⊕ ⊖ Computational ⊗ ⊘ Photography

Denoising

Jongmin Baek

CS 478 Lecture
Feb 13, 2012

Announcements

- Term project proposal
  - Due Wednesday
- Proposal presentation
  - Next Wednesday
  - Send us your slides (Keynote, PowerPoint, etc)
  - 4 minutes per group
- Assignment #2 grading
  - Sign up at Rm. 360.

Overview

- Noise model
- Image priors
  - Self-similarity
  - Sparsity
- Algorithms
  - Non-local Means
  - BM3D

Noise

- Every observation incurs some uncertainty.

Ground truth

Observation

Noise

- Every observation incurs some uncertainty.

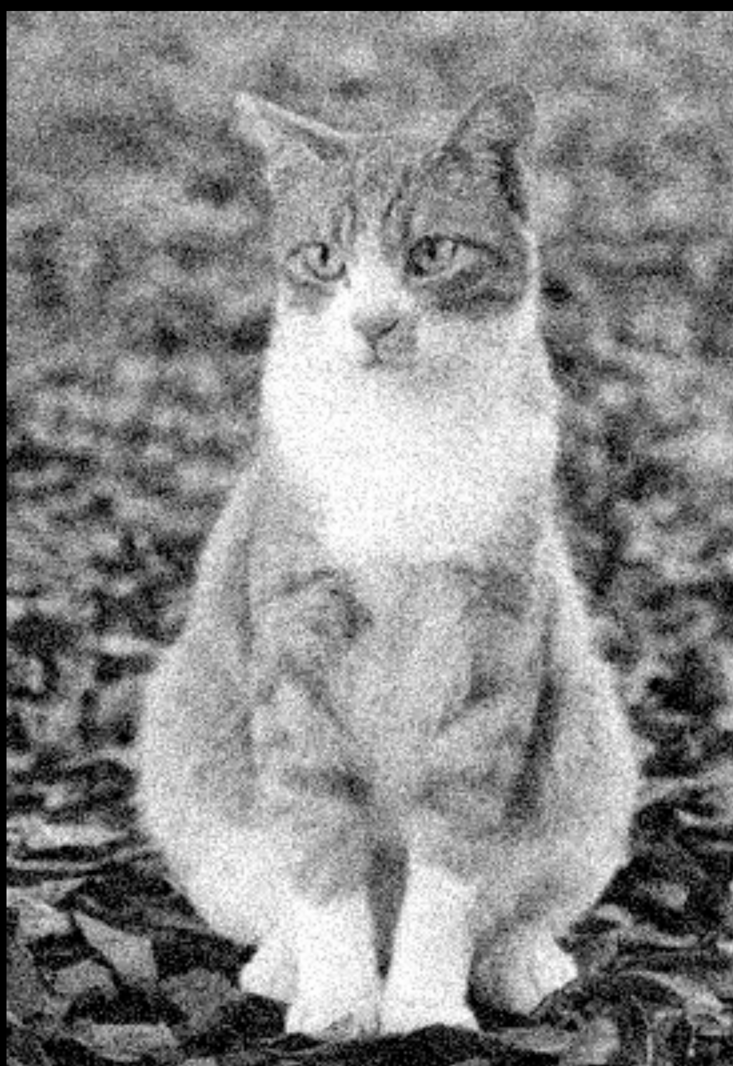
ISO 1600

Source of Noise

- Photon shot noise
- Read noise
- Thermal noise
- Pixel non-uniformity
- Processing artifacts (demosaicking, JPEG)
- ...

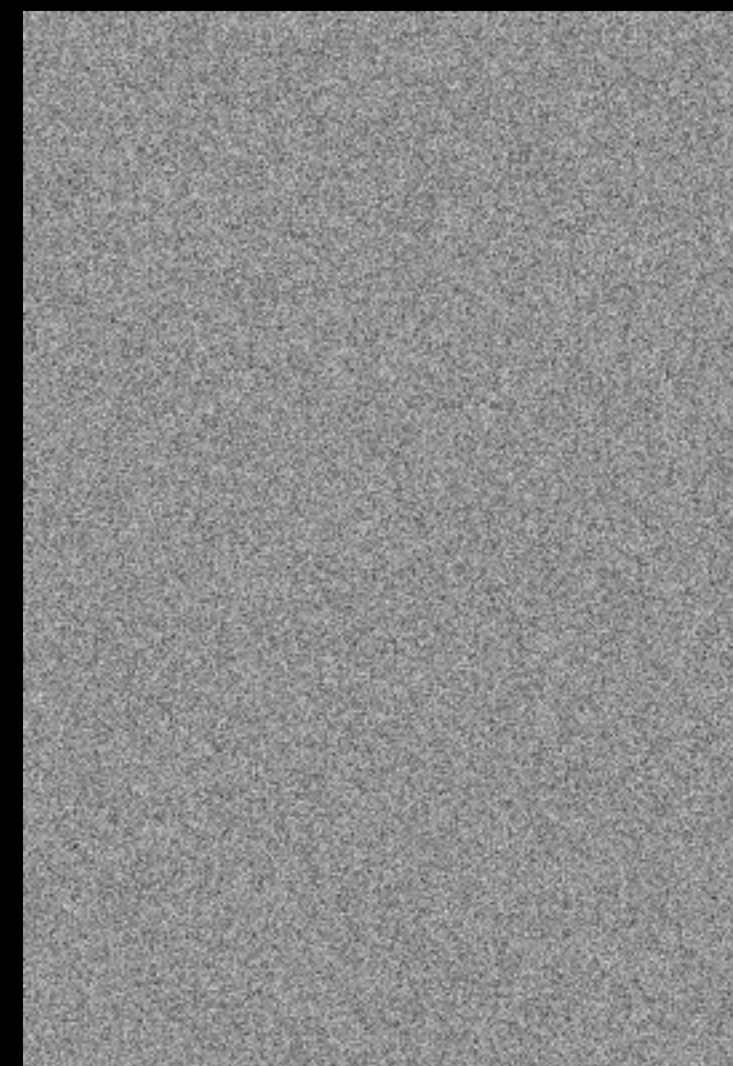
Image Formation Model

Observation | Scene | Additive Noise



=

+



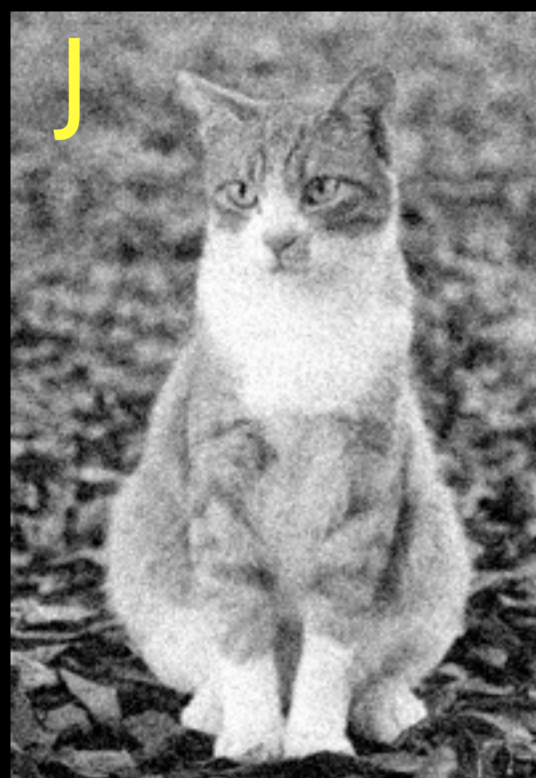
J | I | N_σ

Aside: Why Gaussian?

- Too many noise sources to model individually.
- Sum of random variables tend to a Gaussian distribution.
- Denoising algorithms exist for other models.

