# 1-dimensional line profile analysis

Taeg Sang Cho
Updated July 26 2010

In this report, we analyze regularities in 1-dimensional line profiles. Extracted regularity can serve as a prior on line profiles.

Here's the action plan. We first analyze line profiles taken from images that conform to popular image models: a Gaussian noise image model and a fallen-leaves model. Insights from analyzing these images will help us understand natural images better.

## 1. A Gaussian noise image

We render 30 images with Gaussian noise (std = $1\%$), take line profiles of noise images (each slice 25 pixels long) at random locations, and use PCA and K-SVD to find their basis functions. We randomly select the orientation of each slice.

Fig. 1 shows the principal components, sorted in the descending order of signal variance each captures. We show principal components that capture $99\%$ of signal variance. This experiment shows that we need 25 principal components to capture $99\%$ of signal variance. This result is plausible since a 25-pixel Gaussian sample would occupy a 25-dimensional ball, and therefore we need all 25 basis functions (i.e. principal components) to represent such 25-dimensional space. From this argument, we would also guess that the signal variance captured by each component would be the same, but as Fig. 1 shows, it's not. We attribute this to interpolation. When we extract line slices, we interpolate neighboring pixels to find off-grid pixel values. Such interpolation introduces correlations in neighboring pixels, which allows for compaction. [1] Note that principal components resemble Fourier bases: we have both low-frequency and high frequency components only moderate localized structures.

Fig. 2 shows basis functions extracted using K-SVD. A brief introduction to K-SVD is provided in Appendix A; for a comprehensive introduction, please refer to Aahron *et al.*, " K-SVD: design of dictionaries for sparse representation". In this experiment, we set $K = 100, T = 1$. When $T = 1$, each training sample is represented with only one basis function: in this case, a basis function that has the highest correlation is chosen to represent each training sample.

In Fig. 2, we sort the basis functions according to the number of times each basis function is used to represent training samples. This is a heuristic to quantify which basis function captures the most signal variance. Even with the K-SVD algorithm, there is not a particular structure in Gaussian noise images. This is expected as each noise sample is independent (up to interpolation).

[1] We have verified through experiment that when we do not perform any interpolation, the variance captured by each principal component is the same.

