmetric Gaussian kernel, defined as:

$$G(\mathbf{x}|\mathbf{a}_i, \Psi) = \frac{1}{(2\pi)^{d/2}|\Psi|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{a}_i)^T \Psi^{-1}(\mathbf{x}-\mathbf{a}_i)}. \tag{3}$$

Equipped with this estimate for the PDF of a random variable, we can derive an expression for its entropy. The traditional definition of entropy is based on Shannon's definition: $H_{sh}(X) = -E_f[\log f(X)]$, where E_f is the expected value over f. While H_{sh} can be estimated from $\hat{f}$, it is a difficult function to optimize. We instead use Renyi's quadratic entropy [5] definition which leads to much simplification. Renyi generalized the notion of entropy to yield a family of entropies of different orders, of which Shannon's entropy is a special case. All these entropies are equivalent w.r.t. minimization or maximization. In particular, Renyi's quadratic entropy is defined as:

$$H_{R2}(X) = -\log V(X) = -\log \int_{-\infty}^{\infty} f(\mathbf{x})^2 d\mathbf{x}\,, \quad \text{where} \tag{4}$$

$V(X) = \int_{-\infty}^{\infty} f(\mathbf{x})^2 d\mathbf{x}$ is called the information potential. For the purpose of optimization, we can drop the $\log$ and directly minimize $-V(X)$. From here on, in a slight abuse of notation, we define entropy to be the negative information potential, *i.e.*

$$H(X) = -\int_{-\infty}^{\infty} f(\mathbf{x})^2 d\mathbf{x}, \quad \text{and} \quad \hat{H}(X) = -\int_{-\infty}^{\infty} \hat{f}(\mathbf{x})^2 d\mathbf{x}. \tag{5}$$

A nice property of this definition is that for the Gaussian kernel, $\hat{H}$ can be calculated exactly, modulo the approximation involved in the estimation of the PDF. Specifically,

(5) is a convolution of $\hat{f}$ with itself. Substituting the Gaussian kernel from (3) in (2), and performing some reductions, we obtain:

$$\hat{H}(X) = -\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G(0|\mathbf{a}_i - \mathbf{a}_j, 2\Psi) \quad (6)$$

$$= -\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G(\mathbf{a}_j|\mathbf{a}_i, 2\Psi). \quad (7)$$

This is equivalent to estimating the PDF for the *difference* of *i.i.d.* random variables, by applying the kernel density method over all pairs of samples $\{\mathbf{a}_i - \mathbf{a}_j\}$, and evaluating it at $\mathbf{0}$. Yet another interpretation is that the entropy is the average of kernel functions evaluated at all pairs of samples.

4.1. Entropies for Intrinsic Images

Estimating the entropy in (7) by evaluating all pairs of samples may be too exhaustive and overly redundant for distant pairs, especially if they would not contribute towards reducing the entropy. For example, in the case of illumination, sample pairs that are spatially far enough will not play much role in the optimization, as their contribution to the entropy would remain minimal even if we made their illumination values exactly equal. Similarly, for texture, sample pairs coming from different surfaces would be useless and may be discarded. Hence, we restrict sample pairs based on their expected contribution by modifying (7) as follows:

$$\hat{H}(X) \approx -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} G(\mathbf{a}_j|\mathbf{a}_i, 2\Psi) \quad (8)$$

where the set of samples $\mathbf{a}_j$ in the inner loop are restricted to a subset of samples in the neighborhood of $\mathbf{a}_i$, denoted by $\mathcal{N}(i)$. $\mathcal{N}(i)$ is usually restricted based on the *fixed* components of X. For example, the second term in (1) measures the joint entropy of illumination and spatial location. Let $U = (L, P)$ be the corresponding random variable. Then P forms the fixed component of U. A single sample from U may be denoted as $\mathbf{u}_i = (l_i, \mathbf{p}_i)$, where l_i is the log-illumination[1] at pixel location $\mathbf{p}_i = (x_i, y_i)$. $\mathcal{N}(i)$ is then constructed by only considering pixels that lie within a spatial neighborhood around $\mathbf{p}_i$. To see why this makes sense, consider the kernel function for the sample pair $(\mathbf{u}_i, \mathbf{u}_j)$. If we employ a diagonal smoothing matrix Φ for U, we have:

