

Analysis of Orientation and Scale in Smoothly Varying Textures

Jason Chang

Massachusetts Institute of Technology
32 Vassar St. Cambridge, MA

jchang7@csail.mit.edu

John W. Fisher III

Massachusetts Institute of Technology
32 Vassar St. Cambridge, MA

fisher@csail.mit.edu

Abstract

We present a novel representation for modeling textured regions subject to smooth variations in orientation and scale. Utilizing the steerable pyramid of Simoncelli and Freeman as a basis, we decompose textured regions of natural images into explicit local attributes of contrast, bias, scale, and orientation. Additionally, we impose smoothness on these attributes via Markov random fields. The combination allows for demonstrable improvements in common scene analysis applications including unsupervised segmentation, reflectance and shading estimation, and estimation of the radiometric response function from a single image.

1. Introduction

Texture modeling is a widely studied problem in computer vision with many applications. For example, when using a generative model, one can replicate and extend textures in a random but visually appealing fashion (e.g. [11]). In a discriminative model, a robust texture representation can aid in segmentation to distinguish different objects (e.g. [7]). Here, we present a texture model subject to a deformation field that when combined with information measures is useful for a variety of common computer vision tasks.

Texture analysis is of particular interest for images comprised of textured regions, as such images pose a challenging segmentation task. Many segmentation algorithms assume that observed pixels are statistically independent *conditioned* on the region label. While this assumption does not generally hold, it also does not greatly impact segmentation of non-textured images where it is common to model spatial correlations only in the region labels. However, such correlations cannot be discounted when segmenting textures.

In order to capture smooth variations in scale and orientation of textures, we suggest an image model which explicitly decomposes local texture analysis into four attributes: contrast, bias, scale and orientation. Via Markov random field (MRF) modeling, we impose smoothness in these at-

tributes. In natural images, the perception of such variations may occur for a variety of reasons, including actual changes on the surface or perspective effects of the imaging device. We show that by including such variations in our model, one can improve many texture modeling applications.

There is extensive previous work on applying texture models to image segmentation. For example, [10] and [14] represent each texture with a constant measure within a region. In contrast to the method presented here, these methods either do not consider variations in scale and orientation or treat them as nuisance parameters. Other methods have been developed based on analysis of filter-bank responses. For example, [3] has looked at using wavelets and [4] utilizes Gabor filters. The work in [14] presents an approach that uses a set of Gabor filters, observing that changes in scale and orientation are manifested as shifts in their feature space. Montoya-Zegarra et al. [10] proposed a method using steerable pyramids [6] where they treat each filter output independently. One drawback of these approaches is the coarse discretization of scale and orientation.

We also utilize the steerable pyramid as a precursor to texture analysis, however, we exploit the property that responses at any orientation can be efficiently computed with bounded error. We suggest a novel texture analysis of the response and demonstrate that it accurately measures the local scale and orientation of textures. Furthermore, as a consequence of imposed smoothness in our feature set via MRFs, we obtain a method for estimating a simple model of the nonlinear intensity response of a camera from a single image. We compare to the method of [5] showing significantly better performance. Consequently, we obtain an accurate estimate of the irradiance image that when combined with the aforementioned smoothness assumptions enables estimation of shading and reflectance for textured objects.

2. Texture Descriptors

One goal of the prescribed attributes is to detect and measure the dominant orientation and scale at each point in an image. For our purposes, the features are useful in that they allow one to decompose an image into contrast, bias, scale,

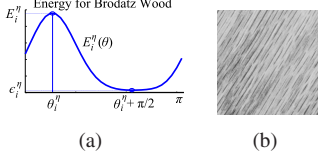


Figure 1. (a) Features at a given scale at the center pixel w.r.t. orientation; (b) A Brodatz image used in the plot

and orientation fields. Smoothness constraints are then imposed separately on each field. While the measure implicitly assumes a dominant orientation, we also demonstrate the utility of the measure on textures with more than one dominant orientation and with non-textured regions.

2.1. Extracting the Dominant Scale and Orientation

Within the steerable pyramid representation, the filter output at any orientation can be well approximated [6] by interpolating the outputs of the filter bank at a discrete set of orientations. Within each scale, denoted by η , we use four orientations: $y^\eta(0)$, $y^\eta(\frac{\pi}{4})$, $y^\eta(\frac{\pi}{2})$, and $y^\eta(\frac{3\pi}{4})$. The filter output at an arbitrary orientation, θ , can be found by:

$$y_i^\eta(\theta) = \sum_{\phi} \frac{y_i^\eta(\phi) [\cos(\theta - \phi) + \cos(3(\theta - \phi))]}{2} \quad (1)$$

where i is a pixel location and $\phi \in \{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\}$. Let

