

Supplementary: Learning Discriminative Illumination and Filters for Raw Material Classification with Optimal Projections of Bidirectional Texture Functions

Chao Liu, Gefei Yang, Jinwei Gu
Rochester Institute of Technology

1. Algorithm for Optimizing Discriminative Illumination \mathbf{w} and Filters \mathbf{W}

In Section 3 of the main paper, we showed that Equation (5) is used to solve for the optimal discriminative illumination \mathbf{w} and filters \mathbf{W} by solving the following optimization problem:

$$\max_{\mathbf{W}, \mathbf{w}} J = \frac{\text{Trace}(\mathbf{S}_b)}{\text{Trace}(\mathbf{S}_w)}, \quad \text{st.} \quad \|\mathbf{w}\| = 1, \quad (1)$$

where

$$\begin{aligned} \mathbf{S}_b &= \sum_{c=1}^C (\bar{\mathbf{r}}_c - \bar{\mathbf{r}})(\bar{\mathbf{r}}_c - \bar{\mathbf{r}})^T, \\ \mathbf{S}_w &= \sum_{c=1}^C \sum_{i=1}^{N_c} (\mathbf{r}_{i,c} - \bar{\mathbf{r}}_c)(\mathbf{r}_{i,c} - \bar{\mathbf{r}}_c)^T. \end{aligned}$$

$\mathbf{r}_{i,c}$ is the texture descriptor for the i -th sample in the c -th class:

$$\mathbf{r} = \mathbf{W}^T \mathbf{B}^T \mathbf{F}^T \mathbf{w} = \mathbf{W}^T \mathbf{R}^T \mathbf{w}. \quad (2)$$

N_c is the number of samples in the c -th class. $\bar{\mathbf{r}}_c$ is the average descriptor for the c -th class, and $\bar{\mathbf{r}}$ is the average descriptor for all classes.

In this section, we show the detailed algorithm for solving this problem. We optimize \mathbf{w} and \mathbf{W} in Equation (1) alternatively. If we fix \mathbf{W} , Equation (1) becomes

$$\max_{\mathbf{w}} \mathbf{J} = \frac{\text{Trace}(\mathbf{w}^T \mathbf{S}_1 \mathbf{w})}{\text{Trace}(\mathbf{w}^T \mathbf{S}_2 \mathbf{w})}, \quad \text{st.} \quad \|\mathbf{w}\| = 1,$$

where

$$\begin{aligned} \mathbf{S}_1 &= \sum_{c=1}^C (\bar{\mathbf{R}}_c - \bar{\mathbf{R}}) \mathbf{W} \mathbf{W}^T (\bar{\mathbf{R}}_c - \bar{\mathbf{R}})^T, \\ \mathbf{S}_2 &= \sum_{c=1}^C \sum_{i=1}^{N_c} (\mathbf{R}_{i,c} - \bar{\mathbf{R}}_c) \mathbf{W} \mathbf{W}^T (\mathbf{R}_{i,c} - \bar{\mathbf{R}}_c)^T, \end{aligned}$$

where $\mathbf{R}_{i,c}$ is the corresponding matrix for the i -th sample in the c -th class, $\bar{\mathbf{R}}_c$ is the average matrix for the c -th class,

and $\bar{\mathbf{R}}$ is the average matrix for all the classes. This is the well-known **Rayleigh quotient** problem and can be solved by eigenvalue decomposition in a closed form.