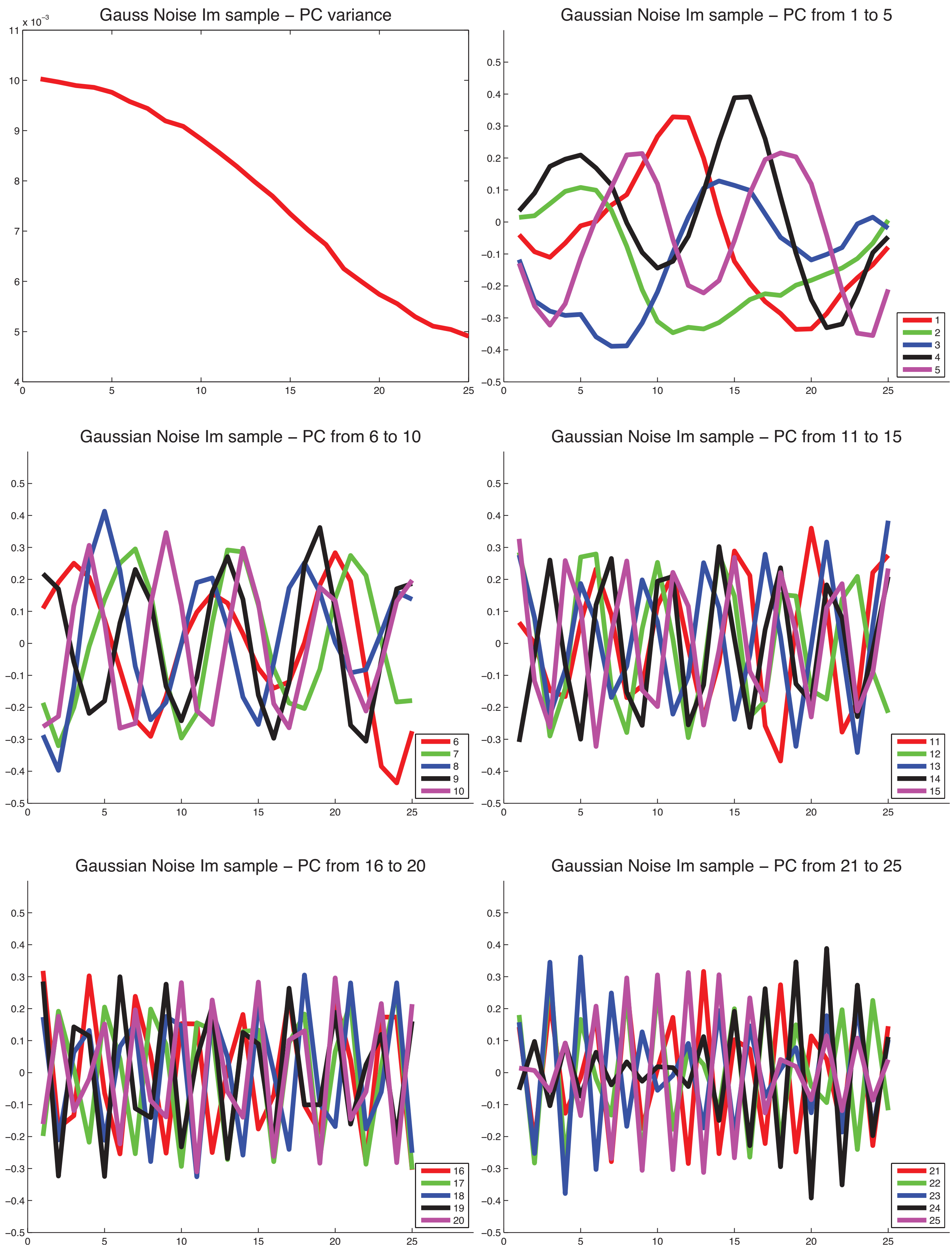


**Figure 1:** *Principal components of line profiles from Gaussian noise images.*

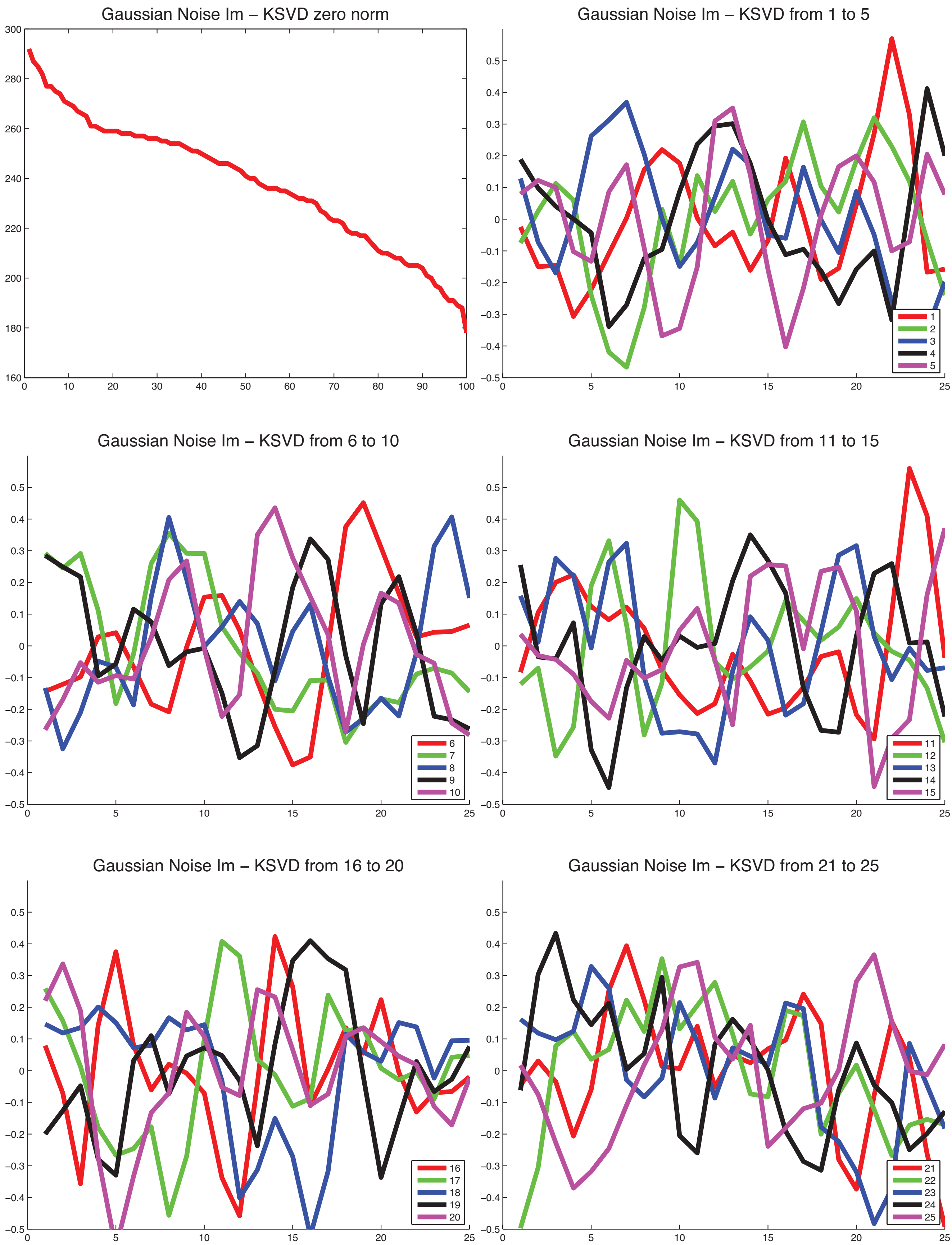


**Figure 2:** *K-SVD basis functions of line profiles from Gaussian noise images.*

## 2. A fallen-leaves image

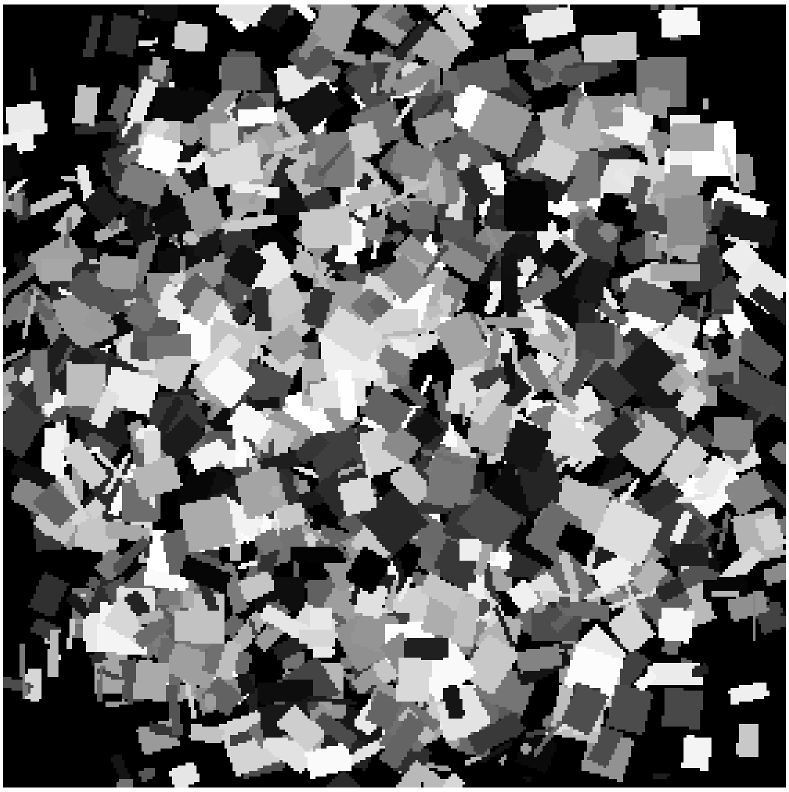

**Figure 3:** *One of the rendered fallen-leaves images.*

Here, we analyze line profiles from fallen-leaves images. We render 5 images with occluding squares as shown in Fig. 3. The width and height of each square is randomly chosen with the maximum length of 25 pixels, and the orientation of each square is also randomly chosen. The maximum length changes the characteristics of rendered images. When the maximum size of the square is significantly larger than the line profile length (25 pixels in our case), extracted line slices consist mostly of step edges. On the other hand, when the maximum length is significantly less than the line profile length, the rendered image has characteristics similar to a Gaussian noise image. To balance these two characteristics, we set the maximum length of the squares to be equal to the line profile length.