Problem

Observation

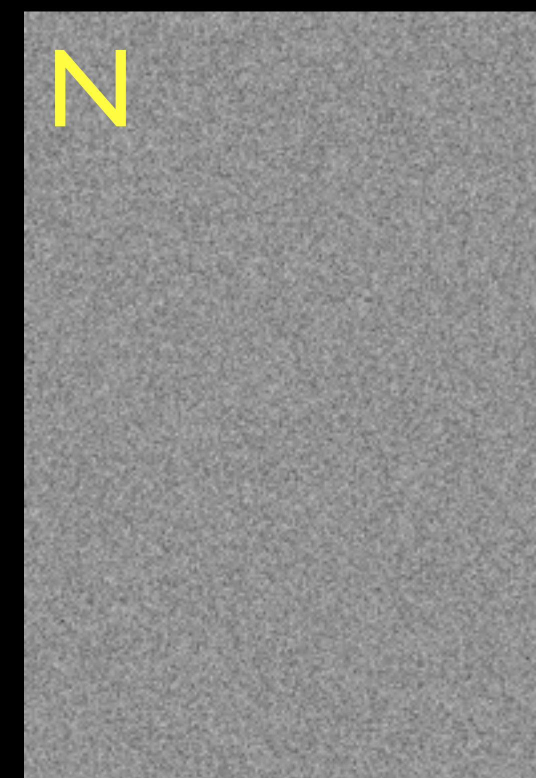


=

Scene

+

Noise



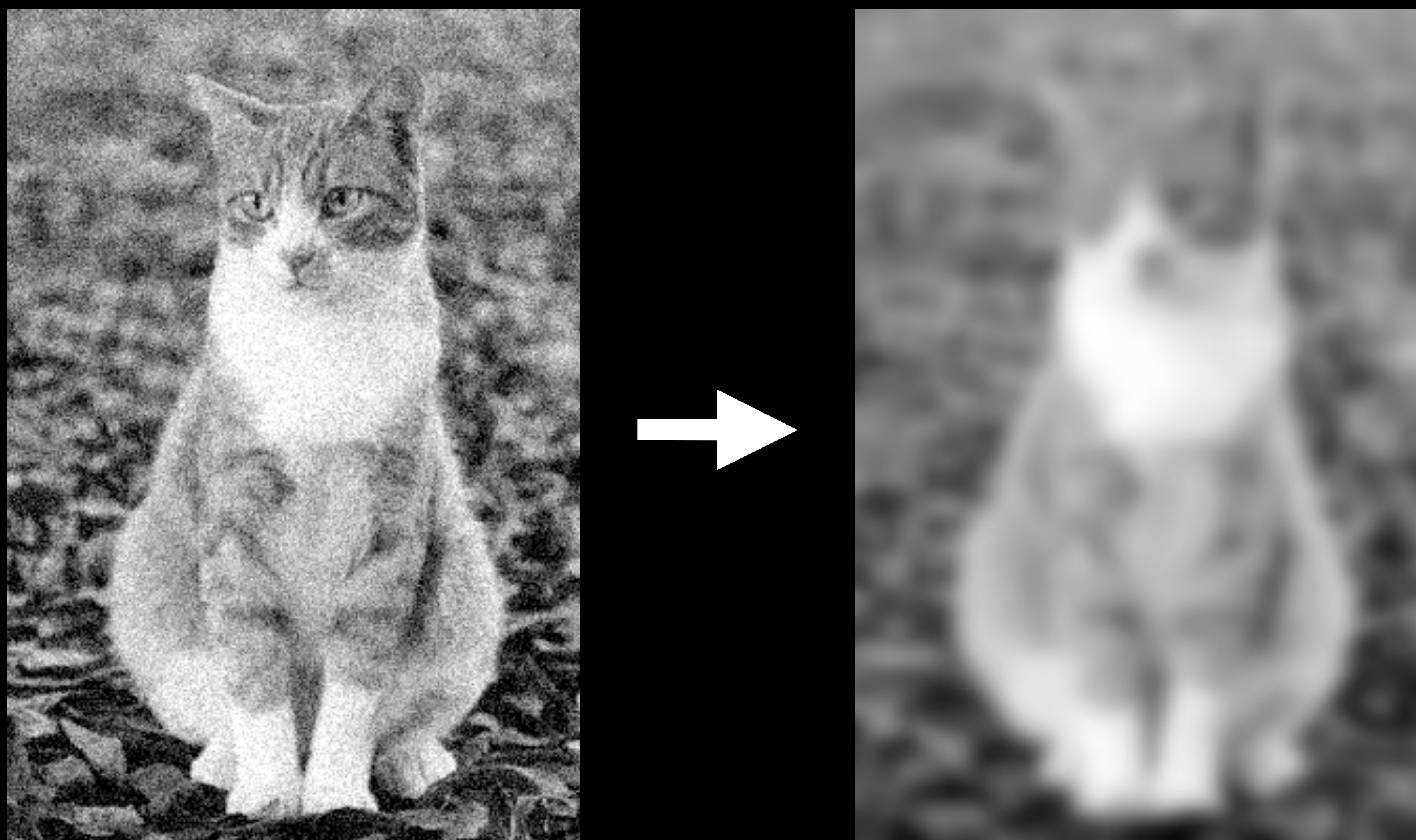
- Given J, compute I (and N.)
- More unknowns than constraints!

Image Priors

- Images are not just random collection of intensity values.
  - High correlations among nearby pixels
  - Denoise by averaging with nearby pixels?

Gaussian Filter

- Gaussian of stdev 5



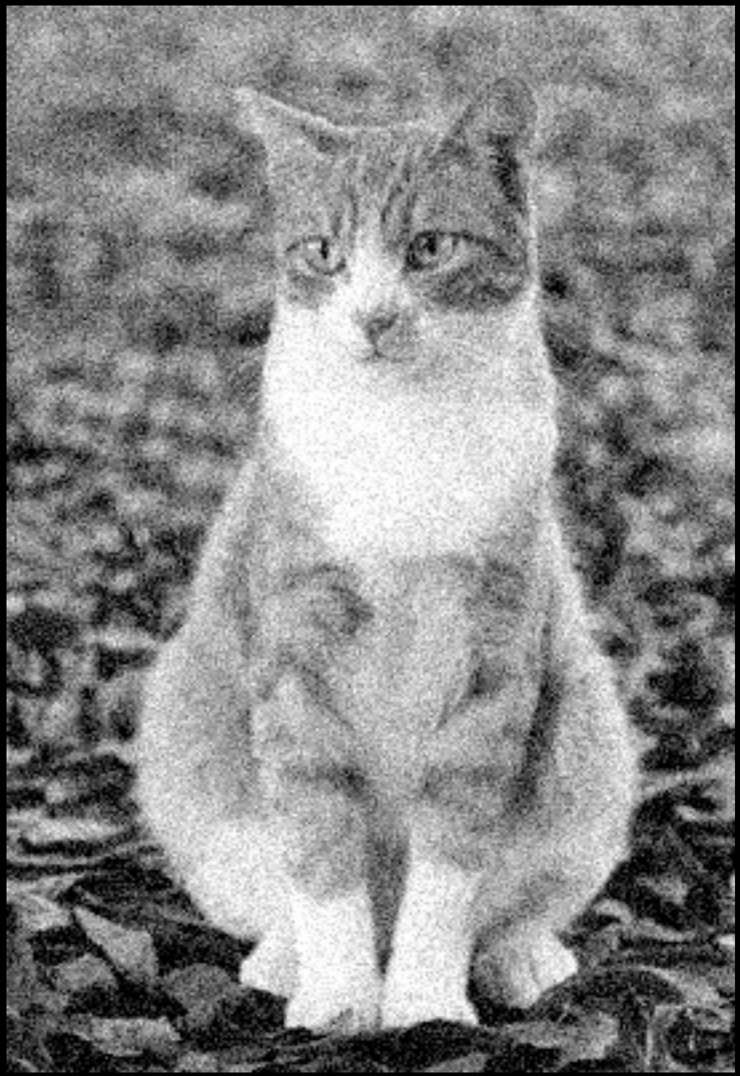
- Noise is gone, but so is detail.

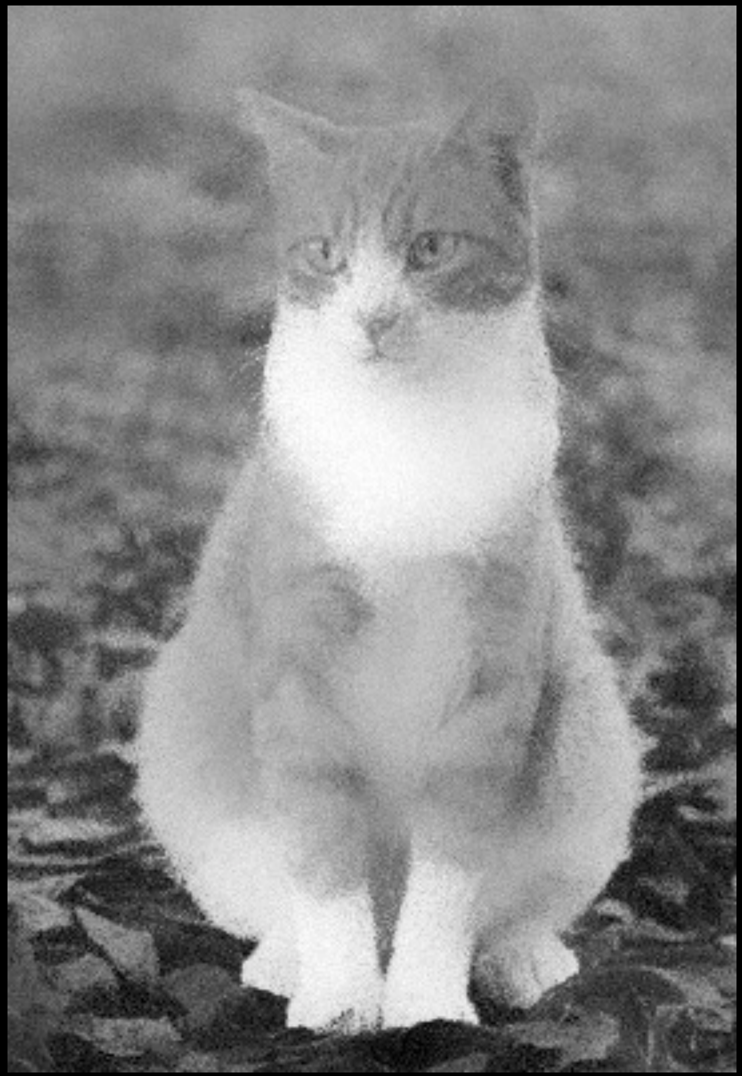
Image Priors

- Images are not just random collection of intensity values.
  - High correlations among nearby pixels
  - Denoise by averaging with nearby pixels?
  - Denoise by averaging with nearby pixels of similar color?