If we fix \mathbf{w} , Equation (1) becomes

$$\max_{\mathbf{W}} \mathbf{J} = \frac{\text{Trace}(\mathbf{W}^T \mathbf{S}_3 \mathbf{W})}{\text{Trace}(\mathbf{W}^T \mathbf{S}_4 \mathbf{W})},$$

where

$$\begin{aligned} \mathbf{S}_3 &= \sum_{c=1}^C (\bar{\mathbf{R}}_c - \bar{\mathbf{R}})^T \mathbf{w} \mathbf{w}^T (\bar{\mathbf{R}}_c - \bar{\mathbf{R}}), \\ \mathbf{S}_4 &= \sum_{c=1}^C \sum_{i=1}^{N_c} (\mathbf{R}_{i,c} - \bar{\mathbf{R}}_c) \mathbf{w} \mathbf{w}^T (\mathbf{R}_{i,c} - \bar{\mathbf{R}}_c)^T. \end{aligned}$$

We follow the method presented in [4] to solve this **trace ratio** problem, which is an iterative algorithm that guarantees global optimal solution. Let $\mathbf{S}_p = \mathbf{S}_3$ and $\mathbf{S}_t = \mathbf{S}_3 + \mathbf{S}_4$. It is not hard to show that both \mathbf{S}_p and \mathbf{S}_t are positive semidefinite. Assuming the singular value decomposition of \mathbf{S}_t is $\mathbf{S}_t = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T$, the above trace ratio optimization problem becomes:

$$\max_{\mathbf{W}} \mathbf{J} = \frac{\text{Trace}(\mathbf{W}^T \mathbf{S}_p^u \mathbf{W}^T)}{\text{Trace}(\mathbf{W}^T \mathbf{S}_t^u \mathbf{W}^T)}, \quad (3)$$

where $\mathbf{S}_p^u = \mathbf{U} \mathbf{S}_p \mathbf{U}^T$ and $\mathbf{S}_t^u = \mathbf{U} \mathbf{S}_t \mathbf{U}^T$.

Then ITR (Iterative algorithm for the Trace Ratio optimization problem):

1. Initialize \mathbf{W}^0 as an arbitrary columnly orthogonal matrix;
2. If iteration number $<$ max iteration number, do:
 - 2.1 Compute the trace ratio λ_n for the projection matrix \mathbf{W}_{n-1} estimated from the previous iteration:

$$\lambda_n = \frac{\text{Trace}(\mathbf{W}_{n-1}^T \mathbf{S}_p^u \mathbf{W}_{n-1})}{\text{Trace}(\mathbf{W}_{n-1}^T \mathbf{S}_t^u \mathbf{W}_{n-1})}$$

- 2.2 Construct the trace difference problem:

$$\max_{\mathbf{W}^T \mathbf{W} = \mathbf{I}} d = \text{Trace}(\mathbf{W}^T (\mathbf{S}_p^u - \lambda_n \mathbf{S}_t^u) \mathbf{W})$$

2.3 Solve the trace difference problem using eigenvalue decomposition:

$$(\mathbf{S}_p^u - \lambda_n \mathbf{S}_t^u) \mathbf{v}_k^n = \tau_k^n \mathbf{v}_k^n, \quad (4)$$

where τ_k^n is the k -th largest eigenvalue of $\mathbf{S}_p^u - \lambda_n \mathbf{S}_t^u$ corresponding to the eigenvector \mathbf{v}_k^n .

2.4 Set $\mathbf{W}_n = [\mathbf{v}_1^n, \mathbf{v}_2^n, \dots, \mathbf{v}_d^n]$. d is the desired lower feature dimension.

2.5 Set $\mathbf{S}_t^v = \mathbf{W}_n (\mathbf{W}_n)^T \mathbf{S}_t^u \mathbf{W}_n (\mathbf{W}_n)^T$

2.6 Perform singular value decomposition:

$$\mathbf{S}_t^v = \mathbf{W}_n \mathbf{\Lambda}^n \mathbf{W}_n^T \quad (5)$$

2.7 If $\|\mathbf{W}_n - \mathbf{W}_{n-1}\| < \sqrt{m'} d \epsilon$, then break. ϵ is set to 10^{-4} in this work.

3. Output $\mathbf{W} = \mathbf{W}_n$.

Thus, in our algorithm, we iteratively solve the Rayleigh quotient problem and the trace ratio problem to find optimal \mathbf{w} and \mathbf{W} . In all the experiments, we found the algorithm will converge within 3 to 5 iterations. Please refer to the supplemental video for a video of iterative optimization.

