

w03b — Separable filtering: solution

The steps. A 2D filter K is separable iff it can be written as an outer product $\mathbf{v}\mathbf{h}^\top$, which happens iff $\text{rank}(K) = 1$. So:

1. Compute the SVD $K = U\Sigma V^\top$.
2. Inspect the singular values. Exactly one non-zero $\Rightarrow \text{rank } 1 \Rightarrow \text{separable}$.
3. If separable, $\mathbf{v} = \sigma_1 \mathbf{u}_1$ and $\mathbf{h}^\top = \mathbf{v}_1^\top$ (the scale σ_1 may go into either).

(Equivalently, and quicker by hand: check that every 2×2 minor vanishes, or that all rows are scalar multiples of one another.)

This filter.

$$K = \begin{pmatrix} 1 & 3 & 1 \\ 3 & 9 & 3 \\ 1 & 3 & 1 \end{pmatrix}$$

Rows are $1 \times (1, 3, 1)$, $3 \times (1, 3, 1)$, $1 \times (1, 3, 1)$ — all multiples of the same vector, so rank 1 and separable.

$$K = \begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 3 & 1 \end{pmatrix}$$

How to prove you got it right. Multiply the two 1D components back out and check you recover K exactly; or convolve a test image both ways and compare (they should agree to floating-point precision).

Via SVD numerically, $\sigma_1 = 11$ and $\sigma_2 = \sigma_3 = 0$; the recovered 1D kernels will be scaled versions of $(1, 3, 1)$, e.g. $\pm(1, 3, 1)/\sqrt{11}$ each. **Any consistent scaling is correct** — the split of the constant between the two factors is arbitrary, and sign flips cancel. A common surprise: numpy hands back $(-0.30, -0.90, -0.30)$, which looks wrong but is not.

Cost. Separable filtering is $O(k)$ per pixel instead of $O(k^2)$ — here 6 multiplications instead of 9, a modest gain that grows quickly with kernel size.