Bilateral Filter

- Bilateral filter with $\sigma_x = \sigma_y = 10$, $\sigma_r = 0.2$





- Better, but still not great when noise is high.

Image Priors

- Images are not just random collection of intensity values.
  - Denoise by averaging with nearby pixels?
  - Denoise by averaging with nearby pixels of similar color?
  - Denoise by averaging with nearby pixels of similar texture?

Non-Local Means

Buades et al., 2005 (CVPR)

- Natural images have repetitive textures.
  - Pixels with similar textures will probably have similar values.
  - More discriminative than bilateral filtering.

How-to: NL-Means

- It turns out this can be ensconced in the Gaussian filtering framework.

$$v'(x) = \sum'_y v(y)\, f(\;\underline{p(x)}\; - \;\underline{p(y)}\;)$$

- Here p(x) is the image patch centered at x, in a vectorized form.

How-to: NL-Means

$$v'(x) = \sum'_y v(y)\, f(\ \ p(x)\ -\ p(y)\ \ )$$

- For every *x*,
  - Compute vector p(*x*).
  - For every neighbor *y*, Only look at 21x21 window around x
    - Compute vector p(*y*). 7x7 patch around pixel, so 49-dim vector
    - Calculate the weight $\exp(\ -|p(x)-p(y)|^2\ /\ 2\sigma^2\ )$
    - Do the weighted sum.

How-to: NL-Means

$v'(x) = \sum'_y v(y)\, f(\ \ p(x)\ -\ p(y)\ \ )$

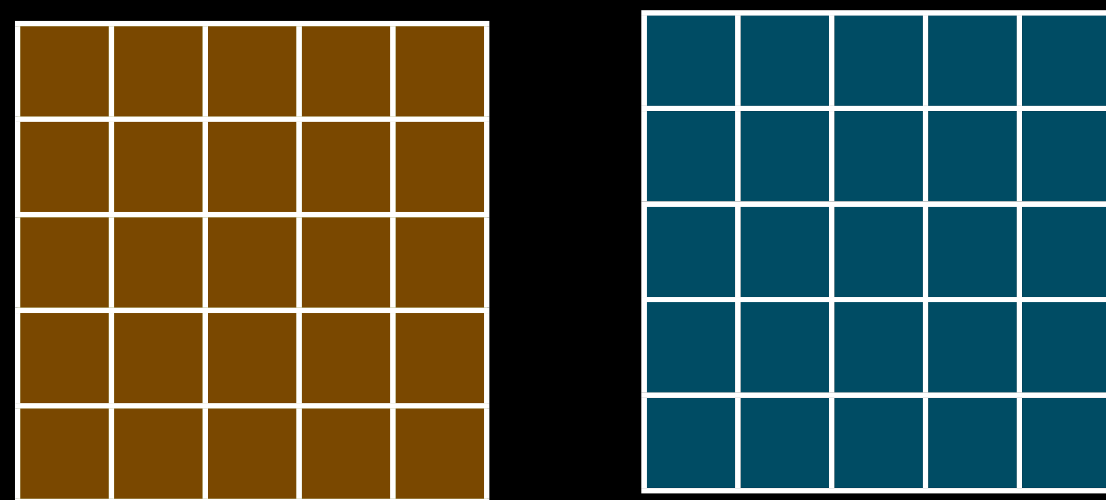
- Slow part
  - Calculate, for every pair (x,y),

    $|p(x)-p(y)|^2$

  - Sum of squared difference between two patches.

Calculating SSD

A_x $\quad$ A_y



$$\sum_i \sum_j | A_x(i,j) - A_y(i,j) |^2$$

$$= \underline{\sum_i \sum_j A_x^2(i,j)}$$

$$+ \underline{\sum_i \sum_j A_y^2(i,j)} \quad \text{Easy with integral image}$$

$$- 2 \underline{\sum_i \sum_j A_x(i,j) A_y(i,j)} \quad \text{Same as } A_x \otimes A_y^T$$

NL-Means with FFT

$$v'(x) = \sum'_y v(y)\, f(\; p(x) \; - \; p(y) \;)$$

- Compute the integral image of v^2.
- For every x,
  - Compute A_x.
  - Compute $f^{-1}\{\, f\{A_x\}\, f\{v\} \,\}$ Simulates $A_x \otimes A_y^{\top}$ for all y
  - For every neighbor y,
    - Calculate the weight $\exp(\, -|p(x)-p(y)|^2 / 2\sigma^2 \,)$
    - Do the weighted sum.

NL-Means with FFT

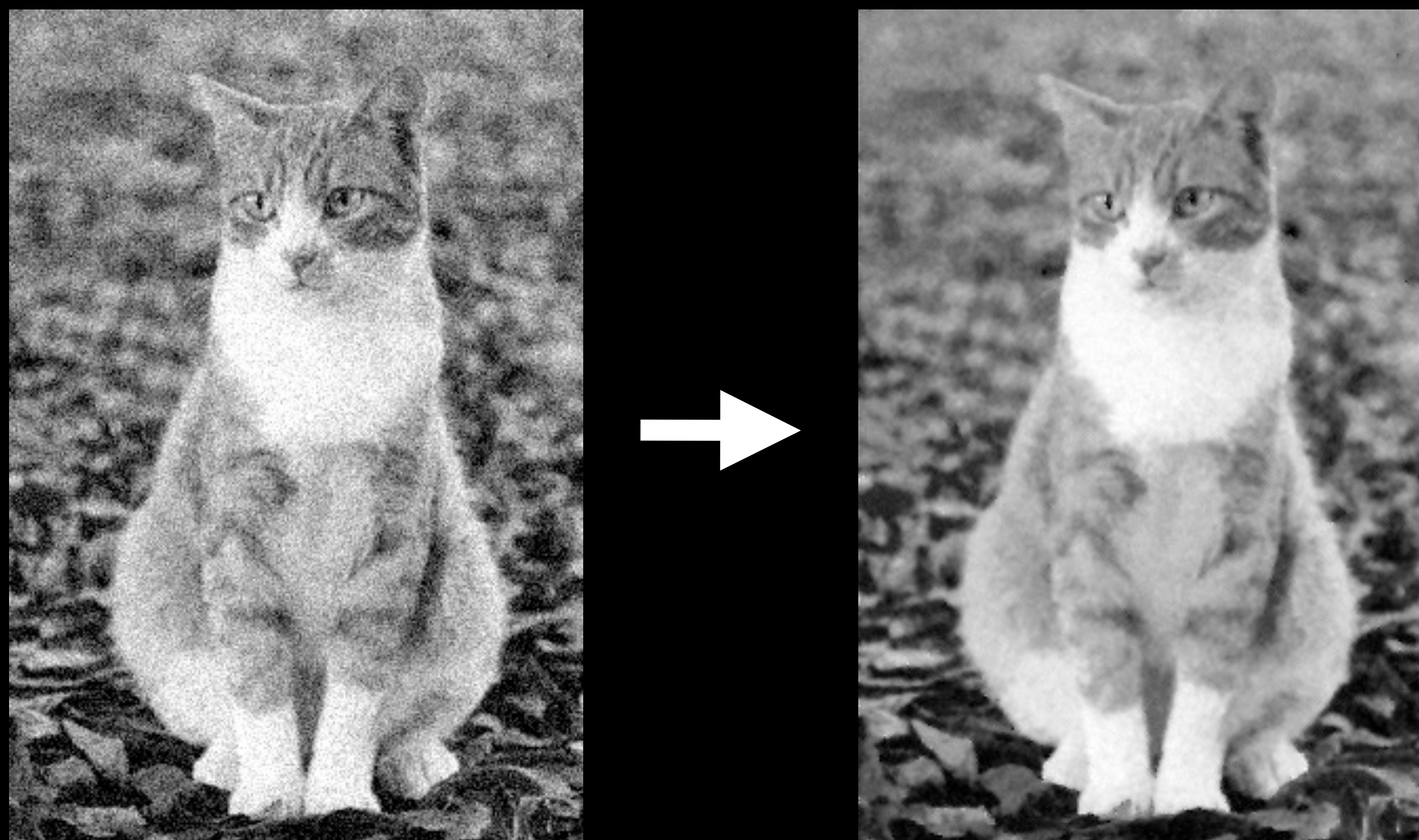
	Runtime N = # of pixels, M = dim. of p-space
Naive	$O(N^2M)$
With FFT	$O(N^2 \log N)$

NL-Means with FFT

when neighbor search is restricted to N'

	Runtime N = # of pixels, M = dim. of p-space
Naive	O(N N' M)
With FFT	O(N (N'+M) log (N'+M))

NL-Means Filter



- Why does it work better than bilateral?