2. The Effect of Filter Patch Size and Number of Filters

We also did extensive simulation to evaluate the effect of filter patch size and the number of filters learned to the BTF classification performance.

To evaluate the effect of the filter size, we trained five filter banks with different filter sizes to classify between aluminum and stainless and test the performances of these filter banks. In Fig. 1, we show the trained filters and their classification rates. On one hand, as the filter size increases, the patterns of some filters changes in a way different from scaling for the filters with higher spatial frequencies. On the other hand, the scaling on filter patterns is obvious for the filters with lower spatial frequencies. This is because more pixels are included in the training sets as the filter size increases. The filters with higher frequencies are more sensitive to the changes of training set while the filters with lower frequencies are more stable to including more training samples, which can be thought of adding the identical samples (the repetitive patterns in texture) plus high frequency noise (the variance of patterns within a texture). Also, the performance does not increase with the filter size monotonically. This is due to two reasons. First, the texture consists of repetitive patterns. So increasing the filter size does not necessarily include more informations about the texture. Second, as the filter size increases, it is more likely to include outliers, such as the specular lobes, into the training set. In all the experiments, we set the patch size to be 19×19 .

In addition to the filter patch size, the number of filters to optimize is another important parameter for the proposed

filter-based BTF classification method. Figure 2 shows the trained filter banks with different number of filters for classifying aluminum and stainless. As shown, the performance for this task increases fast with the number of filters. This indicates that the classification of aluminum and stainless can be performed well on a subspace of BTF with lower dimensionality. This corresponds to the observation in [3] that some texture classification tasks can be performed well even though the sampling patch size is small (*i.e.*, using more local feature). Within each filter bank, the spatial frequency of the learned filter increases with the index of filter. This indicates that the difference of the projection of BTF is concentrated on the low spatial frequencies. This is reasonable since the projection vector of BTF (optimal light pattern \mathbf{w}) is optimized so that the distance of within-class projections is maximized. In all the experiments, we set the number of filters to optimize to be 16.

3. Comparisons and visualization of different methods

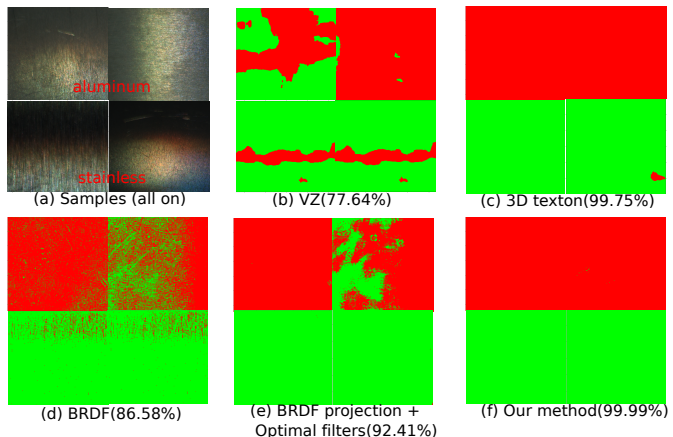


Figure 3: The classification for aluminum and stainless steel samples. (a) Images of samples when all LEDs are turned on; (b) VZ classifier[3]; (c) 3D texton[2]; (d) BRDF projection[1]; (e) BRDF projection coupled with optimal filters; (f) Our method. The accuracy is shown in the bracket.

As discussed in the paper, since the VZ classifier and 3D texton are both bag-of-words methods, we also evaluate them with different numbers of words. We found their performance is sensitive to this parameter which makes them less robust compared to our method. The comparisons are shown in Table. 1. Fig. 3 shows the classification results for some samples of aluminum and stainless steel. We found our method has quite close performance compared to 3D texton by using only a single coded image.