$$G(\mathbf{u}_j|\mathbf{u}_i, 2\Phi) = G(l_j|l_i, 2\Phi_l)\cdot G(\mathbf{p}_j|\mathbf{p}_i, 2\Phi_\mathbf{p}) \quad (9)$$

which implies that the sample pair is always weighted by $G(\mathbf{p}_j|\mathbf{p}_i, 2\Phi_\mathbf{p})$, no matter what values we assign to (l_i, l_j). Consequently, pixels which are too far to have any significant weight may be discarded. $\Phi_\mathbf{p}$ is typically chosen to have a small variance, so that only adjacent pixels need to be considered in the sample set.

A similar argument can be made for the case of joint texture-surface entropy (the second term in (1)). However, the surface S is fixed but *unknown*, since we do not assume a prior segmentation of the scene into surfaces. Consequently, it is not clear how to construct the neighborhood $\mathcal{N}(i)$. We overcome this issue by replacing the kernel function for S with a distribution that can be computed purely based on observed variables, such as the image I and pixel location P. Another advantage of this approach is that other priors based on the knowledge of the scene can be easily rolled into this distribution. Following the same derivation as above, let $V = (T, S)$ denote the joint random variable, and $\mathbf{v}_i = (\mathbf{t}_i, s_i)$ denote a sample from V. Here $\mathbf{t}_i$ is the vectorized neighborhood of pixels around $\mathbf{p}_i$ in the *reflectance* image R, while s_i is the surface that the sample belongs to. Using the same decomposition as (9), we get:

$$G(\mathbf{v}_j|\mathbf{v}_i, 2\Omega) = G(\mathbf{t}_j|\mathbf{t}_i, 2\Omega_\mathbf{t})\cdot G(s_j|s_i, 2\Omega_s). \quad (10)$$

The lack of knowledge of S implies that the second term in the above equation cannot be computed directly. We therefore replace it with the probability distribution $\Pr(s_i = s_j|I, P)$, resulting in:

$$\widetilde{G}(\mathbf{v}_j|\mathbf{v}_i, 2\Omega) = G(\mathbf{t}_j|\mathbf{t}_i, 2\Omega_\mathbf{t})\cdot \Pr(s_i = s_j|I, P) \quad (11)$$

which is a *weighted* Gaussian kernel. Note that since S is discrete-valued, the above definition is in any case more suitable than the Gaussian kernel based density estimate.

4.2. Surface and Texture Priors

We model $\Pr(s_i = s_j|I, P)$ as a product of terms corresponding to different priors. The choice of priors may depend on prior knowledge of the scene, while others may be applicable generally. A common prior is that a given surface is expected to be spatially continuous. Therefore it is best to select samples within a spatial radius around $\mathbf{p}_i$. Formally, this is expressed as:

$$\Pr(s_i = s_j|P) \propto G(\mathbf{p}_j|\mathbf{p}_i, \Omega_\mathbf{p}) \quad (12)$$

where $\Omega_\mathbf{p}$ has a large variance, allowing far away pixels to still be selected as samples. This procedure essentially performs importance sampling to compute the sum in the inner loop of (7), using the importance function in (12) for sampling.

Another prior is that surface texture appearance under different illuminations usually changes monotonically. In other words, pixels in a local spatial neighborhood around $\mathbf{p}_i$ and $\mathbf{p}_j$ should differ by nearly a constant amount for all pixels. But this notion is exactly what the kernel function $G(\mathbf{t}_j|\mathbf{t}_i, 2\Omega_\mathbf{t})$ in (11) captures, albeit based on the unknowns $(\mathbf{t}_i, \mathbf{t}_j)$. To construct a prior based on the observed

[1] l is restricted to be a scalar for now; color images are treated later.

image I, we compute the kernel function assuming the best case scenario: that the mean log-intensities at the sample pair perfectly predict the illumination difference between the samples. This results in the following prior:

$$\Pr(s_i = s_j | I) \propto G(\boldsymbol{\eta}_j - \bar{\boldsymbol{\eta}}_j | \boldsymbol{\eta}_i - \bar{\boldsymbol{\eta}}_i, 2\Omega_{\mathbf{t}}) \quad (13)$$

where $\boldsymbol{\eta}_i$ is the vectorized neighborhood of pixels around $\mathbf{p}_i$ in the original image I and $\bar{\boldsymbol{\eta}}_i$ is its mean value. We use rejection sampling to bias the samples based on this prior.