Even Faster NL-Means

- Last time, we discussed how to make Gaussian filters very, very fast.
    - Applicable here as well?

Challenges

$$v'(x) = \sum\nolimits'_y v(y)\, f(\ \underline{p(x)}\ -\ \underline{p(y)}\ )$$

- p is very high-dimensional.
  - Time complexity of our filtering algorithms scale with dimensionality of p.
  - Can we lower the dimensionality of p?

Patch Space

$$v'(x) = \sum'_y v(y)\, f(\ \underline{p(x)}\ -\ \underline{p(y)}\ )$$

- Not all values in the p-space are equally plausible.
  - There are subspaces that are much more likely.
  - PCA to reduce dimensionality?

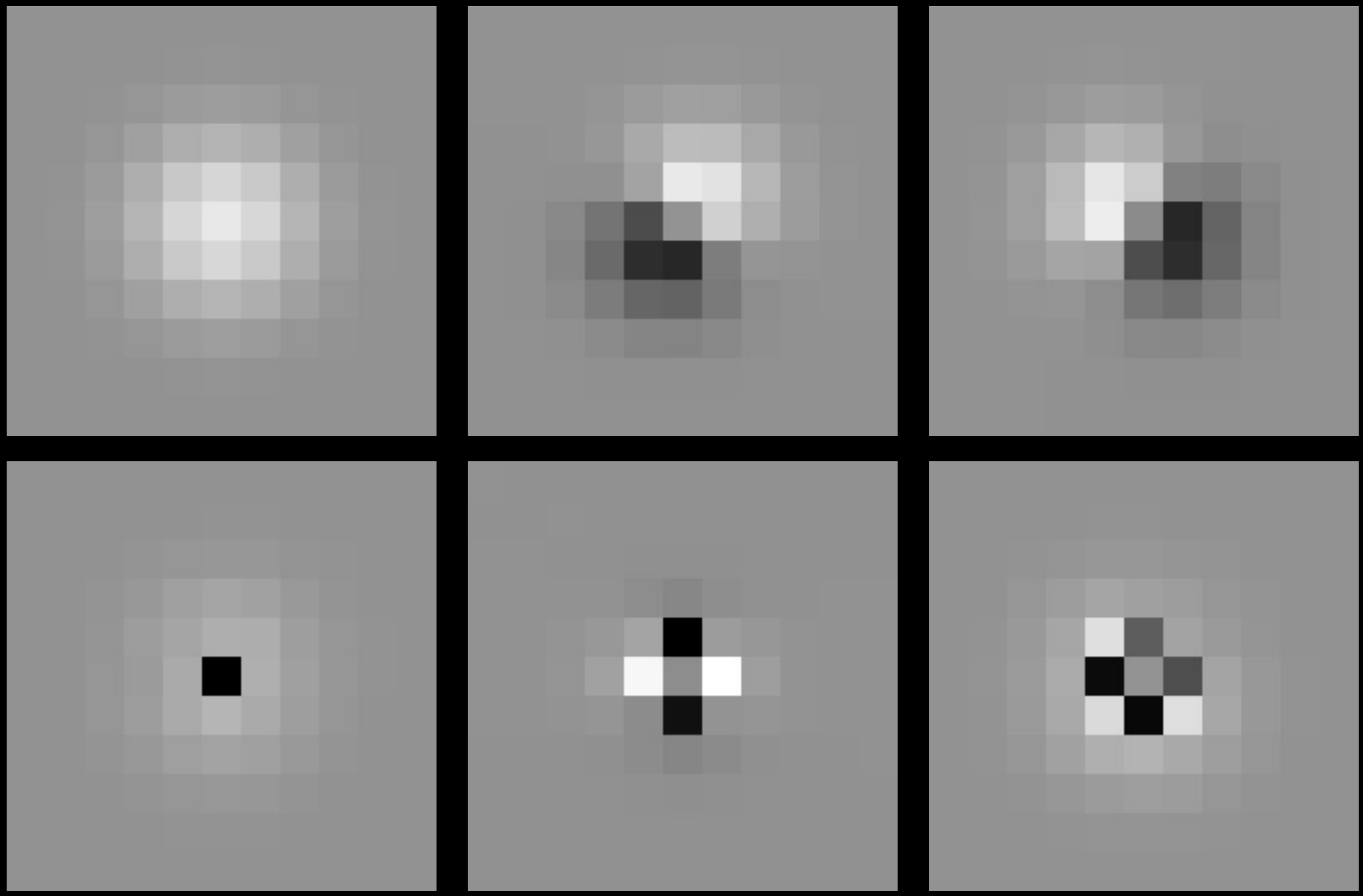
PCA on Patches

$$v'(x) = \sum'_y v(y)\, f(\underline{p(x)} - \underline{p(y)})$$

- Generate the p-vectors for all pixels.
  - High dimensional! (e.g. 147 if 7x7 patch on 3-channel img)
- Perform PCA to identify commonly occurring subspaces.
  - Perhaps find ~6 principle components.
- Project the p-vectors onto this subspace.
- Voilà.

PCA on Patches

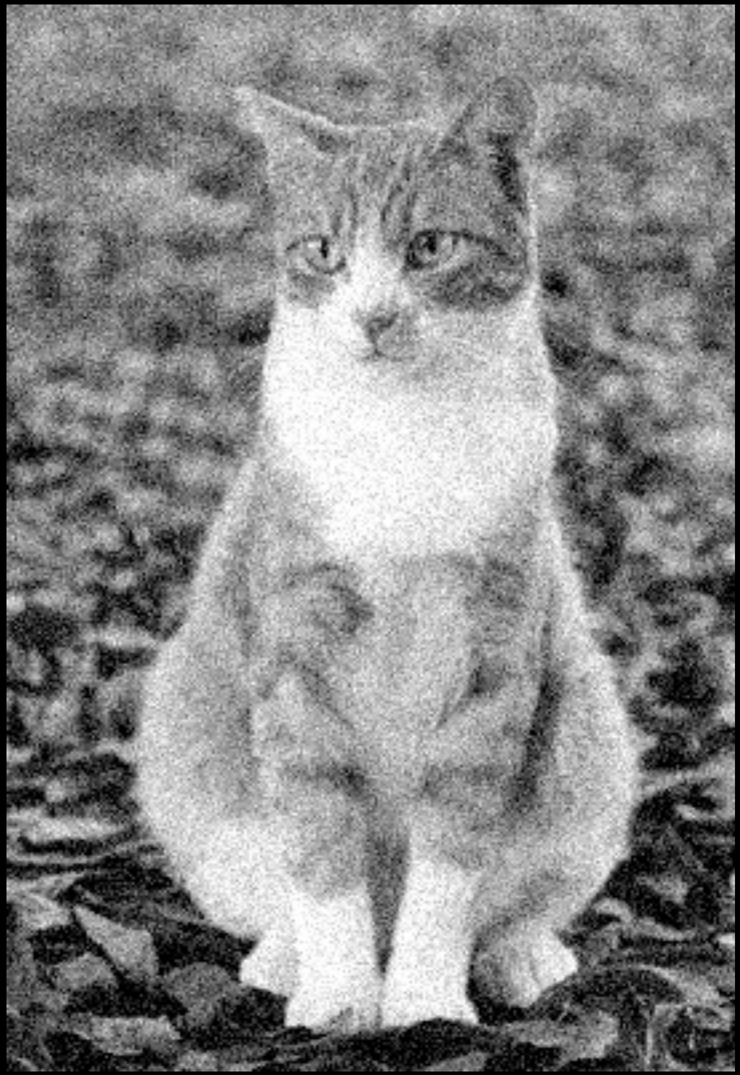
Six principal components from the cat image



PCA on Patches

- PCA performs denoising!
  - PCA throws away non-principal components
  - Makes patches "closer" together.

NL-Means Filter



NL-Means: Analysis

- Find similar patches and average them.
  - Should we do something besides averaging?