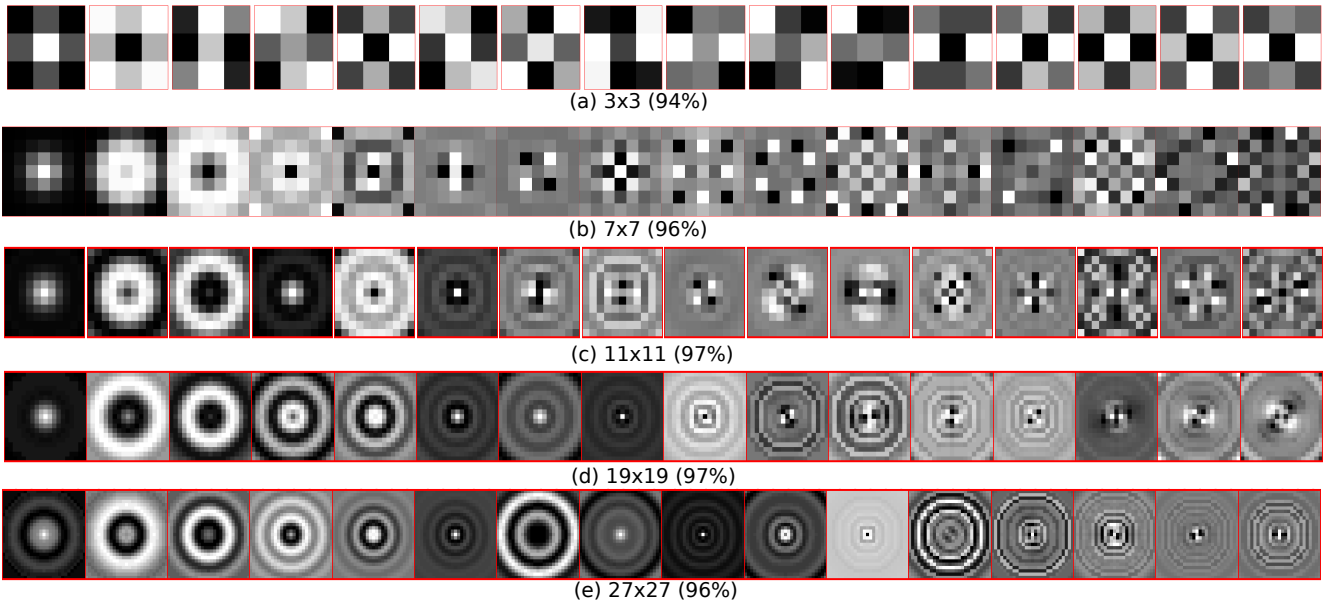


Figure 1: The trained filter banks with different filter sizes. From top to bottom, the filter sizes are: 3×3 , 7×7 , 11×11 , 19×19 and 27×27 , with the classification rate for the task aluminum vs. stainless shown to the bottom of each filter bank. The corresponding filters, shown in the same column, are not necessarily the scaled versions of each other due to two reasons: 1) increasing the filter size does not necessarily include more informations about the texture due to the repetition of patterns. 2) As the filter size increases, it is more likely to include outliers, such as the specular lobes, into the training set.