$$E_i^\eta(\theta) = \frac{1}{|R_i^\eta|} \sum_{j \in R_i^\eta} |y_j(\theta)|^2 \quad (2)$$

be the angular energy over a local region R_i^η (typically 3-by-3) at an angle θ , scale η and location i . Using golden section search [8], we find the orientation within each scale, θ_i^η , with the maximum angular energy. This is defined to be the orientation of the texture at scale η , that is

$$\theta_i^\eta = \arg \max_{\theta} E_i^\eta(\theta). \quad (3)$$

As $E_i^\eta(\theta)$ is periodic with period π , we only search the range $[0, \pi)$. As a function of scale η and location i , we extract the contrast energy

$$E_i^\eta = E_i^\eta(\theta_i^\eta) \quad (4)$$

and what we refer to as the residual contrast energy

$$\epsilon_i^\eta = E_i^\eta\left(\theta_i^\eta + \frac{\pi}{2}\right). \quad (5)$$

The term residual energy is used since, for strongly oriented textures, the energy of the response when the filter is orthogonal to the dominant orientation tends to be small. These features are depicted graphically in Figure 1. Within a scale, the last feature that we introduce is the bias of the

texture, μ_i^η . This term captures the low-frequency energy of the image that is not directly measured by the steerable pyramid. While nearly any lowpass filter is suitable, we use a circularly symmetric Gaussian blur filter.

Summarizing, in each scale and at each pixel, we calculate the contrast energy (E_i^η), the residual energy (ϵ_i^η), the orientation (θ_i^η), and the bias (μ_i^η). For each location i we then select the set of features corresponding to the scale, η_i , with maximum contrast energy:

$$\eta_i = \arg \max_{\eta} E_i^\eta. \quad (6)$$

These become the feature set at the pixel. We drop the superscript of scale for each feature once we have η_i . This leads us to our final feature set at pixel i : $\{\eta_i, E_i, \epsilon_i, \theta_i, \mu_i\}$.

Excepting η_i , all features are continuous. We sample scale logarithmically in order to approximate η as a continuous parameter. Specifically, we use scale factors of: unity, $2^{-0.25}$, $2^{-0.5}$, and $2^{-0.75}$. Sampling at a finer scale can be accomplished at the cost of memory and computation time. Hereafter, the detected scale feature is treated as a continuous quantity.

2.2. Smoothness Priors on Extracted Features

Following our definition, we take each measured attribute as the output of a smooth MRF *conditioned* on the segmentation label. Noise is assumed to be additive. The model for each of these random fields is similar modulo minor parameter choices. For reasons of brevity, we derive the inference procedure for the orientation field only; the derivation for scale, contrast, and bias follow accordingly. One notable departure from standard MRF methods is that we model the additive noise term in each field nonparametrically. This is due to the fact that the “noise” terms correspond to quantities of interest. Effectively, smoothness is utilized as a means to separate the intrinsic features from the observed features. We also derive a novel fixed point update utilizing the nonparametric model.

Under a standard Gaussian MRF model of smooth variations one assumes the observation model shown in Figure 2(a), where \underline{n} is typically Gaussian and $\hat{\phi}$ is a smooth MRF. However, in the types of images we consider, the fields within a commonly labeled region may be locally smooth while exhibiting a small number of abrupt changes. One way to account for this is to add an additional term which is piecewise constant as in Figure 2(b), where \underline{c} is a random variable capturing the abrupt changes and $\hat{\phi}$ is completely smooth. Under the assumption that the abrupt changes in \underline{c} are large relative to the standard deviation of \underline{n} , this is equivalent to an MRF where the additive noise term comprised of both \underline{c} and \underline{n} is equivalent to a mixture of Gaussians or, more generally, a kernel density estimate (KDE).

In total we infer four different smooth fields: an orientation field on $\underline{\theta}$, a scale field on $\underline{\eta}$, a gain field on \underline{E} , and

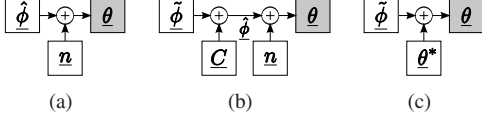


Figure 2. Smooth field model: (a) $\hat{\phi}$ is smooth with abrupt changes; (b) $\tilde{\phi}$ is completely smooth; (c) Intrinsic image $\underline{\theta}^*$

a bias field on $\underline{\mu}$. It is important to note that the implied residual noise terms are of interest as well. This is most apparent when considering the gain and bias fields whose specific details are covered later. These two fields capture a slowly changing contrast and bias. After the gain and bias have been removed from the original image, the so called *noise*, $\underline{C} + \underline{n}$, is an image that has a uniform contrast and bias, by which we approximate the intrinsic image. Likewise, the residual terms associated with the orientation and scale fields provide what we call the intrinsic texture image. Upon estimating each field, the image can be reconstructed with a uniform contrast and bias and with a single scale and orientation. A visualization of this is shown in Figure 6(g).