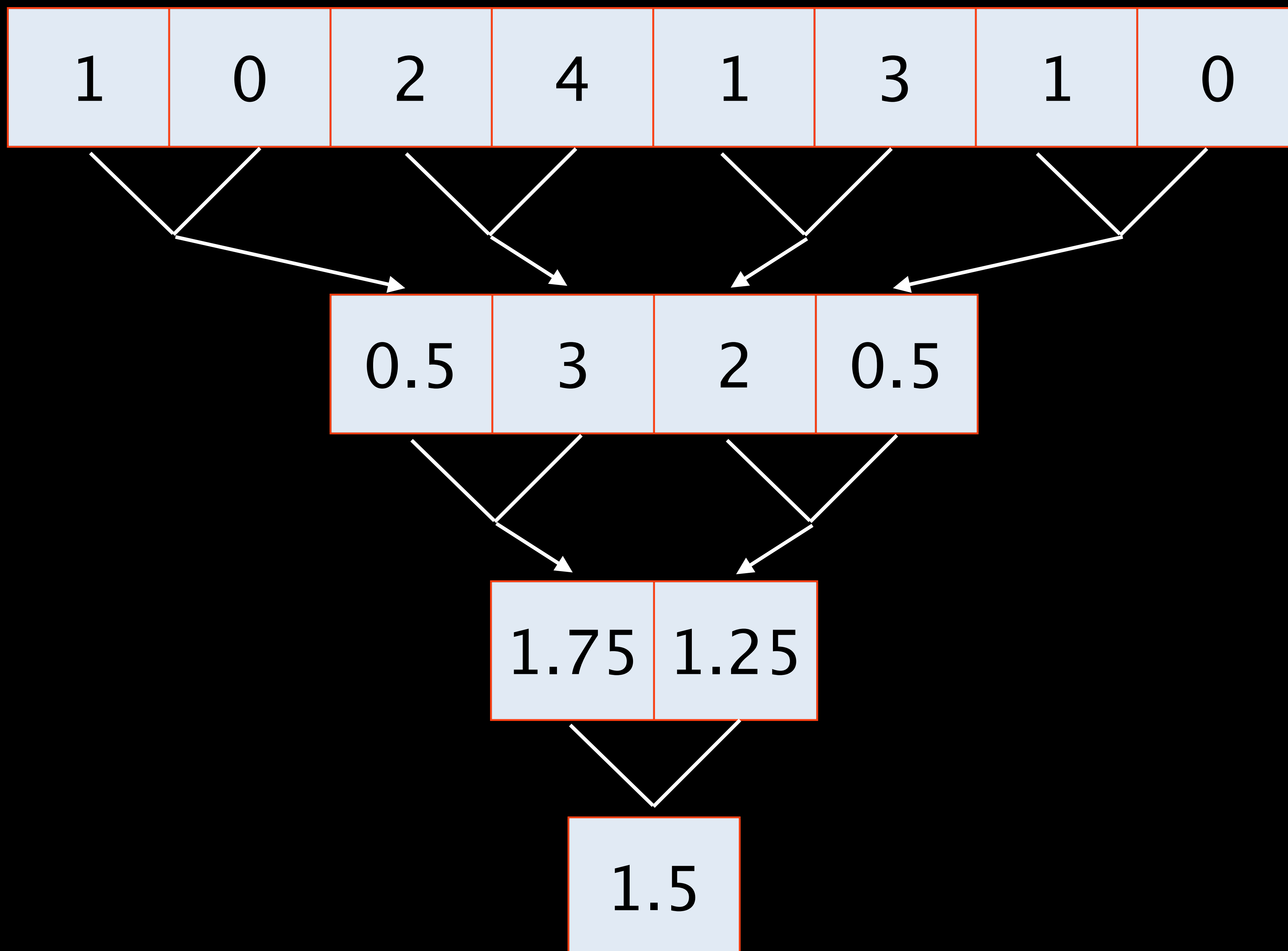
Wavelet Shrinkage

- Hypothesis
  - There exists a transform *T* such that applying *T* to patches will admit a sparse representation.
- This is useful in compression.
  - DCT in JPEG encoding.
  - PCA in NL-Means dimensionality reduction.

Wavelet Shrinkage

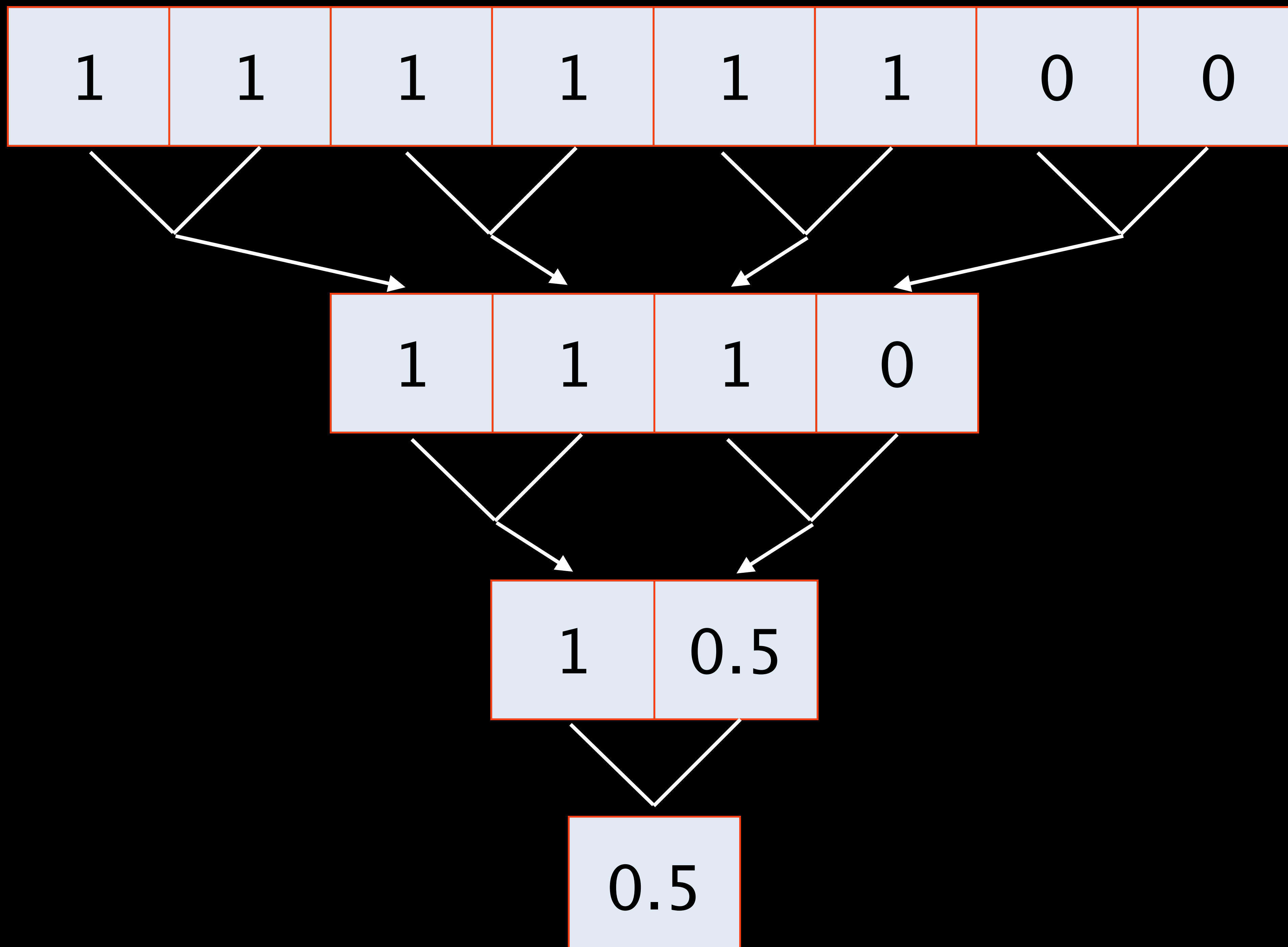
- Take a patch in the image.
  - Apply Haar wavelet transform