5. Energy Minimization

The total energy that we wish to minimize (1) may be rewritten as:

$$E = H(U) + \lambda H(V) =$$

$$-\sum_{i=1}^{N}\sum_{j=1}^{N}\left\{\mu_{ij} \cdot G_{2\Phi_l}(l_{ij}) + \lambda\nu_{ij} \cdot G_{2\Omega_{\mathbf{t}}}(\mathbf{t}_{ij})\right\} \quad (14)$$

where N is now the number of pixels, and μ_{ij} and ν_{ij} are fixed non-negative weights, computed based on (9) and (11). These weights are non-zero only corresponding to the illumination and texture sample sets, *i.e.* for $j \in \mathcal{N}_U(i)$ and $j \in \mathcal{N}_V(i)$ respectively, and include the normalization by number of samples. We choose $\mathcal{N}_U$ and $\mathcal{N}_V$ to be symmetric, so that $\mu_{ij} = \mu_{ji}$, $\nu_{ij} = \nu_{ji}$. We have used shorthand notation: $\mathbf{x}_{ij} \equiv \mathbf{x}_i - \mathbf{x}_j$ and $G_\Psi(\mathbf{x}_{ij}) \equiv G(\mathbf{x}_j | \mathbf{x}_i, \Psi)$.

We simplify (14) by treating l_i as locally constant for pixels used to construct $\mathbf{t}_i$. This allows us to replace the kernel function $G(\mathbf{t}_j | \mathbf{t}_i, 2\Omega_{\mathbf{t}})$ with $G(\boldsymbol{\eta}_j - l_j | \boldsymbol{\eta}_i - l_i, 2\Omega_{\mathbf{t}})$, leaving $\{l_i\}$ as the only free variables. Minimization is carried out by searching for a stationary point of E, *i.e.* where $\frac{\partial E}{\partial l_i} = 0\ \forall i$. Redefining the diagonal smoothness matrices as $\Phi_l = \frac{1}{2}\phi^2$ and $\Omega_{\mathbf{t}} = \frac{1}{2}d\omega^2\mathbf{I}$, where d is the dimensionality of $\mathbf{t}_i$, we obtain:

$$\begin{aligned}\frac{\partial E}{\partial l_i} &= 2\sum_j \mu_{ij} \cdot G_{\phi^2}(l_{ij})\left(\frac{l_{ij}}{\phi^2}\right) + \\ &\quad 2\lambda\sum_j \nu_{ij} \cdot G_{d\omega^2}(\mathbf{t}_{ij})\left(\frac{l_{ij} - \bar{\boldsymbol{\eta}}_{ij}}{\omega^2}\right) \\ &= \sum_j u_{ij} \cdot l_{ij} + \lambda\sum_j v_{ij} \cdot (l_{ij} - \bar{\boldsymbol{\eta}}_{ij}) \quad (15)\end{aligned}$$

where u_{ij} and v_{ij} are still non-negative, albeit dependent on l. Nevertheless, we can devise an iterative scheme where these weights are fixed based on the current solution l^*, resulting in a linear system of equations, which is then solved to update l^*. Note that since all weights are positive, solving $\frac{\partial E}{\partial l} = 0$ based on (15) is equivalent to solving the following weighted least squares problem:

$$l^* = \arg\min_l \sum_j u_{ij} \cdot l_{ij}^2 + \lambda\sum_j v_{ij} \cdot (l_{ij} - \bar{\boldsymbol{\eta}}_{ij})^2. \quad (16)$$

This also provides some intuition behind what is going on. The two sums in (16) are competing terms. While the first term wants to keep the illumination function locally smooth, the second term wants the change in illumination between a sample pair to match the change in their mean log-intensities. The weights make the estimation robust by suppressing outliers, down-weighting their contribution via the kernel functions.