Ultimately, we iterate between estimation of these fields conditioned on a given segmentation and segmentation based upon the estimated intrinsic texture images. Figure 2(c) depicts the observation model incorporating the intrinsic unoriented texture image, $\underline{\theta}^*$ where the superscript, *, denotes the intrinsic feature.

We formulate MAP estimation of the various fields in a similar fashion to [15], with the primary difference being the nonparametric noise model. Furthermore, as a consequence of the nonparametric model, we derive a fixed point iteration in the absence of an analytic solution. A key assumption is that each θ_i is *i.i.d.* conditioned on ϕ_i . A full derivation can be found in [2]. We assume that $\tilde{\phi} \sim \mathcal{N}(\underline{\theta}, \Lambda_\phi)$.

$$\tilde{\phi} = \arg \max_{\phi} P(\phi | \underline{\theta}) \quad (7)$$

Using Bayes' rule and differentiating w.r.t. ϕ , we arrive at the following equality

$$\underline{\theta} - \tilde{\phi} - w_p(\underline{\theta} - \tilde{\phi}) - \frac{h^2}{2} \Lambda_\phi^{-1} \tilde{\phi} = 0, \quad (8)$$

where h is the bandwidth used in the KDE, and $w_p(\cdot)$ is the ratio of a weighted KDE to the KDE defined as follows

$$w_p(\cdot) = \frac{\sum_{s=1}^N (\theta_s - \tilde{\phi}_s) e^{-\frac{(\cdot - \theta_s + \tilde{\phi}_s)^2}{h^2}}}{\sum_{s=1}^N e^{-\frac{(\cdot - \theta_s + \tilde{\phi}_s)^2}{h^2}}}. \quad (9)$$

Rearranging this equation results in a fixed-point iteration to solve for $\tilde{\phi}$

$$\tilde{\phi}^{(k+1)} = F^{-1} \left(\underline{\theta} - w_p \left(\underline{\theta} - \tilde{\phi}^{(k)} \right) \right), \quad (10)$$

where the matrix, F , is defined as follows

$$F = \left(\frac{2}{h^2} \Lambda_\phi \right)^{-1} + I = F_1^{-1} + I. \quad (11)$$

As in [15], by treating Λ_ϕ as the result of filtered i.i.d. noise, the fixed-point update can be computed efficiently using an FFT. Consequently, we write the matrix, F_1 , in (11) as follows

$$F_1 = \frac{2}{h^2} \Lambda_\phi = \frac{2}{h^2} H \sigma_\phi^2 I H^T = \frac{2\sigma_\phi^2}{h^2} H H^T \quad (12)$$

where H is the matrix that performs a convolution with a unity DC gain lowpass filter and σ_ϕ^2 is the variance of ϕ_i . This implies that multiplying by the covariance matrix, Λ_ϕ , is equivalent to applying two lowpass filters. In general, the bandwidth used to estimate the PDF, h , is chosen to be much smaller than the variance of the smooth field, σ_ϕ^2 . With this assumption, it can be shown that F^{-1} in the fixed-point update of (10) must be a lowpass filter with unity DC gain [2].

2.3. Specifics of Various Fields

While each of the separate smooth fields is modeled similarly, there are differences between them which we briefly explain now. For the orientation feature, $\underline{\theta}$, we assume there is an underlying smooth, additive orientation field, ϕ . We cannot use a conventional lowpass filter in (10) because the orientation is periodic (with period π). Instead, we double each angle and convert it into the sine and cosine components (i.e. mapping it onto the unit circle). We lowpass these X and Y coordinates, take the inverse tangent, and then half the angle to find the equivalent lowpassed angle. To find the intrinsic angle, we simply subtract the smooth field from our measurement:

$$\theta_i^* = \theta_i - \phi_i. \quad (13)$$

As mentioned, we treat the discrete scale measurement as a continuous quantity. The bandwidth of the KDE is chosen to be large enough to sufficiently smooth over adjacent scales. The additive term is denoted by the symbol $\underline{\nu}$ and the intrinsic scale is computed as

$$\eta_i^* = \eta_i - \nu_i. \quad (14)$$

The treatment of the remaining fields is somewhat more complex. The gain field \underline{g} , is a multiplicative field accounting for contrast changes. The bias field, \underline{b} , is an additive field that corrects the low-frequency component of the texture. We impose smoothness on the log of the gain field allowing us to treat it as an additive field. Having accounted for these fields, we are left with an estimate of the intrinsic reflectance image, \underline{R} . Figure 5(a) depicts this relationship.

In contrast to the orientation and scale fields, the gain and bias fields both operate directly on the pixel intensities.

