Textons

Malik&Belongie ICCV1999, Leung&Malik, IJCV 2001, Nava&Kybic

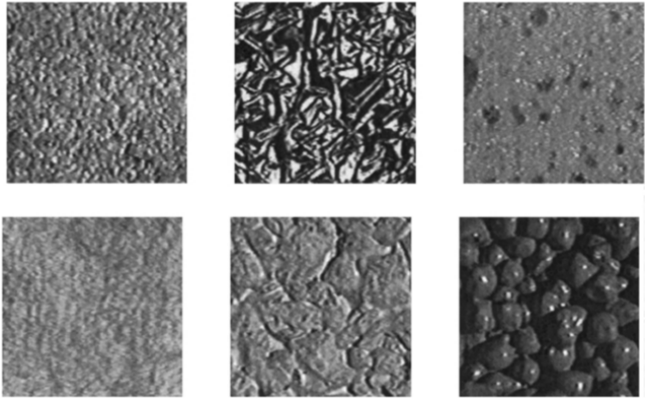
Jan Kybic

2020

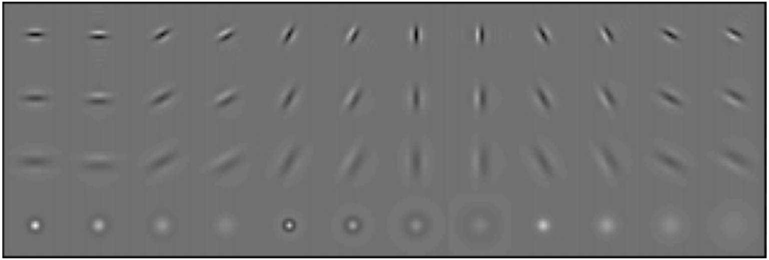
Textons

- Extract patches
- Build a dictionary of patches (textons) by k-means clustering
- Characterize new textures using this dictionary
- Originally 2D, extended to 3D

Natural textures



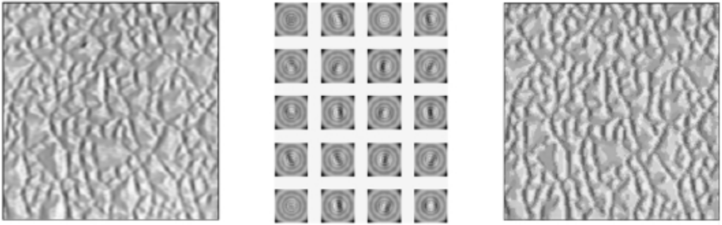
Filterbank



- 48 filters: 36 oriented filters, with 6 orientations, 3 scales, and 2 phases, 8 center-surround derivative filters and 4 low-pass Gaussian filters, DoG, normalized, size 3-28 pixels

Descriptors

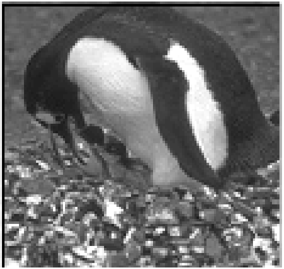
- descriptors $d_i = \tilde{g}_i * f$, $d_i(x) = \langle g_i(t-x), f \rangle$, $\boldsymbol{d}(\boldsymbol{x}) = \boldsymbol{G}\boldsymbol{f}(\boldsymbol{x})$
- k-means clustering of $\boldsymbol{d}(\boldsymbol{x}) \rightarrow$ *appearance vectors* $\boldsymbol{c}_k$
- quantize $\boldsymbol{d}$using $\boldsymbol{c}_{\boldsymbol{k}}$- one k per pixel. Texton *channels*.
- reconstruct $\hat{} = \boldsymbol{G}^+ \hat{\boldsymbol{d}}_{\boldsymbol{k}} = \boldsymbol{G}^+ \boldsymbol{c}\boldsymbol{f}_{\boldsymbol{k}}$, $\boldsymbol{G}^+ = (\boldsymbol{G}^{\boldsymbol{T}}\boldsymbol{G})^{-1}\boldsymbol{G}$, or SVD