We take slices of rendered images at detected edges. We use two different edge detection algorithms: a canny-edge detector and a steerable filter based detector. Different edge detection algorithms key on different structures, therefore we analyze edge profiles from both edge detectors. To account for phase at edges, we enforce that the edge profiles' center of mass lie on the right-half of the profiles. In other words, if the center of mass is on the left-half of the profile, we flip the profile.

### 2.1. Edges from the canny edge detector

We show the principal components in Fig. 10. We only need 20 components to capture 99% of signal variance, therefore the PCA is able to extract some regularities in fallen-leaves images. The top 10 principal components exhibit localized structures that resemble edge profiles. However, if we look at the components with small variance, we get basis functions similar to Fourier bases.

More interesting are basis functions from K-SVD, shown in Fig. 11. We preprocess the line profiles such that each line profile has zero mean. Empirically, this allows us to exact more meaningful structures. In Fig. 11, we sort the basis functions according to the number of times each basis function is used to represent training samples, and we shot the top 25 basis functions. Interestingly, the top 5 basis functions are center-shifted step edges, and as we go down the list, we observe line structures with varying widths. This shows that the fallen leaves images have lines with varying widths. Although not shown here, the 25 basis functions with smallest zero norms are very similar to random basis functions as shown in Fig. 2.

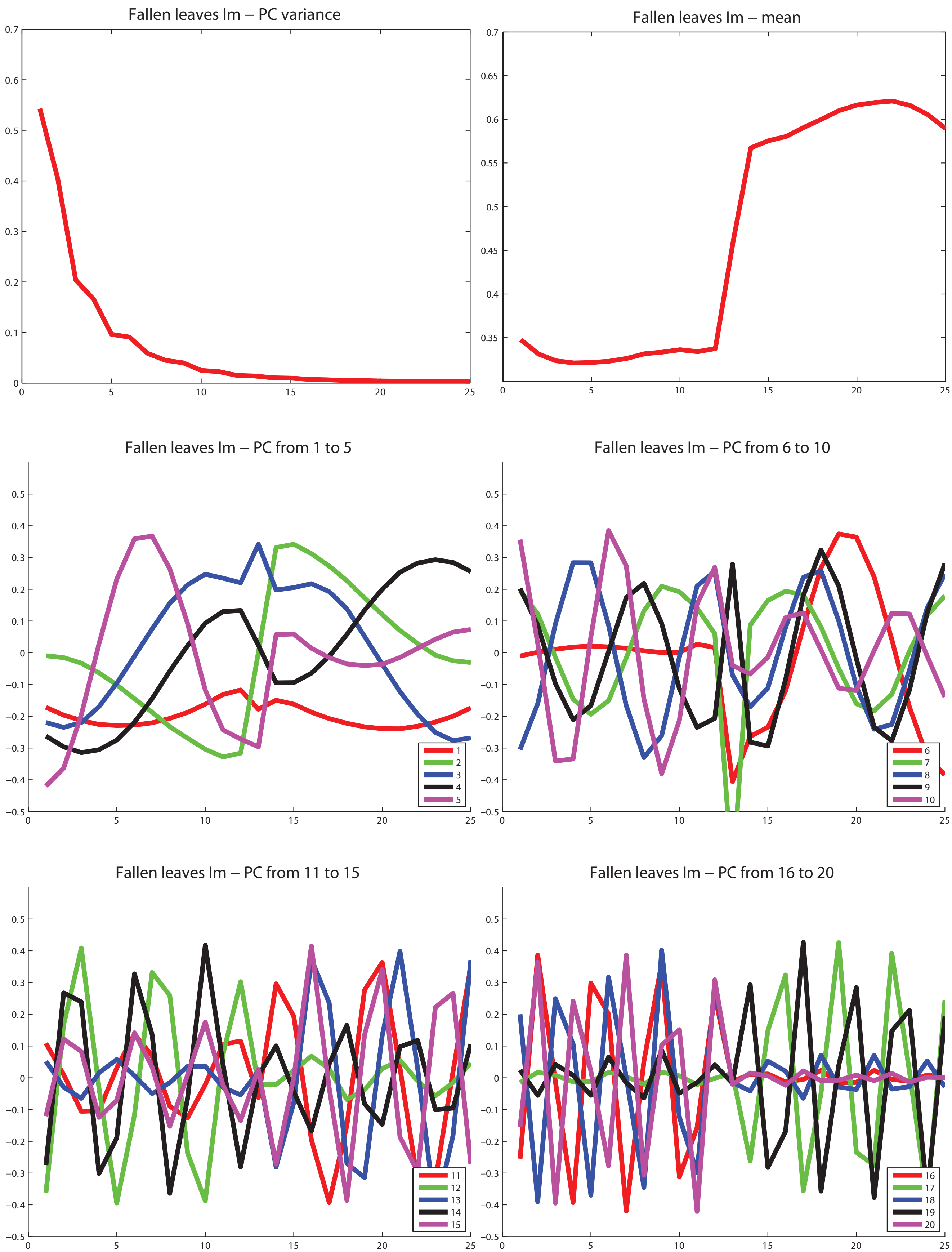


**Figure 4:** *Principal components of line profiles from fallen-leaves images. We used the canny edge detector to detect edges.*