The optimization scheme therefore reduces to an iteratively re-weighted least squares algorithm, which is solved at each iteration using Preconditioned Conjugate Gradients (PCG). The algorithm converges quickly in practice (5-10 iterations) and is not too sensitive to the initialization, especially when solved in a multi-scale fashion as described next.

Multi-scale Optimization: At every iteration, we solve a large linear system whose size is governed by the number of pixels N. However, since the illumination field is expected to be mostly smooth except at shadow boundaries, we can solve it efficiently using a multi-scale procedure. We run the optimization in a pyramidal fashion, starting at the coarsest level. At each successive level, we up-sample the illumination image computed at the previous level and use it as an initialization. Additionally, any pixel for which the illumination is sufficiently smooth in its neighborhood is excluded from the optimization and retains the value from the previous level. The size of the linear system consequently reduces from $O(N)$ to $O(N_b)$, where N_b is the number of shadow boundary pixels.

Color: To extend our technique to color images, we treat illumination at pixel i as the joint variable (l_i^r, l_i^g, l_i^b). When constructing the kernel functions, an isotropic (diagonal) smoothness matrix is used, which implies that the kernel weights for the three channels are multiplied with each other. Other than that, the iterative framework remains the same. We also exploit color information to bias the sampling prior discussed in Section 4.2. Specifically, if two regions belong to the same surface, then their average observed log-intensities in the various color channels should either all increase or all decrease, because we only expect them to differ due to an illumination change. We therefore reject sample pairs for which the condition $\mathrm{sgn}(\bar{\boldsymbol{\eta}}_j^r - \bar{\boldsymbol{\eta}}_i^r) = \mathrm{sgn}(\bar{\boldsymbol{\eta}}_j^g - \bar{\boldsymbol{\eta}}_i^g) = \mathrm{sgn}(\bar{\boldsymbol{\eta}}_j^b - \bar{\boldsymbol{\eta}}_i^b)$ is not satisfied. Additionally, for outdoor scenes, we expect the illumination to be white in well lit regions, and close to white in the dark regions – there is a slight shift in the spectrum in dark regions due to ambient light. We therefore reject samples that do not satisfy: $\max(|\xi^r - \xi^g|, |\xi^g - \xi^b|, |\xi^b - \xi^r|) < \gamma \max(\xi^r, \xi^g, \xi^b)$, where $\xi^a = |\bar{\boldsymbol{\eta}}_i^a - \bar{\boldsymbol{\eta}}_j^a|$ and γ is a threshold (0.2 in our experiments).

Figure 2: Left-to-right: Original image; Clusters of illumination values, cluster 1 is the representative non-shadow cluster; Saturated result if cluster 2 is wrongly picked as the non-shadow cluster; Darkened result if cluster 3 is chosen; Our generated non-shadow mask based on cluster 1; Shadow removal result based on generated mask.

6. Processing Aerial Imagery

One of the applications that we have employed our approach is for shadow removal in aerial imagery. We consider two classes of images. Firstly, we have images from long-distance shots, that may be used for generating satellite and 45° views in maps. These images often contain cloud shadows, which can be fairly diffuse and large, and therefore troublesome to remove. Secondly, we consider rectified images of building facades used for texturing 3D models in virtual environments, which often have distinctive texture patterns. Another practical requirement is to be able to process fairly high resolution images, automatically.

6.1. Building Facades

A characteristic of building facades is that they usually exhibit strong periodic patterns. This is especially true in our case, since we work with *rectified* facades, which are ready for texture mapping onto 3D models. We exploit this prior knowledge to adapt the sampling of pairs in (12) to limit them to *axis-aligned* periodic neighbors only. In order to extract the periods p_x for the x direction and p_y for the y direction, normalized correlation is computed between each pixel position and its corresponding positions obtained by shifting the facade along x or y directions. The correlation values with different shifts are then normalized for each position, and used as a weighted vote for the shift. These votes are accumulated from all the pixels in the image, and the first salient peaks (other than 0) in each of the x and y directions are selected as the periods along those axes. The entire computation can be performed efficiently by implementing it as multiple 1D auto-correlations.

6.2. Automatic Non-Shadow Area Detection

Our optimization requires the illumination to be constrained at some pixels. Given a rough mask containing only non-shadowed pixels, we can constrain $l_i = 0$ at those pixels. This mask need not be accurate, it just needs to be conservative to exclude shadows, and may either be user specified or determined automatically. An automatic approach for mask initialization is as follows.

