EECE 4353 Image Processing

Lecture Notes on Mathematical Morphology: The Median Filter

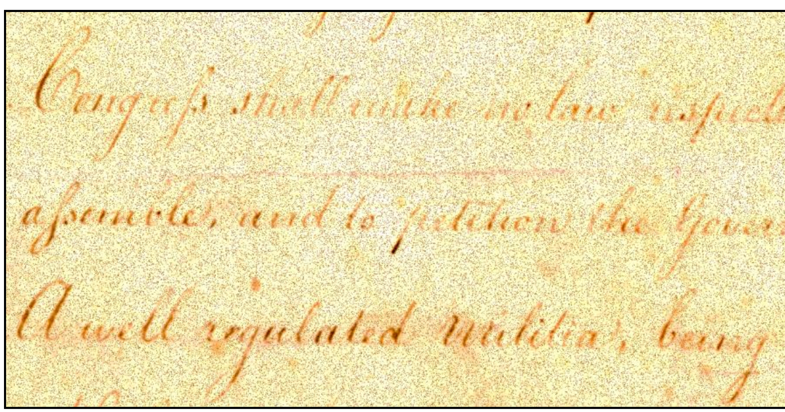
Richard Alan Peters II

Department of Electrical Engineering and Computer Science

Fall Semester 2016

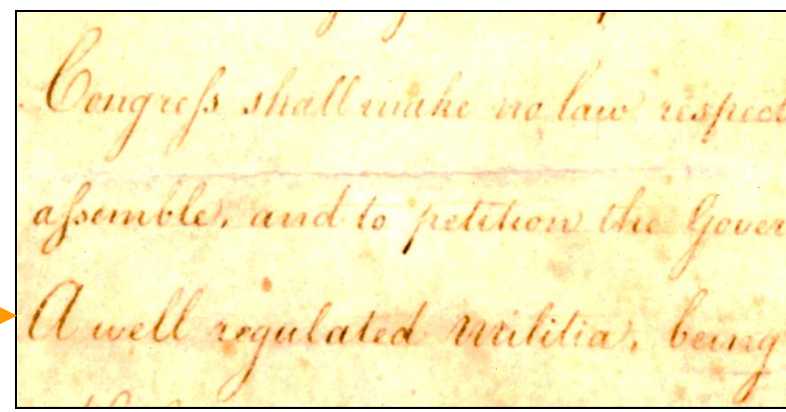
The Median Filter

- Returns the median value of the pixels in a neighborhood
- Is non-linear
- Is a morphological filter
- Is similar to a uniform blurring filter which returns the mean value of the pixels in a neighborhood of a pixel
- Unlike a mean value filter the median tends to preserve step edges