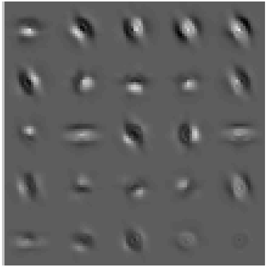
original, textons, reconstruction

Texton example

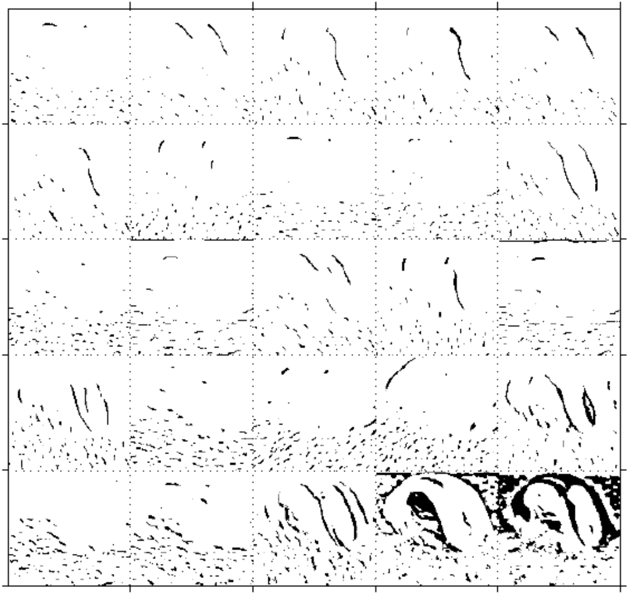
(a)



(b)



Texton decomposition



Texton segmentation

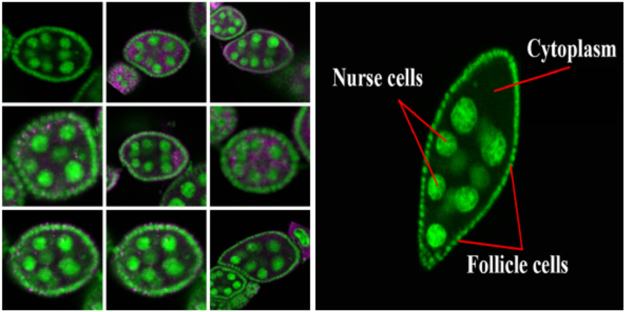
- Texton histogram in a local neighborhood →
    - measuring texture similarity

$$\chi^2(h_i, h_j) = \frac{1}{2}\sum_{k=1}^{K}\frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}$$

    - descriptor to classify textures
- other cues - contours, isotropy... (probably outdated)

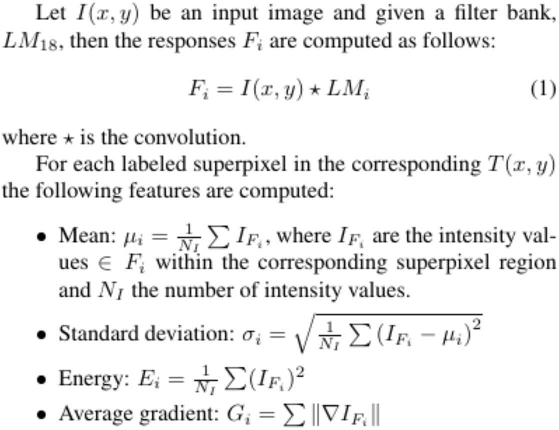
Example - Drosophila segmentation

Nava, Kybic, ICIP2015



Supertextons

- SLIC superpixels
- 18 filters max over 6 orientations (for invariance) of 3 scales of $1^{\text{st}}+2^{\text{nd}}$DoG, plus 8 LoG, 4 Gaussians



Let $I(x, y)$ be an input image and given a filter bank, LM_{18}, then the responses F_i are computed as follows:

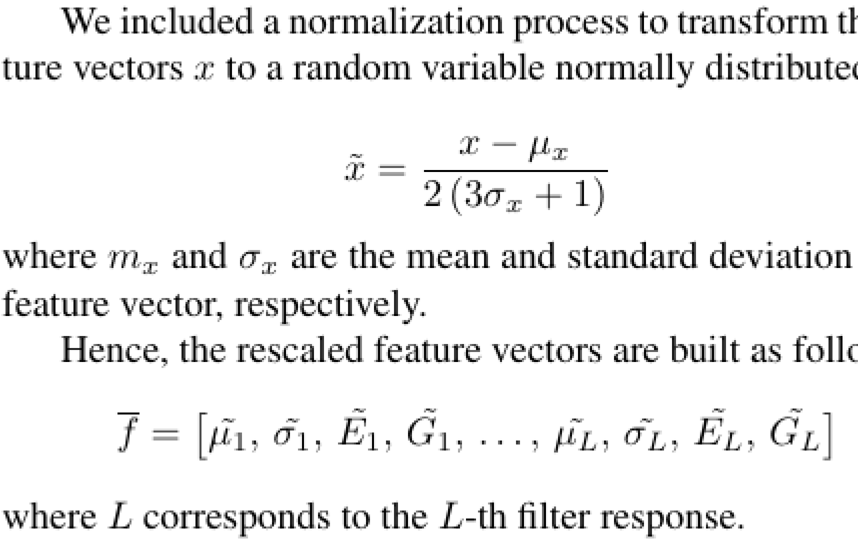
$$F_i = I(x, y) \star LM_i \qquad (1)$$

where $\star$ is the convolution.

For each labeled superpixel in the corresponding $T(x, y)$ the following features are computed:

- Mean: $\mu_i = \frac{1}{N_I} \sum I_{F_i}$, where I_{F_i} are the intensity values $\in$ F_i within the corresponding superpixel region and N_I the number of intensity values.
- Standard deviation: $\sigma_i = \sqrt{\frac{1}{N_I} \sum \left(I_{F_i} - \mu_i\right)^2}$
- Energy: $E_i = \frac{1}{N_I} \sum (I_{F_i})^2$
- Average gradient: $G_i = \sum \|\nabla I_{F_i}\|$

Supertextons (2)



We included a normalization process to transform th ture vectors x to a random variable normally distribute

$$\tilde{x} = \frac{x - \mu_x}{2\left(3\sigma_x + 1\right)}$$

where m_x and σ_x are the mean and standard deviation feature vector, respectively.

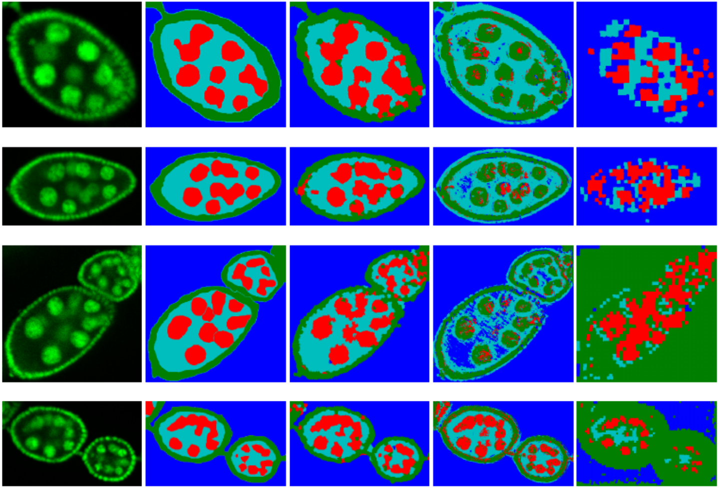
Hence, the rescaled feature vectors are built as follo

$$\overline{f} = \left[\tilde{\mu_1}, \tilde{\sigma_1}, \tilde{E}_1, \tilde{G}_1, \ldots, \tilde{\mu_L}, \tilde{\sigma_L}, \tilde{E}_L, \tilde{G}_L\right]$$

where L corresponds to the L-th filter response.

- k-means clustering for each class, k=0,1 N
- Classification by k-NN (k=6)

Segmentation results



original, ground truth, supertextons, pixel-based, textons