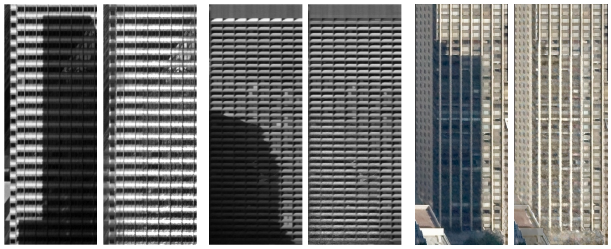


Figure 3: Shadow removal results on building facades.

Firstly, we compute an initial illumination image by filtering the original image. For facade images with periodic patterns, a box filter of size $p_x \times p_y$ is used to get a good approximation. The interesting property of such a filter is that convolving it with an ideal repetitive pattern with periods p_x and p_y will give a constant output, which can be considered as the illumination value. For cloud and natural images, we use bilateral filtering with a kernel of size 21×21. We then perform k-means clustering on these pixel-wise illumination values. The goal is to pick the most likely cluster containing only non-shadow pixels. We set $k = 4$ for clusters with non-shadow pixels, non-shadow outliers (such as very bright or saturated pixels), shadow pixels, and shadow outliers (such as dark areas with little color information).

We train an Adaboost classifier to pick the best cluster to represent non-shadow regions. The features extracted from each cluster include: illumination value l_i of cluster center, cluster size, variance of illumination values, mean intensity value I_i from the original image, and normalized mean intensity value of other clusters if i is picked as the non-shadow cluster: $I_j \times \frac{l_i}{l_j}$ for $j \neq i$. The intuition is that if cluster i is the representative non-shadow cluster, l_i and l_j together would normalize the pixels from other clusters, including shadow clusters, to the correct intensity. Otherwise, picking the wrong cluster will saturate or change the

color of other clusters, as shown in Figure 2. The classifier is trained using 90 images, generated by simple user strokes. On 43 test images, we obtain a precision of 98% at 93% recall on cluster-level classification.

7. Results and Discussion

We have applied our shadow removal algorithm to aerial images including building facades and cloud shadows, as well as natural scenes. Aerial images with cloud shadows (Figure 4 and 5) are likely to have soft shadow boundaries, whereas building facades (Figure 3) and natural photos (Figure 6) more likely contain hard shadow boundaries.

Since we minimize texture entropy over the entire shadow region, as opposed to just near a hard shadow boundary, we obtain smoothly varying shadow (inverse illumination) maps in soft shadow regions, as shown in Figure 5. We have used our algorithm to remove shadows from high resolution images by employing the multi-scale optimization described in Section 5. To handle resolutions as high as 64 MPixel, we additionally employ a tiled approach where the entire image is divided into a grid of overlapping tiles, which are solved sequentially from left-to-right and top-to-bottom. The overlap provides constraints which ensure consistency across tiles. Figure 4 shows zoom-ins to selected crop regions to portray the resolution of the underlying images and the softness of removed shadows. Figure 1, 2 and 3 show examples of shadow removal in building facades. We are successfully able to remove hard shadows by exploiting the periodic structures of these facades.

Figure 6a compares the result of applying our algorithm on a photograph with hard shadows to two other state-of-the-art approaches. Of these, we out-perform [7], which is also an entropy minimization based approach, albeit their entropy is defined over a different quantity. Our result in this case is comparable to [14], which is especially designed for hard shadow boundaries. The mask we use here is a hybrid of automatic detection and user specification. The initial mask (red+blue pixels) is refined by the user by removing the blue pixels and keeping the red ones. The automated initialization makes the user refinement relatively simple.