Haar Wavelet Transform



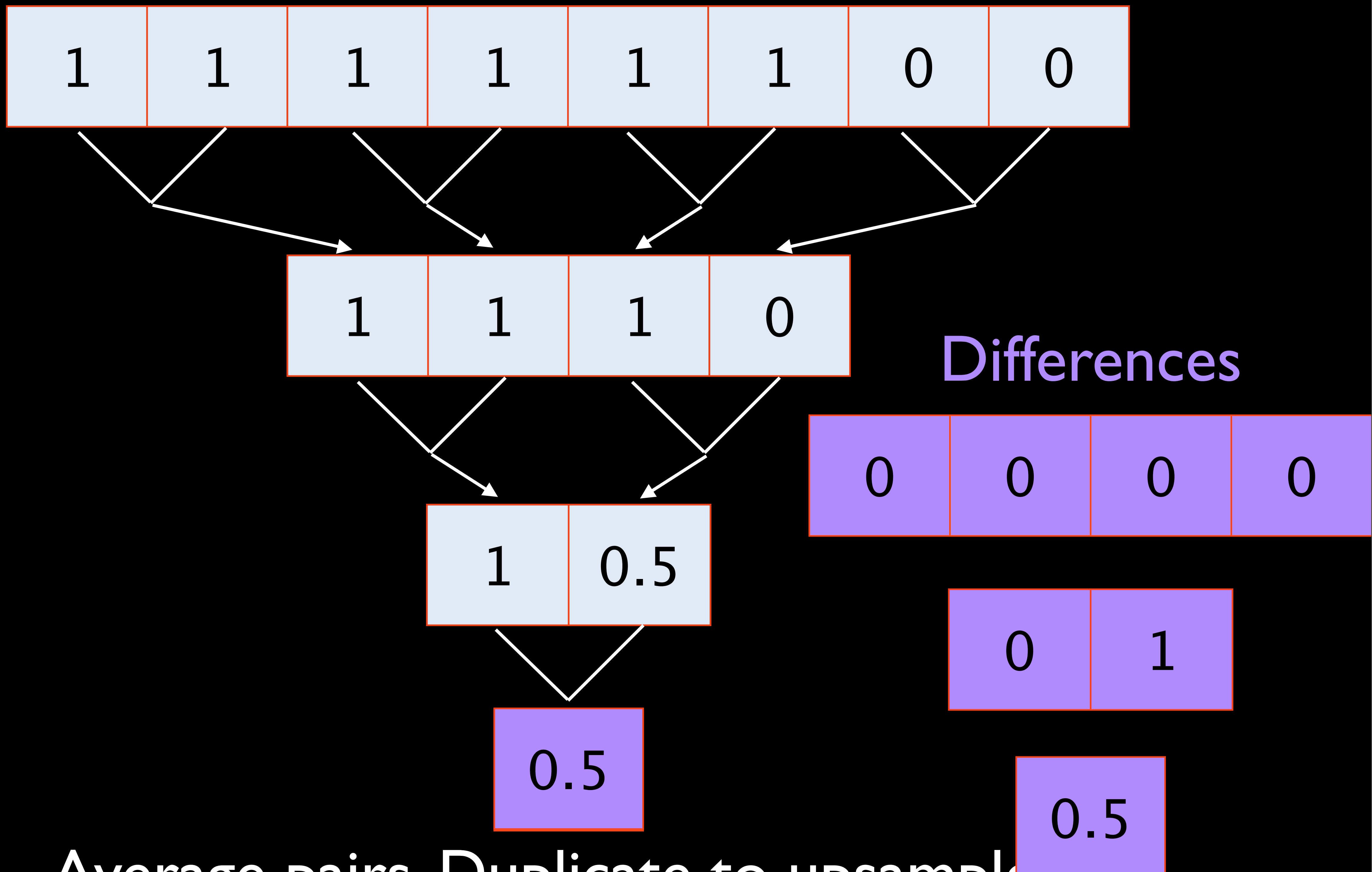
Average pairs. Duplicate to upsample

Haar Wavelet Transform



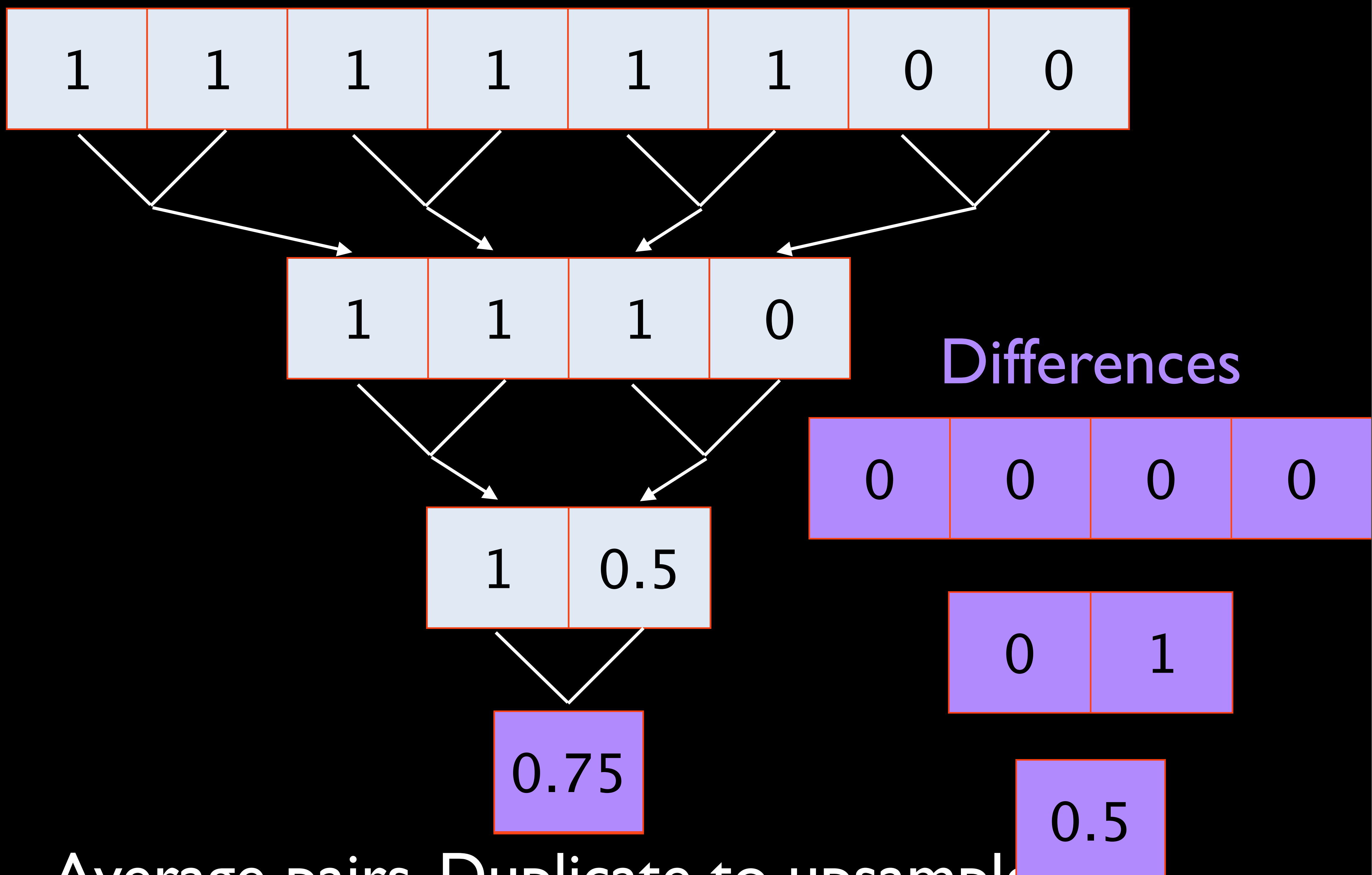
Average pairs. Duplicate to upsample

Haar Wavelet Transform



Average pairs. Duplicate to upsample

Haar Wavelet Transform



Average pairs. Duplicate to upsample

Haar Wavelet Transform

Input	1	1	1	1	1	1	0	0
Output	0	0	0	0	0	1	0.5	0.75

Note that the coefficients are sparse!

Haar Wavelet Transform

Input	1	1	1	1	1	1	0	0
Output	0	0	0	0	0	1	0.5	0.75
Noisy Input	1.03	0.94	1.00	0.95	1.04	0.98	0.00	0.02
Noisy Output	0.09	0.05	0.06	-0.02	0.01	1.00	0.47	0.745

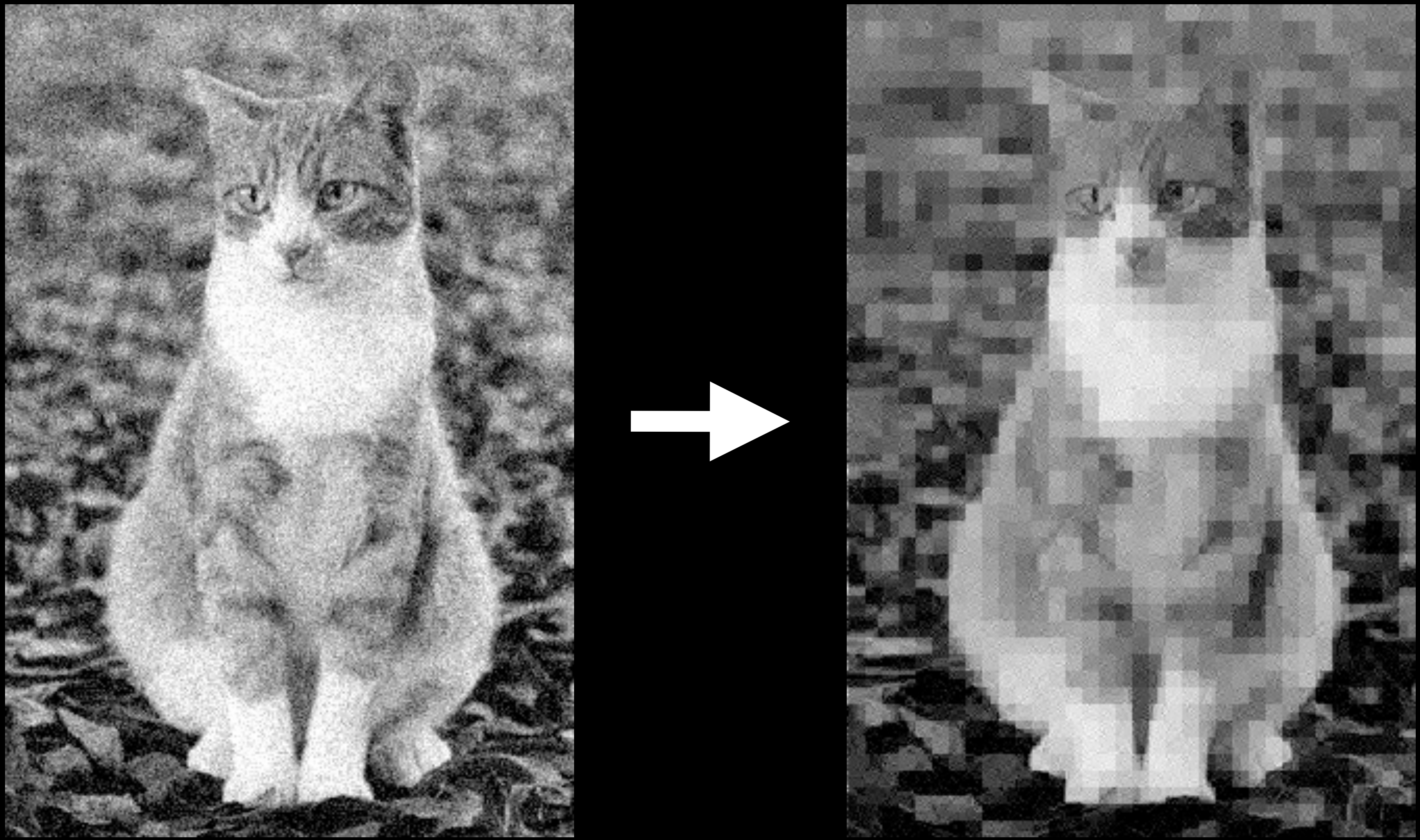
The coefficients are no longer sparse!