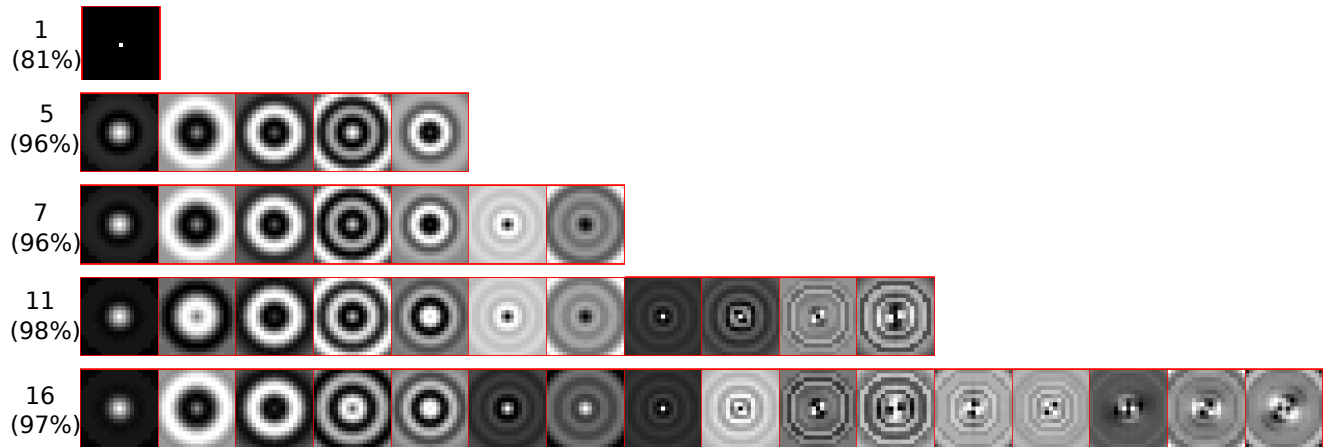


Figure 2: The filters in the optimal filter banks with different number of filters. Shown on the left side are the number of filters in the filter bank and the classification rate for the task aluminum vs. stainless steel.

References

- [1] J. Gu and C. Liu. Discriminative illumination: Per-pixel classification of raw materials based optimal projections of spectral brdfs. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012. 2, 4
- [2] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision (IJCV)*, 43(1):29–44, June 2001. 2, 4
- [3] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision (IJCV)*, 62:61–81, 2005. 2, 4
- [4] H. Wang, S. Yan, D. Xu, X. Tang, and T. Huang. Trace ratio vs. ratio trace for dimensionality reduction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 1

Table 1: Comparison results with the VZ classifier [3], 3D Texton [2], the BRDF projection method [1] and the BRDF projection coupled with optimal filters. The number of required images for each method is shown in the bracket. 1* means it uses the subtraction of two images since light cannot be negative. For the VZ classifier and 3D Texton Method, we test with different numbers of words (K).

| Task | VZ (1) [3], $K = 100$ | VZ (1), $K = 500$ | 3D Texton (150)[2], $K = 100$ | 3D Texton (150), $K = 500$ | BRDF Projection (1*) [1] | BRDF projection + Optimal filters (1*) | Our method (1*) |
|------------------------------|-----------------------|-------------------|----------------------------------|-------------------------------|-----------------------------|---|-----------------|
| Aluminum vs. Granite | 79.12% | 81.25% | 100% | 100% | 94.03% | 93.37% | 98.55% |
| Aluminum vs. Stainless | 84.81% | 86.36% | 99.40% | 99.40% | 86.61% | 89.97% | 96.12% |
| Aluminum vs. Wood | 81.50% | 83.55% | 81.75% | 83.39% | 99.56% | 99.99% | 100% |
| Carpet vs. Wood | 80.59% | 79.96% | 86.65% | 87.28% | 96.09% | 95.04% | 99.04% |
| Carpet vs. Paper | 85.28% | 83.15% | 91.48% | 89.37% | 93.61% | 99.98% | 100% |
| Granite vs. Paper | 86.83% | 88.29% | 100% | 100% | 97.32% | 99.99% | 99.74% |
| Paper vs. Wood | 73.84% | 73.99% | 89.81% | 91.20% | 99.20% | 100% | 100% |
| Plastic vs. Stainless | 78.55% | 79.86% | 86.97% | 88.54% | 99.77% | 100% | 100% |
| Paper vs. Aluminum | 85.63% | 86.90% | 99.25% | 99.34% | 99.01% | 99.08% | 100% |