To find the reflectance image, we divide our observed image by the gain field and then subtract the bias field. As the steerable pyramid is comprised of linear filters, dividing the image by a value implies that the filter outputs are divided by the same value. Thus, both the energy and bias features of our texture are affected by the gain field. The contrast energy captures the filter response at some bandpass frequency, whereas the bias captures low frequency components. Consequently, adding a constant value should only change the low frequency bias and not the contrast energy. If we assume that the bias field can correct for any changes that the gain field has made to the bias term, then we can estimate these two fields independently. The gain field is first estimated assuming that it only affects the contrast energy, and then the bias field is estimated only based on the bias term. If we assume that these fields are sufficiently smooth within a small neighborhood, then we can approximate the effect of them on the intrinsic features as follows

$$E_i^* = \frac{1}{|R_i^{\eta_i}|} \sum_{j \in R_i^{\eta_i}} \left| \frac{y_j(\theta_i^{\eta_i})}{g_j} \right|^2 \approx \frac{E_i}{g_i^2}, \quad (15)$$

$$\mu_i^* = \sum_{j \in R_{f,i}^{\eta_i}} \left(\frac{x_j}{g_j} - b_j \right) \cdot f_j \approx \frac{\mu_i}{g_i} - b_i \sum_{j \in R_{f,i}^{\eta_i}} f_j = \frac{\mu_i}{g_i} - b_i, \quad (16)$$

where $R_{f,i}^{\eta_i}$ is the support of the lowpass filter around pixel i and in scale η_i , and f_j is a lowpass filter coefficient. Note that the last equality in (16) is by design since our filter has constant DC gain.

The last implementation detail for field estimation is the assumed degree of smoothness. This is determined by the properties of the covariance matrices associated with each field, which is in our case is determined by the structure of the filter used in (10). In our experiments, we use a 15x15 averaging filter for each field, noting that the segmentation is not overly sensitive to the window size. Another parameter that we can tune is the bandwidth of our KDE. Larger bandwidths equate to grouping more regions into the same intrinsic texture representation. This is advantageous because it allows us to capture more extreme lighting and geometrical effects. However, if the bandwidth is too large, textures that belong to two different intrinsic textures will look the same. We chose a minimum bandwidth for each smooth field and a scaling factor to the “rule of thumb” bandwidth [12] that provided pleasing results, noting that the segmentation is not very sensitive to a fairly large range.

3. Segmentation

One obvious application of our novel feature set is segmentation. We use a level-set based method for segmentation following the method of Kim et al. [9]. In that work, the criterion for the level-set evolution is the maximization

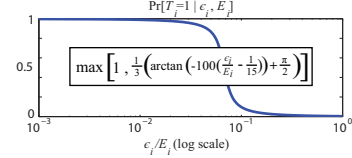


Figure 3. Probability that pixel i is strongly oriented

of the mutual information between a pixel intensity and the pixel label. This choice of criterion allows for unsupervised segmentation of images provided regions are distinguished by homogeneous statistical properties. Differing from that work, we replace single pixel intensities with our estimates of the *intrinsic* features. The form of the gradient flow does not change, but the induced log-likelihood ratio is now over feature vectors rather than pixel intensities. As a side, we note that it can be shown that under the assumption that feature vectors are independent conditioned on the region label, maximizing the empirical mutual information is equivalent to finding the MAP estimate of the labeling conditioned on the observation [2].

Recall that one of our features, ϵ , is called the residual energy. The ratio, $\frac{E}{\epsilon}$ is similar to an SNR, and is related to how strongly oriented the texture is. We note that the scale and orientation with the highest contrast energy is only accurate if the texture is in fact oriented. Thus, we define the probability of a texture being strongly oriented as a Bernoulli random variable, T , with a PMF that is a function of the ratio $\frac{E}{\epsilon}$. This function is empirically chosen to have a fairly sharp transition when the ratio is approximately 15. The specific function chosen is displayed in Figure 3, though we note that any similar function would suffice. The residual energy only plays a role in calculating the probability of the texture being strongly oriented. If the texture is not strongly oriented, we model the scale and orientation as being drawn from uniform distributions (*i.e.* their values do not influence the segmentation). Given the number of samples, we use the fast Gauss transform [16] to approximate these quantities efficiently. The likelihood function for a given pixel i conditioned on being in region R^+ or R^- is then

$$p(E_i, \mu_i, \eta_i, \theta_i | R^\pm) = p_E^\pm(E_i) p_\mu^\pm(\mu_i) p_\eta^\pm(\eta_i) p_\theta^\pm(\theta_i) \quad (17)$$

where the marginal terms over E_i and μ_i are estimated using KDEs and the marginal terms over η_i and θ_i are estimated using a mixture of a KDE and uniform distribution dependent on the value of T .