Limitations: While our method generates reasonable results on hard shadows, it can sometimes get confused between appearance and illumination changes. In Figure 6b, the algorithm treats a portion of the white stripe near the shadow boundary as part of the road, and therefore fails to remove the shadow cleanly. More generally, presence of multiple textures near hard shadow boundaries can sometimes cause failures, or require careful parameter selection for texture entropy. Another limitation is that the initial mask can sometimes misclassify shadow/non-shadow pixels, resulting in missed shadows or brightening of dark regions, *e.g.* the river in Figure 4 (middle-row) is incorrectly

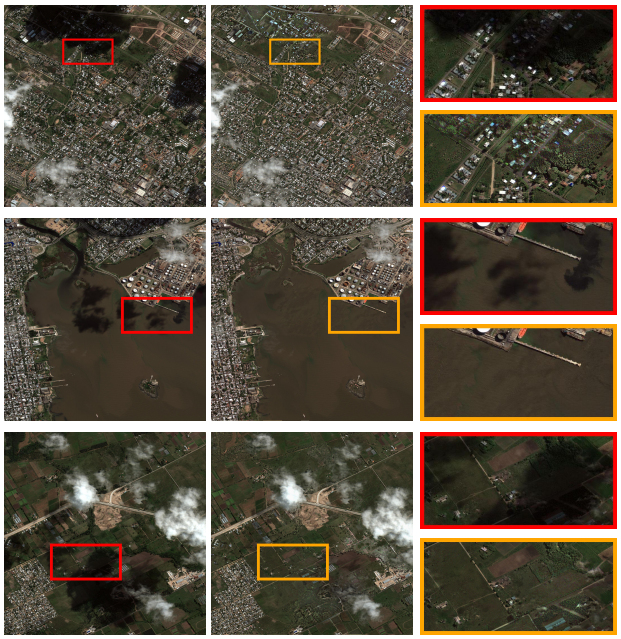


Figure 4: Cloud shadow removal on high-resolution images (64 MPixel). Last column in each row shows zoomed-in versions of before (red, top) and after (orange, bottom) results for corresponding crop rectangles in full images.

treated as a shadow. For semi-automated removal, this can be addressed with user interaction as done in Figure 6a. In future work, we wish to incorporate classification scores directly into the entropy minimization in a soft fashion.

Parameters and Runtime performance: In our experiments, we usually set $\lambda = 1.0$, the diagonal bandwidth matrices use variance of 0.2 for the illumination term, while the texture term uses the the average variance over all image patches. The multi-resolution and tiled solvers allow us to process 512×512 images in 10-20 seconds and 64 MPixel images in 5-10 minutes on a 3.5GHz, 6-core, 12GB RAM Intel Xeon workstation.

8. Conclusion

We have presented a novel approach for shadow removal based on the minimization of texture and smoothness entropy of the image. The texture term enables estimation of long range illumination changes, while the smoothness term encourages smooth illumination fields. The entropic formulation results in a robust estimation that respects shadow boundaries as well as texture variations. It works especially well for soft shadows and distinctive texture patterns, as we have demonstrated through our results on aerial imagery, but is general enough for natural scenes. Our optimization algorithm is efficient and scalable, and may be applicable to other kernel-based methods, to be explored as future work.

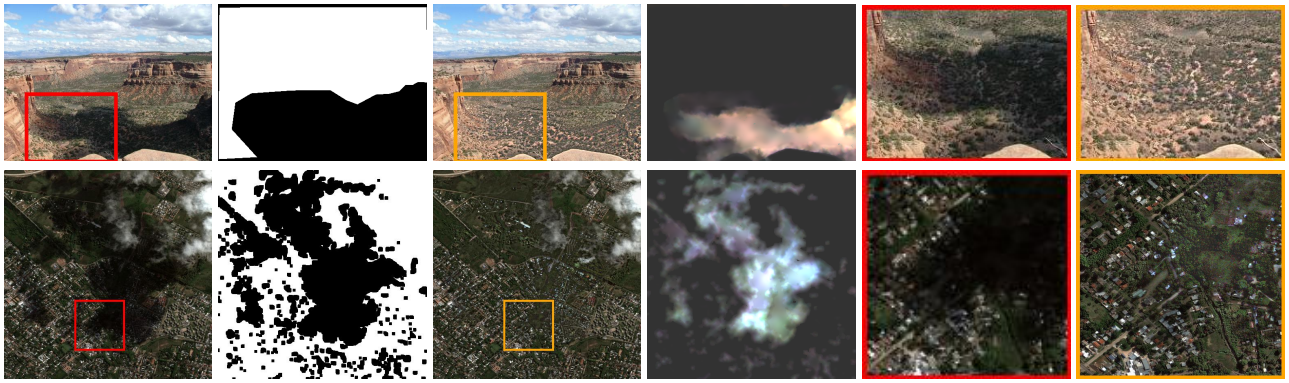
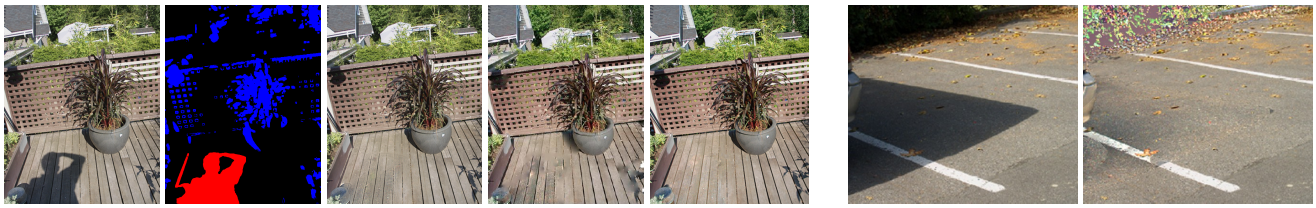