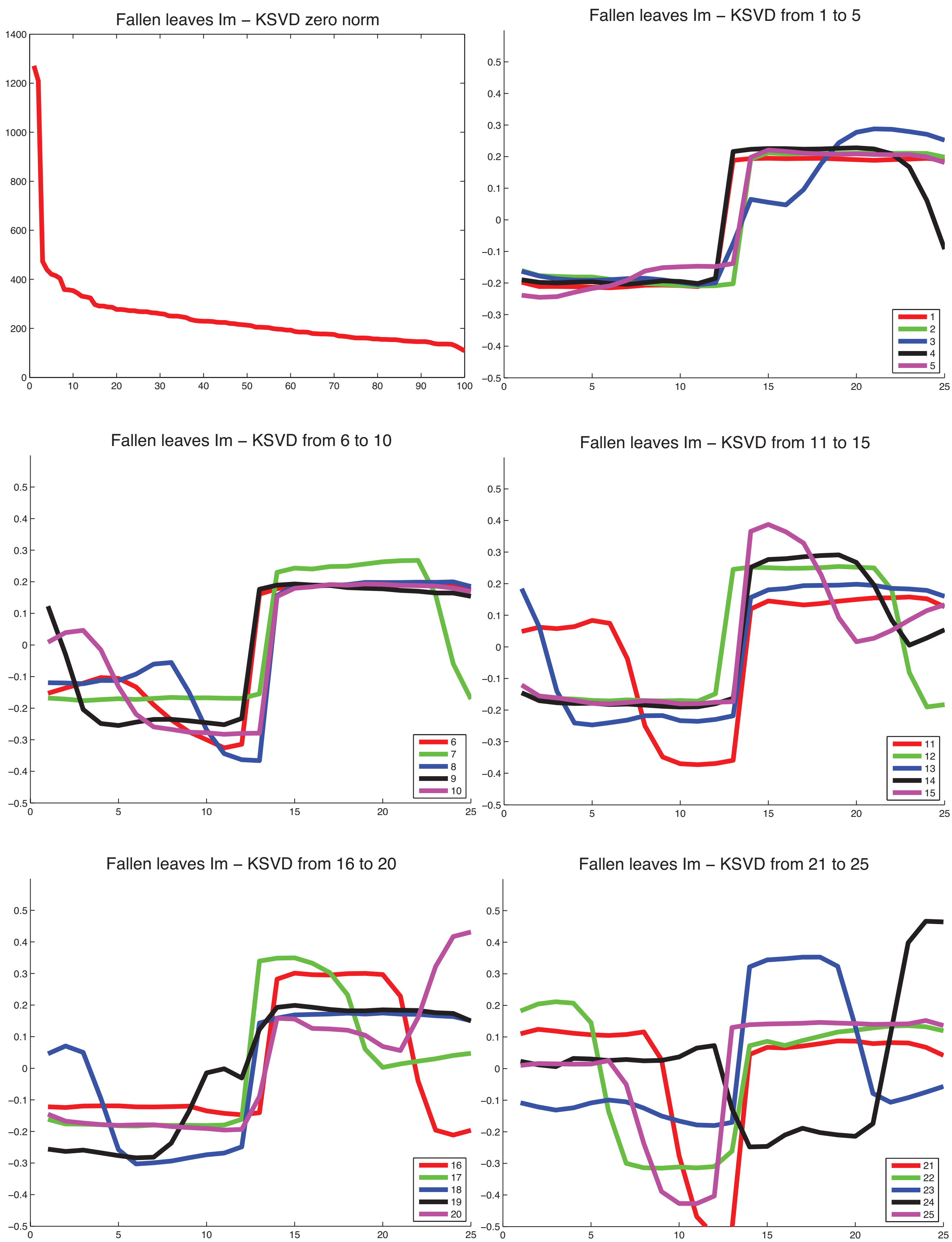


**Figure 5:** *K-SVD basis functions of line profiles from fallen-leaves images. We used the canny edge detector to detect edges.*

### 2.2. Edges from the steerable filters

To detect edges with the steerable filters, we convolve the image with G2a, G2b, G2c filters from Freeman and Adelson, “The design of steerable filters”, and steer them according to local edge orientation. Then we run a nonmaximal suppression algorithm to keep only one edge sample per edge, and threshold the gradient image to retain edges with the top 5% of gradient magnitude.

Even with this new edge detector, we observe characteristics similar to the basis functions we extracted using the canny edge detector. One difference is that the steerable filter fires even for a relatively large line structure in the middle, such as the 5th and 10th basis functions in Fig. 7.