original

median filtered



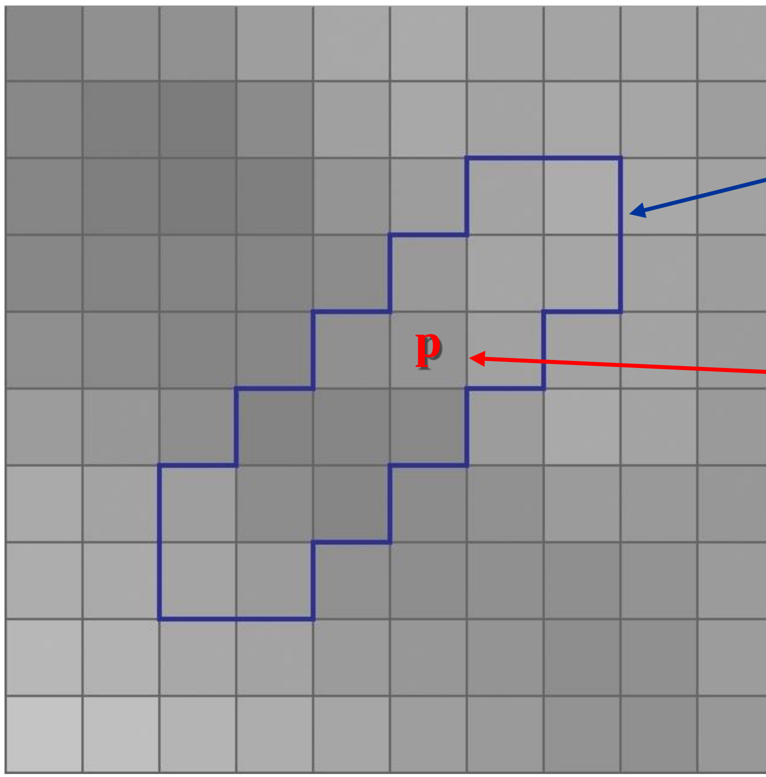
Median Filter: General Definition

$$\operatorname{med}\{\mathbf{I},\mathbf{Z}\}(\mathbf{p}) = \underset{\mathbf{q}\in\operatorname{supp}(\mathbf{Z}+\mathbf{p})}{\operatorname{median}}\{\mathbf{I}(\mathbf{q})\}$$

This can be computed as follows:

1. Let **I** be a monochrome (1-band) image.
2. Let **Z** define a neighborhood of arbitrary shape.
3. At each pixel location, **p** = (r,c), in **I** …
4. … select the n pixels in the **Z**-neighborhood of **p**,
5. … sort the n pixels in the neighborhood of **p,** by value, into a list $L(j)$ for $j = 1,\ldots,n$.
6. The output value at **p** is $L(m)$, where $m = \lfloor n/2 \rfloor + 1$.

Median Filter: General Definition



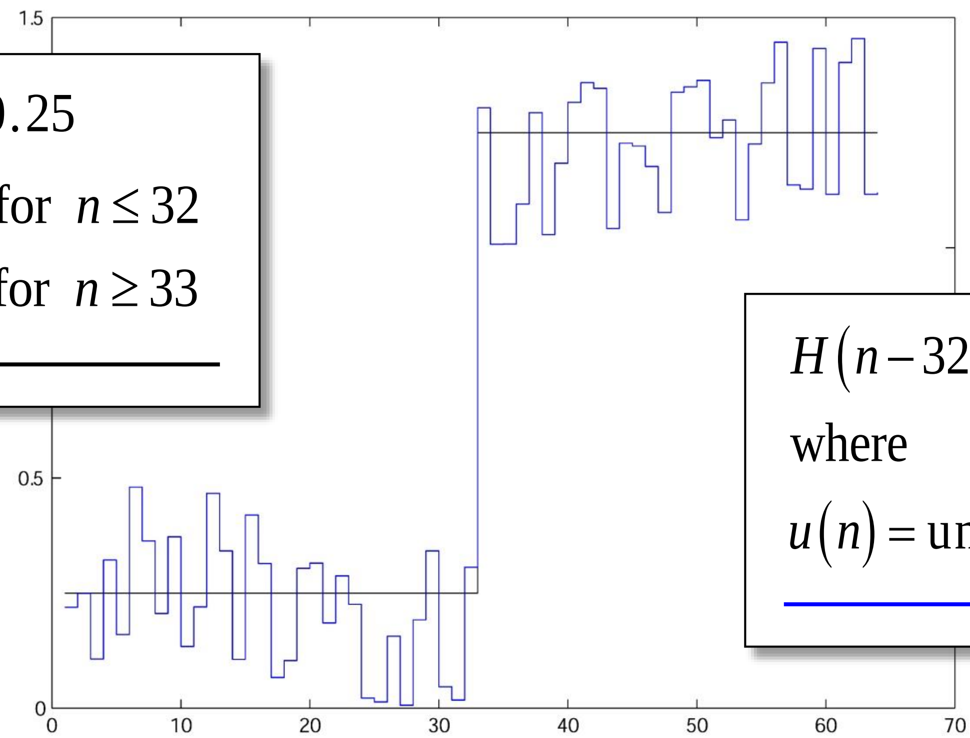
sorted intensity values from neighborhood of **p**.

median assigned to pixel loc **p** in output image.