Wavelet Shrinkage

- For every image patch,
  - Perform 2D Haar wavelet transform.
  - Perform a soft thresholding:
    - Pull each coefficient towards zero (by some amount Δ.)
    - The patch is now more "natural"
  - Invert transform to recover patch.

Wavelet Shrinkage

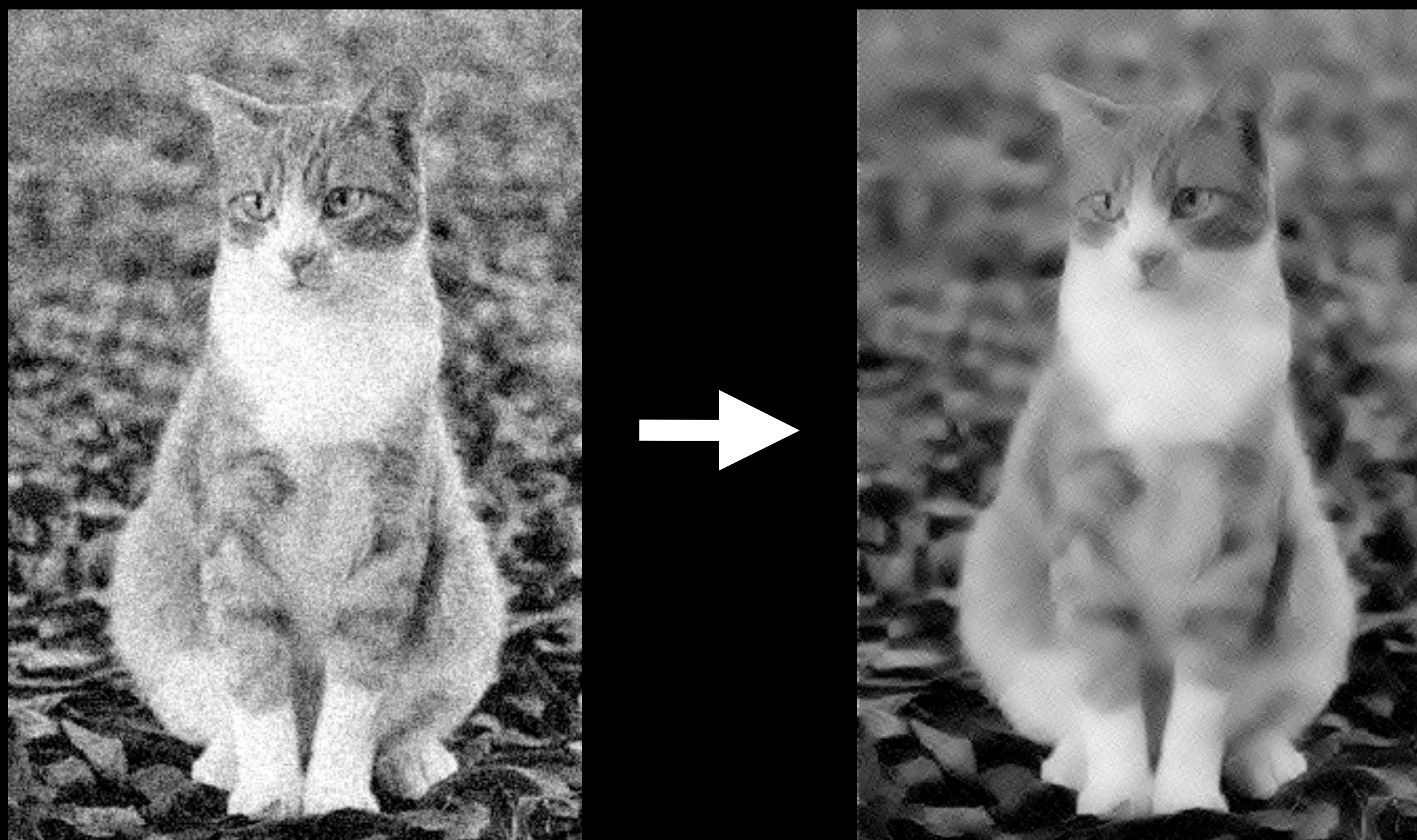
Process 8x8 patches



Huh?

Partitioning the image creates artifacts.

Wavelet Shrinkage



Process all (overlapping) patches and blend them.

Summary

- Non-Local Means
  - Exploit the inter-patch correlations.
- Wavelet Shrinkage
  - Exploit the intra-patch correlations.
- Can we perhaps do both?

BM3D

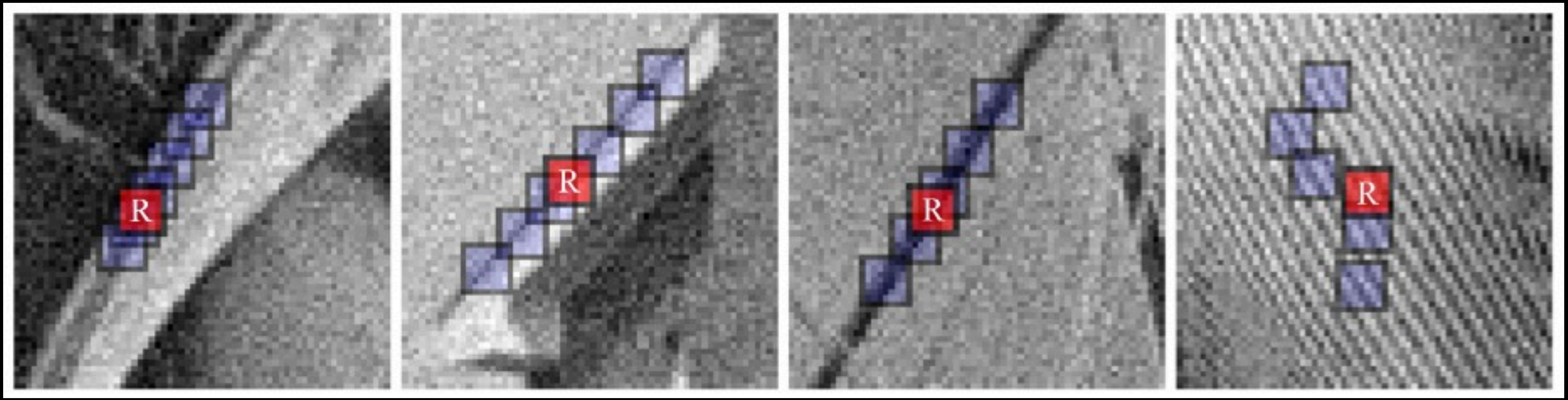
Dabov et al., 2006 (IEEE TIP)

- "Block-Matching 3D"
  - Perform wavelet thresholding.
  - Also combine multiple patches.
- Widely recognized as the state-of-the-art denoising technique.

BM3D

Dabov et al., 2006 (IEEE TIP)

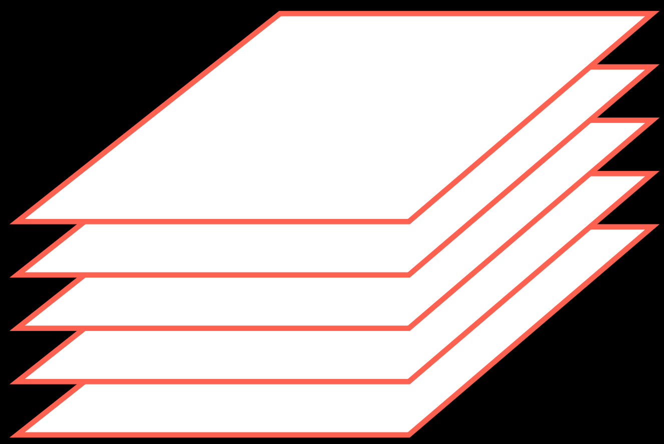
- Step 1. For each patch, find similar patches.