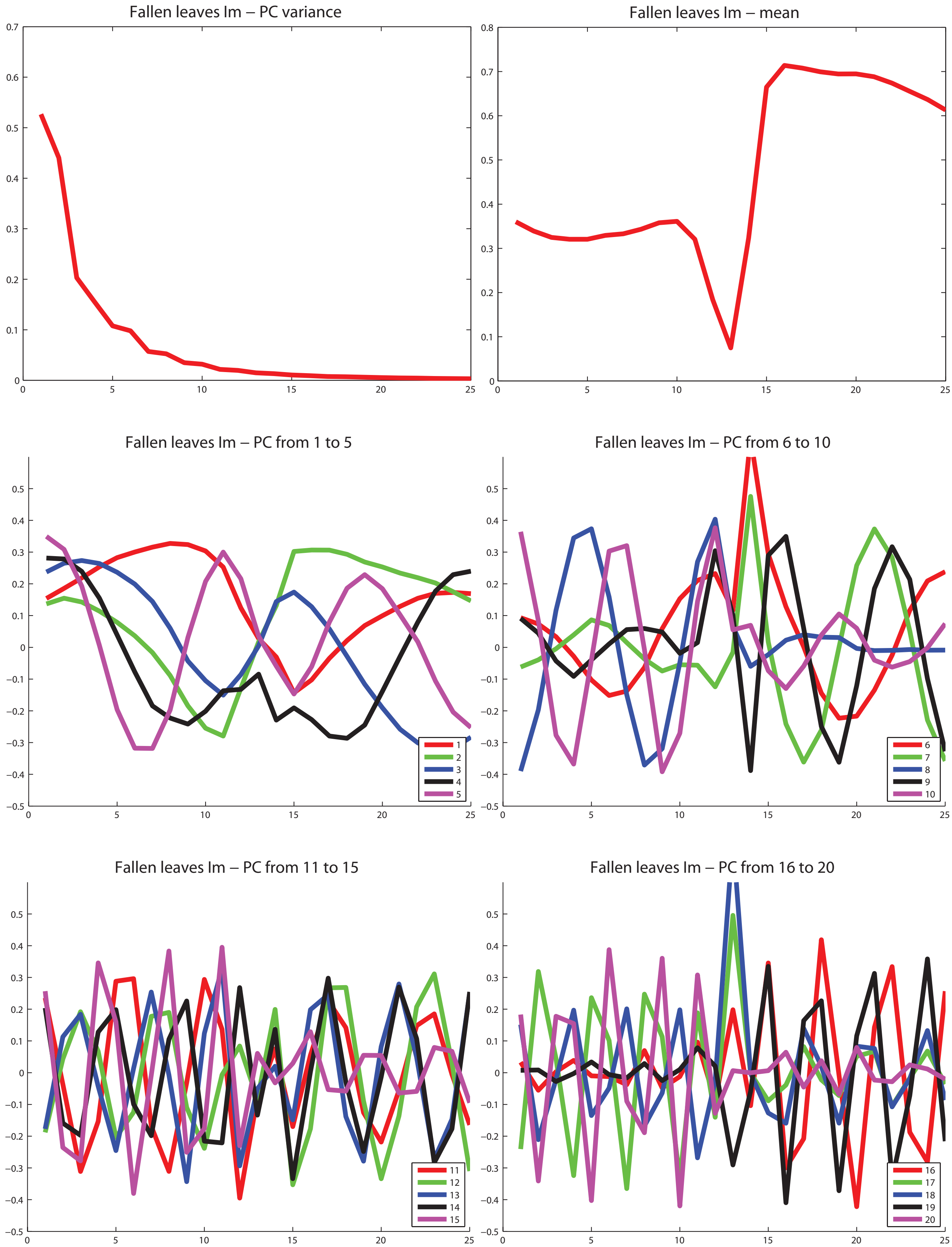


**Figure 6:** *Principal components of line profiles from fallen-leaves images. We used steerable filters to detect edges.*

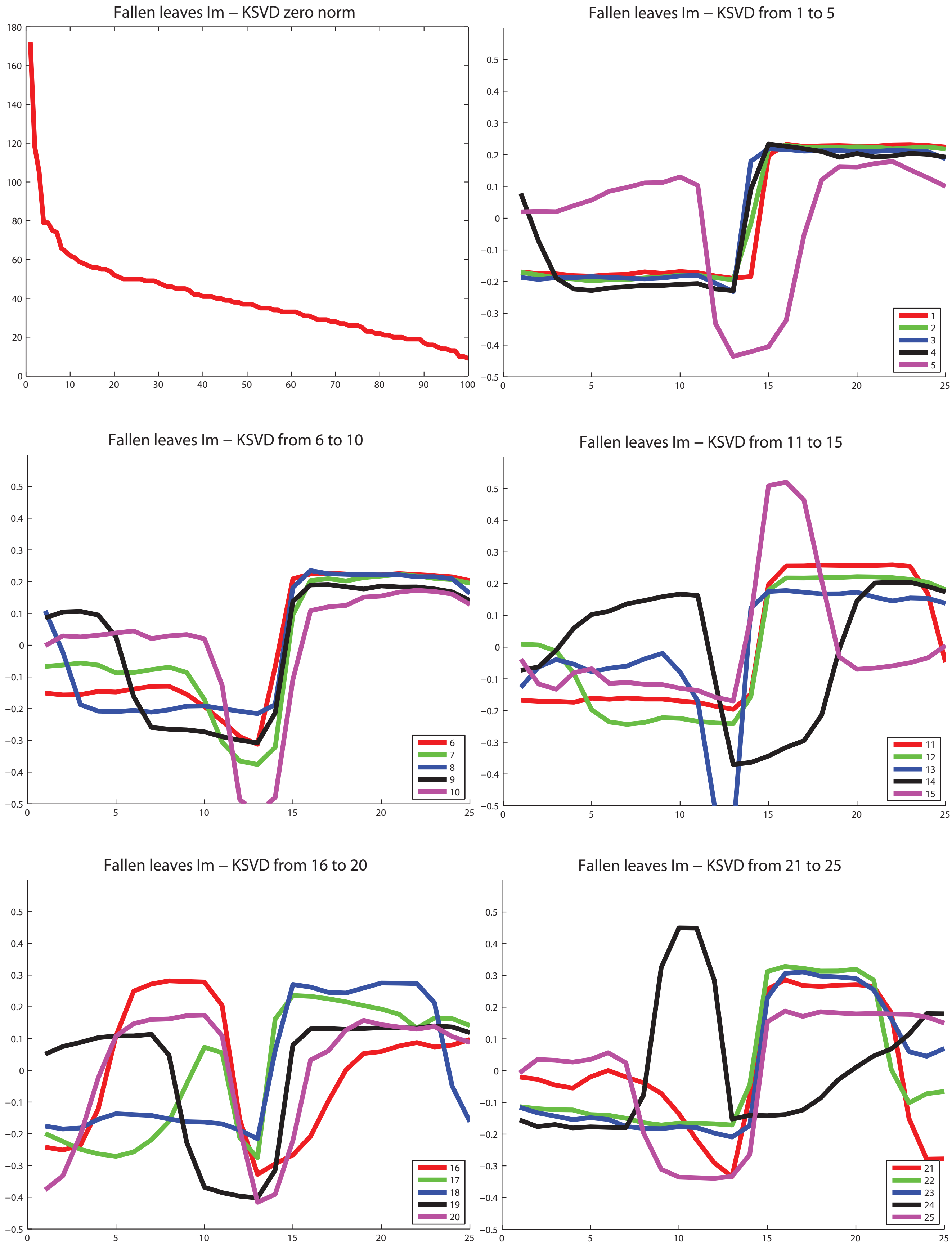


**Figure 7:** *K-SVD basis functions of line profiles from fallen-leaves images. We used steerable filters to detect edges.*

## 3. Natural images

We perform the same set of experiments on natural images. We detect edges using two different edge detectors, take line profiles at edges, and analyze them using PCA and K-SVD.