131
133
133
136
140
143
147
152
154
157
160
162
163
164
165
171

A Noisy Step Edge

$$H(n-32.5)+0.25 = \begin{cases} 0.25 & \text{for } n \le 32 \\ 1.25 & \text{for } n \ge 33 \end{cases}$$



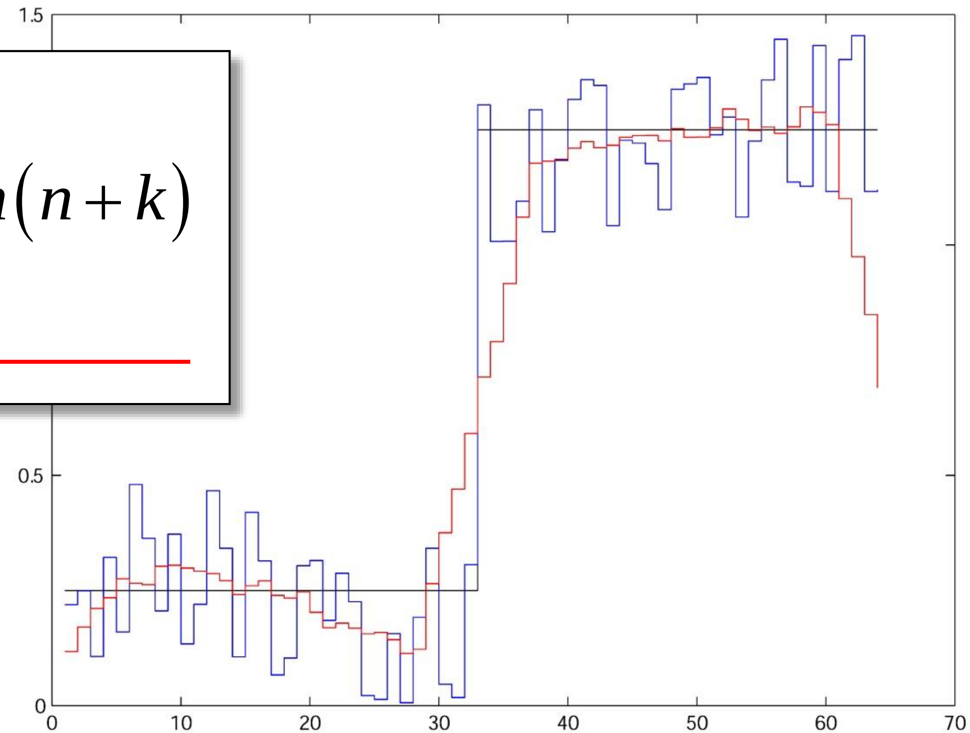
$$H(n-32.5)+u(n)$$

where

$$u(n) = \text{unif}\,(-0.25\,,0.25\,)$$

Blurred Noisy 1D Step Edge