BM3D

Dabov et al., 2006 (IEEE TIP)

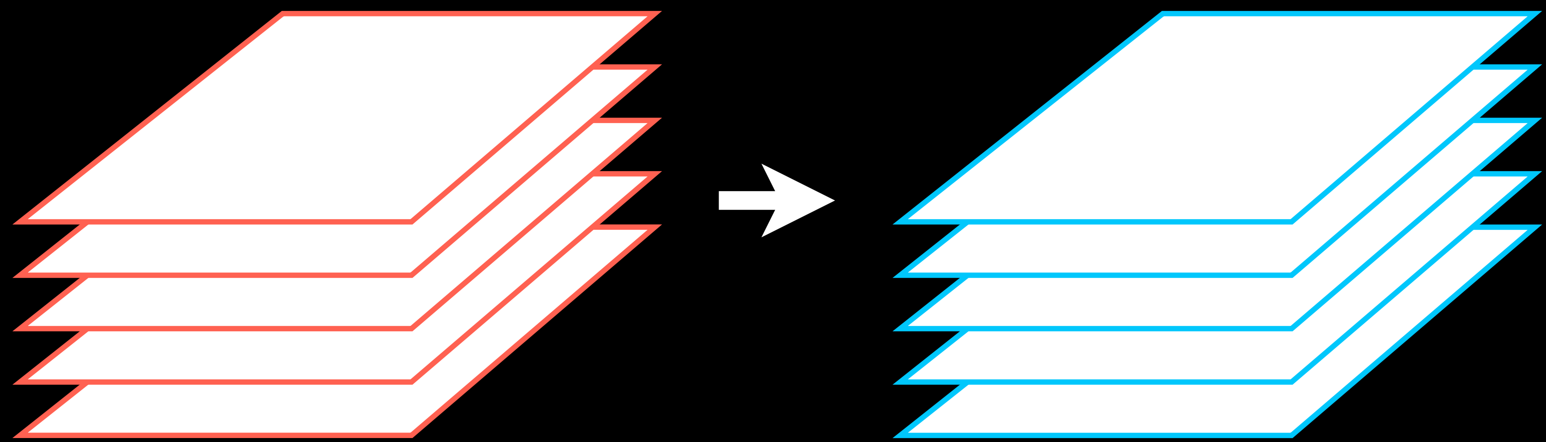
- Step 2. Group the similar patches into a stack.



BM3D

Dabov et al., 2006 (IEEE TIP)

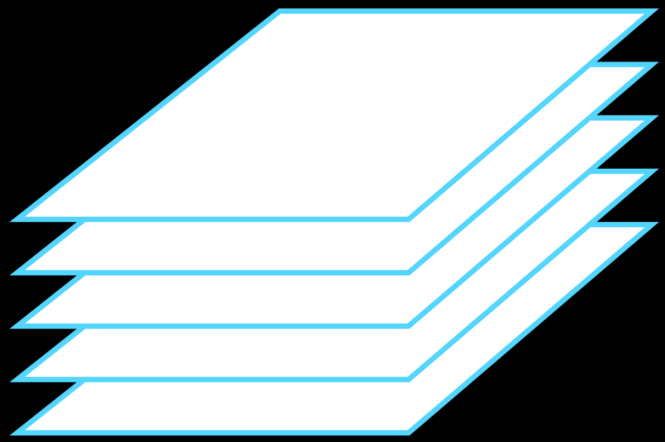
- Step 3. Perform a 3D Haar wavelet transform.



BM3D

Dabov et al., 2006 (IEEE TIP)

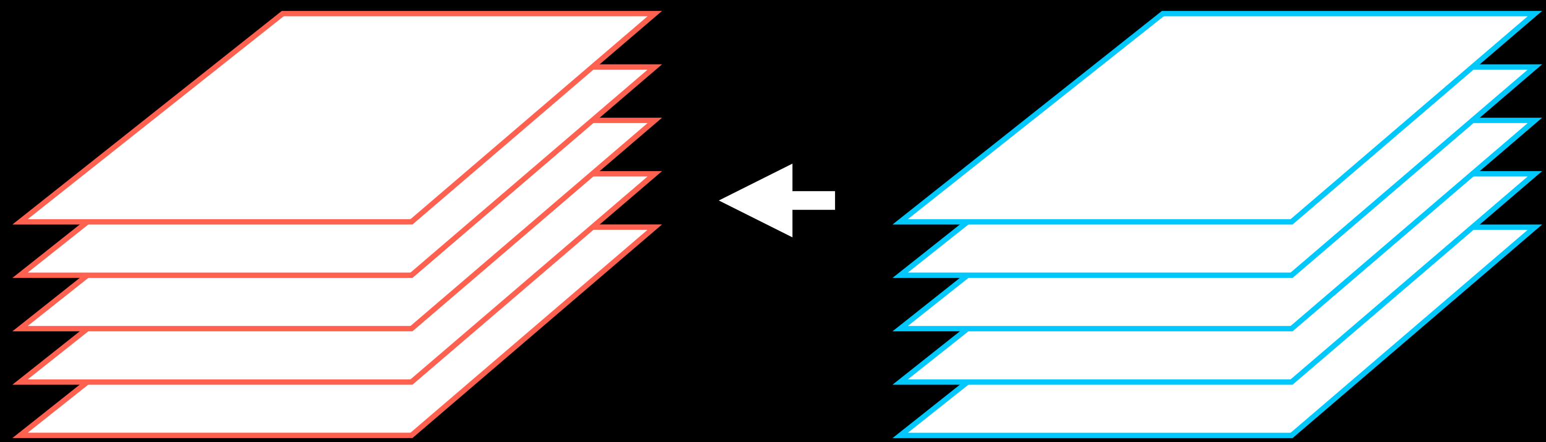
- Step 4. Apply shrinkage (or hard thresholding.)



BM3D

Dabov et al., 2006 (IEEE TIP)

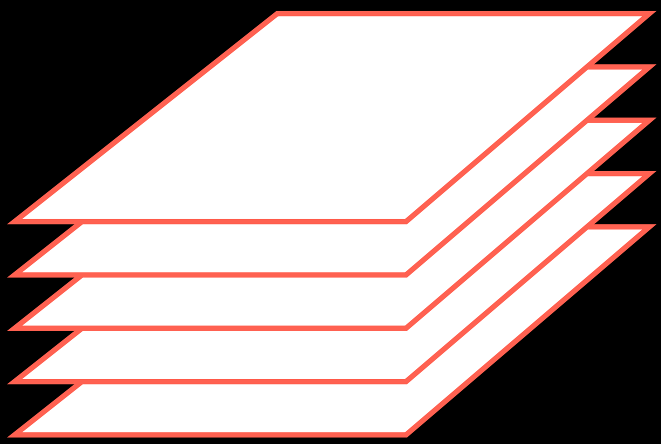
- Step 5. Apply inverse Haar wavelet transform.



BM3D

Dabov et al., 2006 (IEEE TIP)

- Step 6. Combine the patches to form image*.

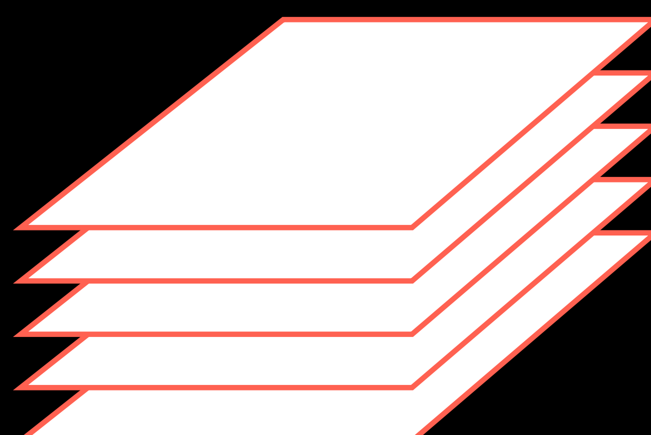


Each patch is a given weight inversely proportional to the # of nonzero entries in wavelet domain.

BM3D

Dabov et al., 2006 (IEEE TIP)

- Step 6. (Optional) Do it again.



Instead of thresholding, apply Wiener filter.
(Attenuate each coefficient by some scale factor.)

BM3D

Dabov et al., 2006 (IEEE TIP)

BM3D

Dabov et al., 2006 (IEEE TIP)

Works even with really bad noise

Summary

- Non-Local Means
  - Exploit the inter-patch correlations.
- Wavelet Shrinkage
  - Exploit the intra-patch correlations.
- BM3D
  - Exploit both.

Parting Thoughts

- In computational photography, we are not limited to taking a single photograph and denoising it!
  - Flash-no-flash pair denoising
  - Blurry-noisy pair denoising
  - Stack denoising
  - ...

Parting Thoughts

- The ideas here can be applied elsewhere.
  - Deblurring
  - Sharpening
  - Super-resolution
  - ...

Filler Slide

- How would you denoise video using one of these algorithms?
- How would you denoise a 3D mesh using one of these algorithms?

Questions?

Reminder

- Term project proposal
  - Due Wednesday
- Proposal presentation
  - Next Wednesday
  - Send us your slides (Keynote, PowerPoint, etc)
  - 4 minutes per group
- Assignment #2 grading
  - Sign up at Rm. 360.