### 3.1. Edges from the canny edge detector

As Fig. 10 shows, it only takes about 15 principal components to capture 99% of signal variance. This suggests that edges have quite regular structures. Principal components do have spatial structures, but not as salient as K-SVD basis functions shown in Fig. 11. The top 5 K-SVD basis functions correspond to step edges that are spatially shifted and that are rising at different slopes. In other words, K-SVD extracts spatial shifts as well as scale regularity. We observe more spatial regularities as we consider basis functions with small zero-norm; many of these profiles correspond to lines with different widths. Many of these regularities resemble basis functions from fallen-leaves images. When we look at basis functions that are much less used, they correspond to very high frequencies, similar to those of Gaussian noise images. This suggests that natural images consist of piecewise smooth regions with Gaussian noise-like texture.

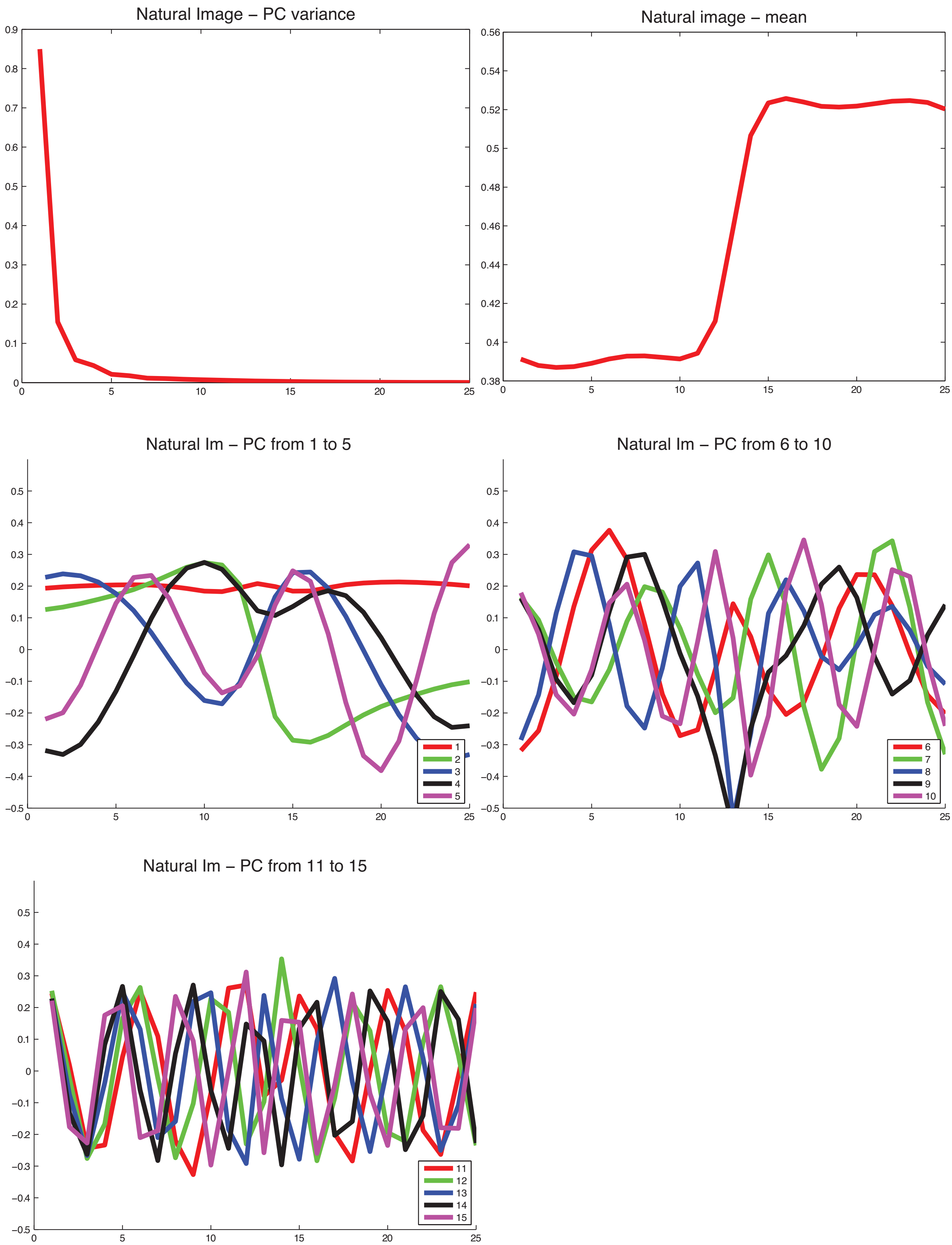


**Figure 8:** *Principal components of line profiles from natural images. We used the canny edge detector to detect edges.*