Figure 5: Left-to-right: original image, non-shadow mask, shadow removed result, computed shadow map, zoomed-in before (red) and after (orange) results for selective crops. Top row uses loosely specified user mask, bottom row uses automatic mask.



(a) (b)

Figure 6: Shadow removal on general natural images. (a) From left-to-right: original image; non-shadow mask (blue+red was initial automatic mask, blue pixels were removed by user, red pixels form the final mask); our result; result from [7]; result from [14]. (b) Another before-and-after example, also showing limitations (discussed in text).

References

[1] E. Arbel and H. Hel-Or. Texture-preserving shadow removal in color images containing curved surfaces. In *CVPR*, 2007. 2

[2] S. P. Awate and R. T. Whitaker. Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:364–376, 2006. 2

[3] A. Bousseau, S. Paris, and F. Durand. User-assisted intrinsic images. In *SIGGRAPH Asia*, 2009. 2

[4] P. M. Dare. Shadow analysis in high-resolution satellite imagery of urban areas. *Photogrammetric Engineering Remote Sensing*, 71(2):169–177, 2005. 2

[5] G. Finlayson, M. Drew, and C. Lu. Entropy minimization for shadow removal. *International Journal of Computer Vision*, 2009. 2, 3

[6] G. Finlayson, C. Fredembach, and M. Drew. Detecting illumination in images. In *ICCV*, 2007. 2

[7] M. S. D. Graham D. Finlayson and C. Lu. Intrinsic images by entropy minimization. In *ECCV*, 2004. 2, 7, 8

[8] R. Guo, Q. Dai, and D. Hoiem. Single-image shadow detection and removal using paired regions. In *CVPR*, pages 2033–2040, 2011. 2

[9] X. Jiang, A. Schofield, and J. Wyatt. Correlation-based intrinsic image extraction from a single image. In *ECCV*, 2010. 2

[10] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Detecting ground shadows in outdoor consumer photographs. In *ECCV*, 2010. 2

[11] F. Liu and M. Gleicher. Texture-consistent shadow removal. In *ECCV*, 2008. 2

[12] B. Maxwell, R. Friedhoff, and C. Smith. A bi-illuminant dichromatic reflection model for understanding images. In *CVPR*, 2008. 2

[13] S. Narasimhan, V. Ramesh, and S. Nayar. A class of photometric invariants: Separating material from shape and illumination. In *ICCV*, 2003. 2

[14] Y. Shor and D. Lischinski. The shadow meets the mask: Pyramid-based shadow removal. In *Eurographics*, 2008. 2, 7, 8

[15] M. Tappen, W. Freeman, and E. Adelson. Recovering intrinsic images from a single image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2005. 2

[16] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV*, 2001. 2

[17] T.-P. Wu and C.-K. Tang. A bayesian approach for shadow extraction from a single image. In *ICCV*, 2005. 2

[18] J. Zhu, K. Samuel, S. Z. Masood, and M. Tappen. Learning to recognize shadows in monochromatic natural images. In *CVPR*, 2010. 2