3.1. Region Boundary Effects

One important issue is differentiating edges between objects from edges within a texture. Because we are using a region based approach, we would like to capture the effects of the edges within a texture, and not the edges between

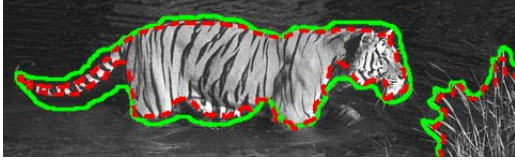


Figure 4. Zoomed in portion of segmentation. Green solid line is before refinement, red dotted line is after refinement

objects. The edges between objects manifest themselves in two different ways. One way is a result of using a local texture measure. The angular energy in (2) was defined over a region, R_i , around pixel i . When we consider pixels near the boundary of our segmentation, we include values from different regions. Thus, we define a new region, R'_i , which is similar to R_i , but also confined to the segmented region, R^\pm .

$$R'_i = R_i \cap R^\pm \quad (18)$$

Using this new local region increases computational complexity and the number of local extrema. We address this issue by first segmenting based on R_i , and then refining our segmentation based on R'_i .

Boundaries between objects also affect our model as a direct result of the steerable pyramid. The coefficients of the filter outputs contain high energy at object edges. We address this issue by extending the textured regions beyond each region, R^\pm , and recalculating the steerable pyramid at every iteration. This also increases computational cost. Consequently, we recalculate the steerable pyramid only at iterations when we are using our refinement based on R'_i . Figure 4 shows the results before and after considering boundary effects. Clearly, the refinement aids in correcting for boundary effects.

3.2. Final Algorithm

We design our final segmentation algorithm by combining our texture measure with the method in Kim et al. [9], and employing the multi-region segmentation of Brox and Weikert [1]. Pseudocode for our algorithm is shown in Algorithms 1 and 2.

4. Nonlinear Intensity Correction

Upon segmenting the image, we can use the results to extract additional information from the image. For example, one can estimate a parametric radiometric function from a single observed image. The radiometric function changes the raw, irradiance data from the camera sensor to the pixel intensities that are saved in the image. The irradiance image can be decomposed into a shading image and reflectance image. Figure 5(a) shows how the reflectance image, \mathcal{R} , fits into our model. Figure 5(b) shows our simple physical model for an unknown camera, where \mathcal{S} is the shading

Algorithm 1 Segment an image

- 1: Subsample scale by creating 4 scaled images
 - 2: **for** each scaled image \mathcal{X} **do**
 - 3: Calculate the steerable pyramid of \mathcal{X}
 - 4: **end for**
 - 5: Start with one region covering the entire image
 - 6: **while** segmented regions < regions of image **do**
 - 7: **for** each segmented region R **do**
 - 8: Temporarily segment R using Algorithm 2
 - 9: **end for**
 - 10: Choose the segmentation with the highest energy
 - 11: Jointly evolve all curves for a global extremum
 - 12: **end while**
-

Algorithm 2 Segment a region within the image

- 1: **while** not converged or features not refined **do**
 - 2: **if** converged or features refined **then**
 - 3: Refine our features
 - 4: **end if**
 - 5: Estimate the smooth fields, g , b , ϕ , and ν
 - 6: Estimate the intrinsic texture image
 - 7: Segment the intrinsic texture image only evolving curves within the region
 - 8: **end while**
-

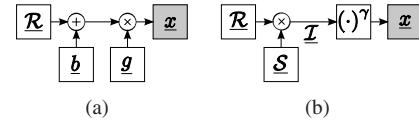


Figure 5. Model for the intrinsic reflectance image with the observed image x : (a) A gain and bias field account for the camera model; (b) A one parameter, γ model for an unknown camera

image, and \mathcal{I} is the irradiance image. Here, we see that the two models in Figure 5 are very similar. As described previously, we first estimate the gain field and then the bias field. If there was no radiometric response function ($\gamma = 1$), then the lighting effects could be completely explained by a multiplicative gain field, and our estimate of the bias field should have very little energy. Thus, in the absence of γ , our gain field is approximately our shading image. We exploit this observation to find the optimal γ estimate in our camera model.

Given a value of γ , we can estimate the gain and bias fields as stated in Section 2.2. Assuming a noiseless model, the reflectance image is simply

$$\mathcal{R}_i(\gamma) = \frac{x_i^{1/\gamma}}{g_i(\gamma)} - b_i(\gamma), \quad \forall i \in \Omega. \quad (19)$$

Setting the bias field to be zero everywhere, which would only occur if the camera model was correct, we reconstruct the observed image using only the reflectance image and

the gain field for a given γ value. The reconstruction error energy under the zero bias field assumption is

$$e(\gamma) = \|(\underline{\mathcal{R}}(\gamma) \cdot \underline{g}(\gamma))^\gamma - \underline{x}\|_2, \quad (20)$$

where all operations are done element-wise. We find the optimal γ value by minimizing (20) within the range [0.2, 1] using golden section search [8].