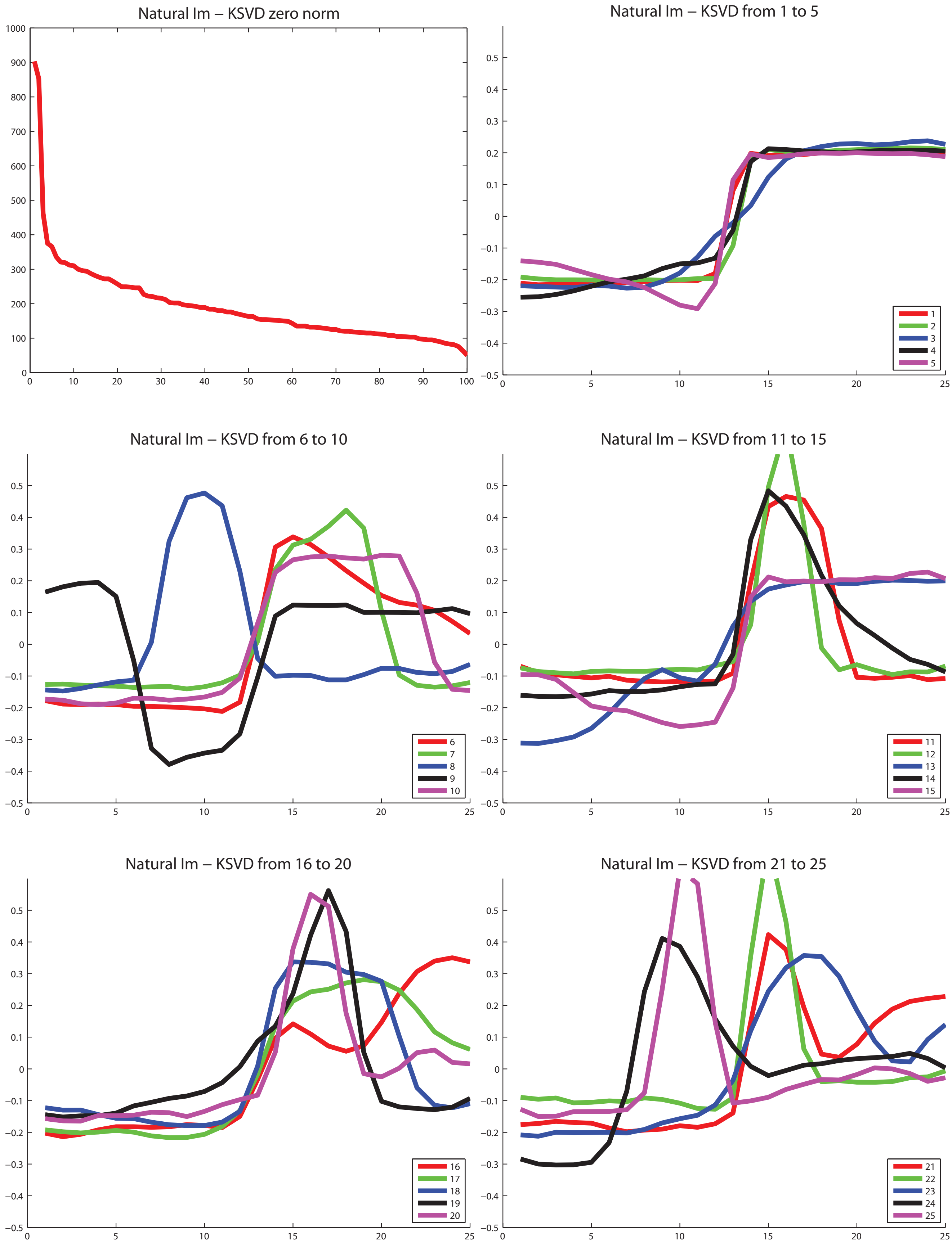


**Figure 9:** *K-SVD basis functions of line profiles from natural images. We used the canny edge detector to detect edges.*

### 3.2. Edges from steerable filters

Similar conclusions can be drawn from edges detected using steerable filters.