5. Shading and Reflectance Decomposition

Given the camera model (i.e. γ), it is straightforward to determine the shading image. One could use the estimated gain field as the shading image, but we have observed empirically that re-estimating the gain field assuming the bias field is zero produces better results. Here, we take into account that the bias feature, $\underline{\mu}$ is also affected by the gain field. As we assume that these two values are independent, it is straightforward to make this slight modification. The details can be found in [2].

6. Empirical Results

We present examples using our feature set in segmentation, radiometric response estimation, and shading estimation. Additional results can be found in [2]. For the example images given here, we show that our method provides a reliable result in all of these common tasks. We evaluate our algorithm by comparing to other related methods.

6.1. Feature Analysis

Figure 6 shows the decomposition of our feature set after running our segmentation algorithm. Our measured features are already fairly smooth and show that they provide a good basis for distinguishing the two regions. Figure 6(e) shows our orientation measure where the length of the vector is proportional to the probability of it being strongly oriented. The orientation vectors line up very well with the actual stripes indicating an accurate measure of orientation. Figure 6(f) shows our estimated reflectance image after removing the effects of the gain and bias field. Although this result shows good performance, it still has flaws near the top of the back of the zebra where we see residual lighting effects. In Figure 6(g), we show a visualization of the intrinsic texture image. The blocky artifacts are a result of how we create this image. This visualization is obtained by rotating and scaling small regions of the reflectance image with the estimated orientation and scale fields and is for illustrative purposes only.

6.2. Segmentation

Figure 7 shows the segmentation results. The left column illustrates the scalar segmentation described in [9] which is

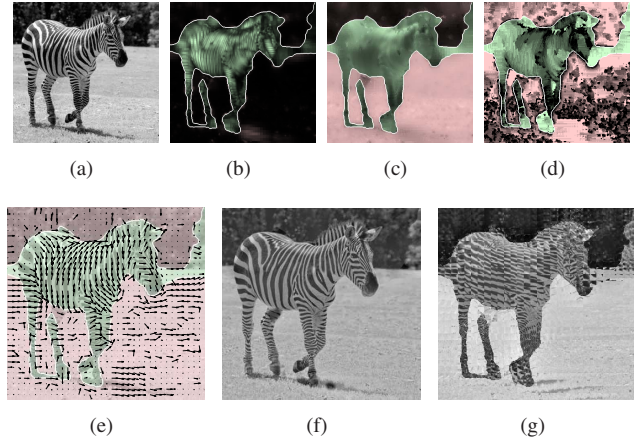


Figure 6. Calculated feature set: (a) Original image; (b) Contrast energy; (c) Bias; (d) Scale (brighter is coarser scale); (e) Orientation; (f) Estimated reflectance image; (g) Visualization of the estimated intrinsic texture image

based solely on intensity values. The middle column is obtained from the algorithm described in [7]. In contrast to our method which exploits joint dependencies, this method uses all outputs of the filter bank assuming they are statistically independent. For this method, we used four orientation and three scales totaling to 12 features. The *exact* algorithms of [9] and [7] produce very poor results on the example images. In order to provide a fairer comparison, we included our gain and bias field derivation into their approaches to show the difference gained by our texture model by incorporating lighting effects into all algorithms. The right column is obtained using our algorithm for segmentation. Here we assume that the number of regions is given, though methods exist for determining this number automatically.

The leopard and tiger images are both segmented very well with our algorithm because the objects contain a slowly changing texture, even though the leopard does not contain a strongly oriented texture. The results obtained using [9] are also fairly good because the animals have a very different intensity compared to the background.

The baseball field was segmented into three regions. The scalar segmentation groups some of the background with the grass, and the sky with the dirt. Our method is able to group all of the background and sky into one region due to the faint texture in the grass. Although the grass is perceived as being very textured, the contrast between the dirt and grass is actually much higher than the contrast in the grass. This is the reason that the independent steerable pyramid algorithm performs poorly. Our bias feature and slowly varying orientation help our algorithm to capture a more accurate segmentation. However, our algorithm is not able to capture the grass between the mound and second base. This error is a result of the region being fairly small, and our local measure combined with the boundary effects and curvature