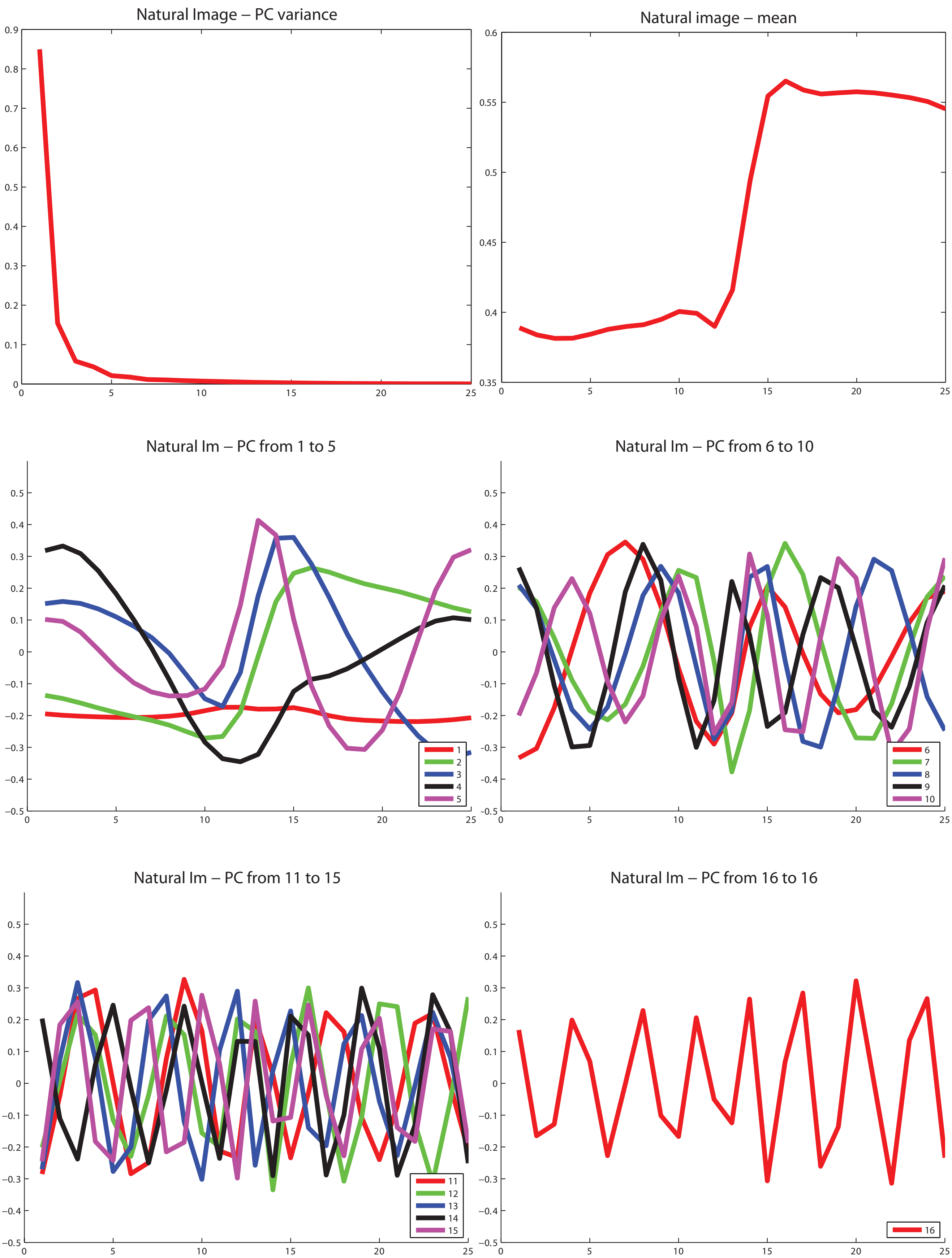


**Figure 10:** *Principal components of line profiles from natural images. We used steerable filters to detect edges.*

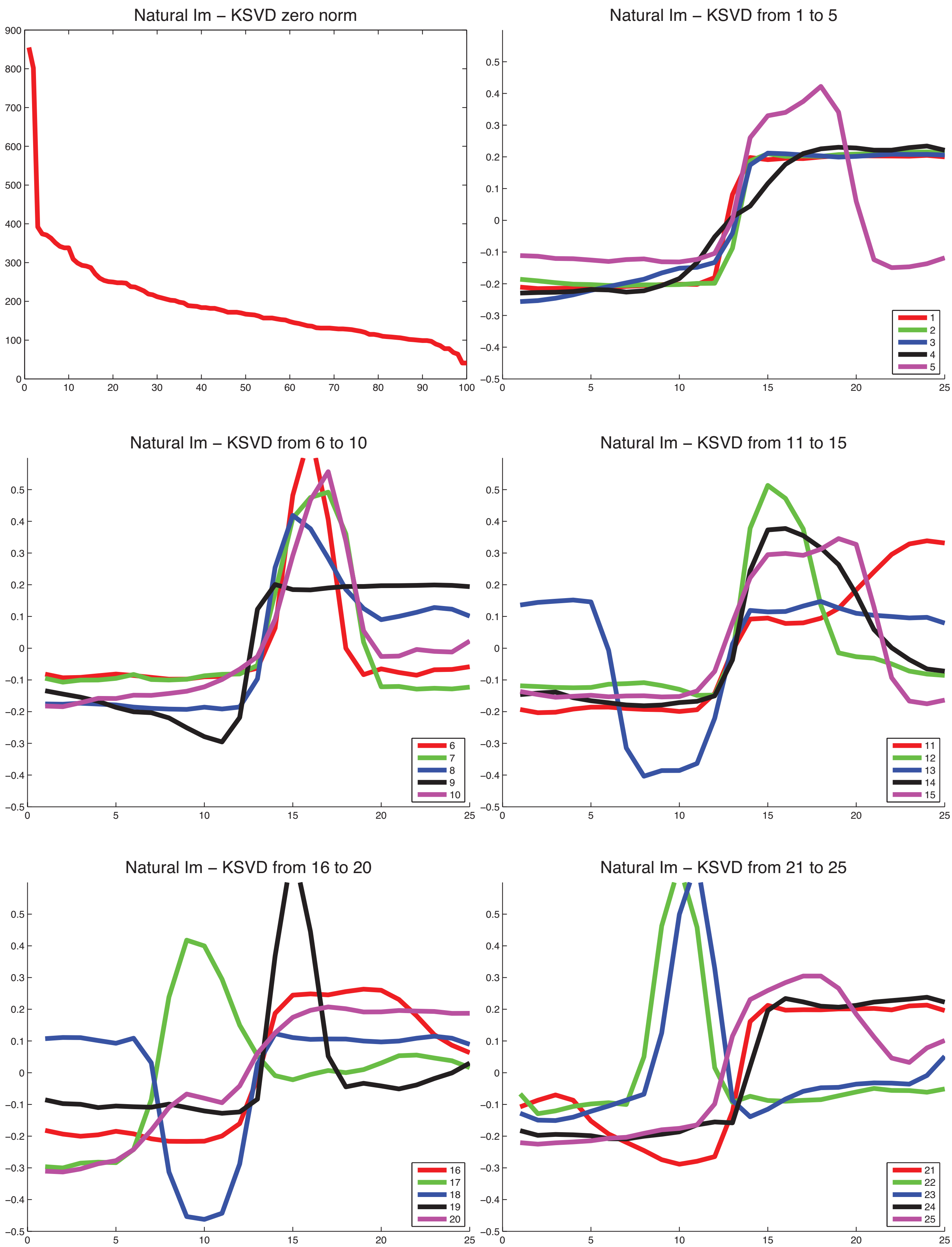


**Figure 11:** *K-SVD basis functions of line profiles from natural images. We used steerable filters to detect edges.*

## Appendix A

K-SVD attempts to find a set of basis functions that would allow a sparse representation of the training data $Y$. Let $Y$ be a $M \times N$ matrix where $M$ is the signal dimension and $N$ is the number of training samples. K-SVD minimizes the following objective function to find a dictionary $D$ and a sparse representation of $Y$:

$$\begin{aligned} [\hat{D}, \hat{X}] = \operatorname{argmin}_{D,X} \|Y - DX\|^2 \\ s.t. \|x_j\|_0 \leq T \end{aligned} \tag{1}$$

where $T$ is the maximum number of non-zero element in a sparse representation $x_j$ of a $j^{th}$ training sample $y_j$. There is another parameter $K$, which is the number of basis functions, i.e. $D$ is a $M \times K$ matrix.