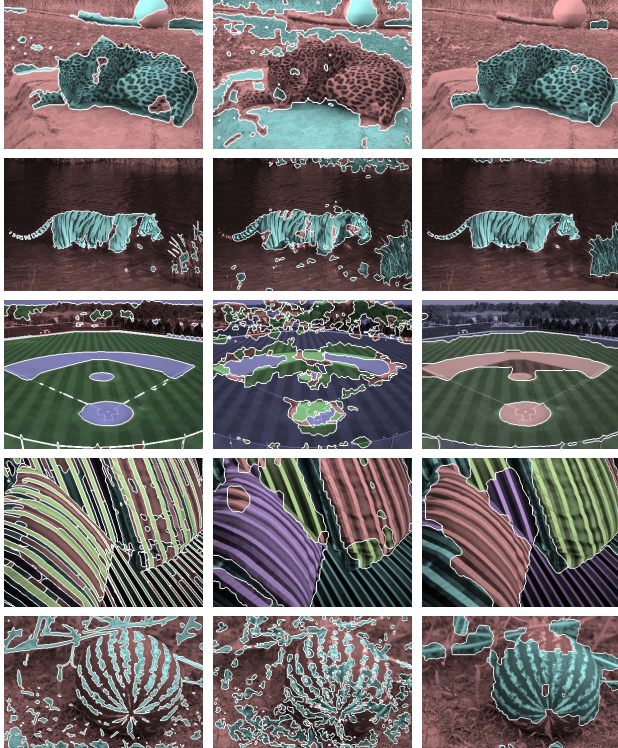


Figure 7. Segmentation results: left column uses [9], middle column uses [7], right column uses our algorithm

penalty favoring the wrong segmentation.

The independent steerable pyramid algorithm does fairly well on the image of pillows because it contains only textures. However, there are small regions within each pillow that are incorrect. In our results, the main error is that the left pillow is split into two regions. The reason for this is that the left pillow has very strong lighting effects and a fast changing scale because it gets very close to the camera. This change is too abrupt for our smooth field to capture.

Our algorithm outperforms the other two in the watermelon image as well. The regions incorrectly identified as part of the watermelon actually resemble the watermelon more than it does the background. The reason our algorithm is not able to capture the bottom of the watermelon is because of the abrupt change in lighting conditions.

6.3. Camera Estimation

Figure 8 shows our results for estimating the camera response. Using a Canon A430, we took multiple photographs of two different scenes under different exposures and used WebHDR¹ to estimate an average polynomial response for the camera. We then found the best possible γ curve in the L2 sense to this polynomial approximation (red squares). Nine pictures containing various textures and

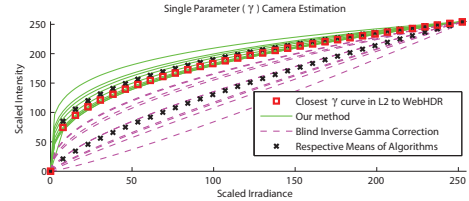


Figure 8. Camera model estimation

lighting condition were then taken. Figure 8 shows the estimated γ values using our method (solid green lines) and the method described in [5] (dotted magenta lines). The mean of the each estimate is shown with X marks. Here we see that the results are superior to that of [5].

6.4. Shading

The incorporation of smooth fields allows for estimation of the radiometric response of the camera. Without the ability to estimate this function, one would not be able to estimate a shading image. Once the camera's radiometric function has been determined, we apply the inverse to the observed image to obtain the irradiance image, shown in Figure 9(a). Given our binary segmentation, we can estimate the reflectance, \mathcal{R} shown in in Figure 9(b), and shading, \mathcal{S} shown in Figure 9(c). For comparison, we also estimate the shading image using the scalar pixel intensity measure of [9] which is shown in Figure 9(d). We give this method the benefit of our estimate of the radiometric response function and our improved segmentation. We do not compare to [7] because the estimation of the shading image produces nonsensical results. The two shading estimates were obtained using the same smoothness constraints, yet there is a considerable difference: undesirable textures of the reflectance image are clearly apparent in the shading image of the scalar case. To validate our shading estimate, we infer the shape from this shading image. Because we are only concerned about the accuracy of our shading estimate and not the segmentation, we extend our shading estimates to the ground truth to avoid incorrect shape artifacts. Once this is obtained, we estimate the shape using Tsai and Shah's algorithm [13] with post Gaussian smoothing. The results of the recovered shape are shown in Figure 10. Our measure is able to recover most of the shape fairly well as compared to using the measure of [9]. The ripples on the back of the zebra using [9] are evidence of an incorrect shading estimate.

There has been extensive work on estimating shape from texture cues such as changes in orientation and scale. Our work realizes that these cues could be due to the texture itself and not the geometric properties of the object. Where typical shape from texture algorithms may fail, our shape from shading from texture does fairly well.

¹WebHDR can be found at <http://luxal.dachary.org/webhdr/>

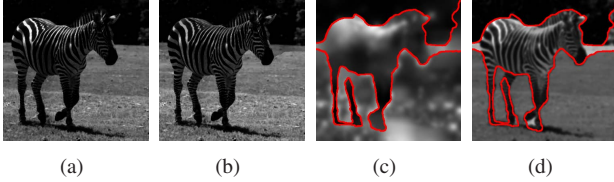


Figure 9. Shading results based on binary segmentation: (a) Estimated irradiance image; (b) Estimated reflectance image; (c) Estimated shading based on our measure; (d) Estimated shading based on scalar feature

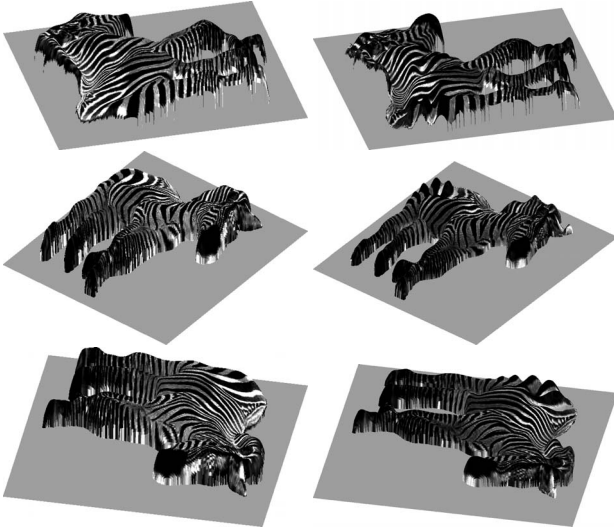


Figure 10. Estimated shape from shading. Left column is obtained using our features, right column is obtained using scalar feature. Observe the undesirable ripple effect on the rear hindquarter of the zebra in the scalar case.

7. Conclusions

We have presented a novel texture measure that captures the contrast, bias, scale, and orientation of a local texture patch. In addition, we imposed smoothness on these parameters which, to our knowledge, has not been considered previously. We showed that our measure is able to segment images with various textures more robustly than other measures. Once segmentation is completed, our model easily extends to estimate a camera’s radiometric response from a single image more accurately than [5]. Lastly, our formulation also allows us to obtain an irradiance, reflectance, and shading image, which we have empirically validated by recovering the shape of the object.

8. Acknowledgments

This research was partially supported by the Air Force Office of Scientific Research under Award No. FA9550-06-1-0324 and the Air Force Research Laboratory under Award No. FA8650-07-D-1220. Any opinions, findings, and conclusions or recommendations expressed in this publication

are those of the authors and do not necessarily reflect the views of the Air Force.

References

- [1] T. Brox and J. Weickert. *Level Set Based Image Segmentation with Multiple Regions*. 2004.
- [2] J. Chang. Extracting orientation and scale from smoothly varying textures with application to segmentation. Master’s thesis, MIT, Cambridge, August 2009.
- [3] T. Chang and C.-C. Kuo. Texture analysis and classification with tree-structured wavelet transform. *Image Processing, IEEE Transactions on*, 2(4):429–441, Oct 1993.
- [4] J. M. H. Du Buf and P. Heitkämper. Texture features based on gabor phase. *Signal Process.*, 23(3):227–244, 1991.
- [5] H. Farid. Blind inverse gamma correction. *Image Processing, IEEE Transactions on*, 10(10):1428–1433, Oct 2001.
- [6] W. Freeman and E. Adelson. The design and use of steerable filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(9):891–906, Sep 1991.
- [7] M. Heiler and C. Schnorr. Natural image statistics for natural image segmentation. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1259–1266 vol.2, Oct. 2003.
- [8] J. Kiefer. Sequential minimax search for a maximum. *Proc. Amer. Math. Soc.* 4, pages 502–506, 1953.
- [9] J. Kim, J. Fisher III, A. Yezzi, M. Cetin, and A. Willsky. A nonparametric statistical method for image segmentation using information theory and curve evolution. *Image Processing, IEEE Transactions on*, 14(10):1486–1502, Oct. 2005.
- [10] J. A. Montoya-Zegarra, N. J. Leite, and R. d. S. Torres. Rotation-invariant and scale-invariant steerable pyramid decomposition for texture image retrieval. In *SIBGRAP ’07: Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing*, pages 121–128, Washington, DC, USA, 2007. IEEE Computer Society.
- [11] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *Int. J. Comput. Vision*, 40(1):49–70, 2000.
- [12] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, April 1986.
- [13] P. sing Tsai and M. Shah. Shape from shading using linear approximation. *Image and Vision Computing*, 12:487–498, 1994.
- [14] A. Teuner, O. Pichler, J. Santos Conde, and B. Hosticka. Orientation- and scale-invariant recognition of textures in multi-object scenes. *Image Processing, 1997. Proceedings., International Conference on*, 3:174–177 vol.3, Oct 1997.
- [15] W. Wells III, W. Grimson, R. Kikinis, and F. Jolesz. Adaptive segmentation of MRI data. *Medical Imaging, IEEE Transactions on*, 15(4):429–442, Aug 1996.
- [16] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 664–671 vol.1, 2003.