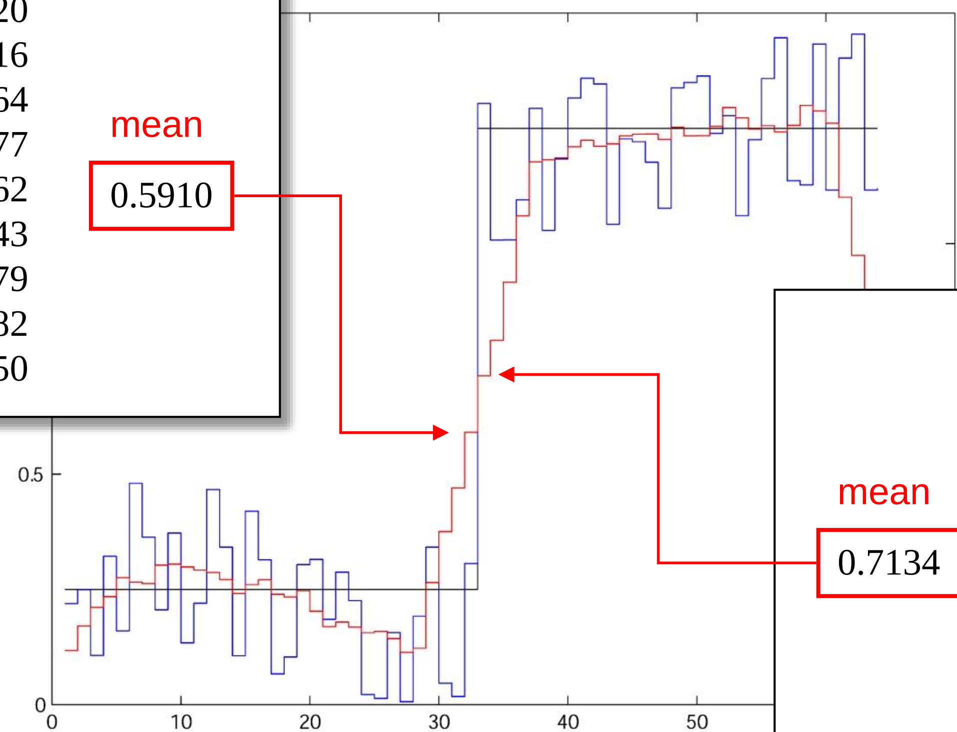
$$h(n) = \frac{1}{9}\sum_{k=1}^{4} h(n+k)$$



Blurred Noisy 1D Step Edge

J(32-4:32+4)=
0.1920
0.3416
0.0464
0.0177
0.3062
1.3043
1.0079
1.0082
1.0950

mean
0.5910

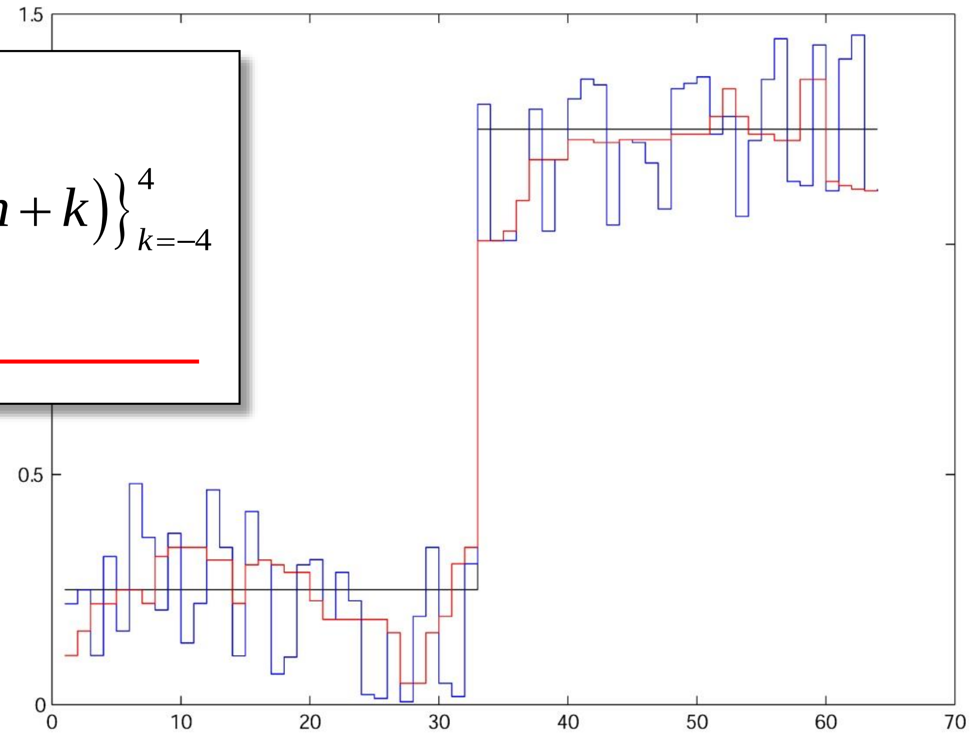


J(33-4:33+4)=
0.3416
0.0464
0.0177
0.3062
1.3043
1.0079
1.0082
1.0950
1.2935

mean
0.7134

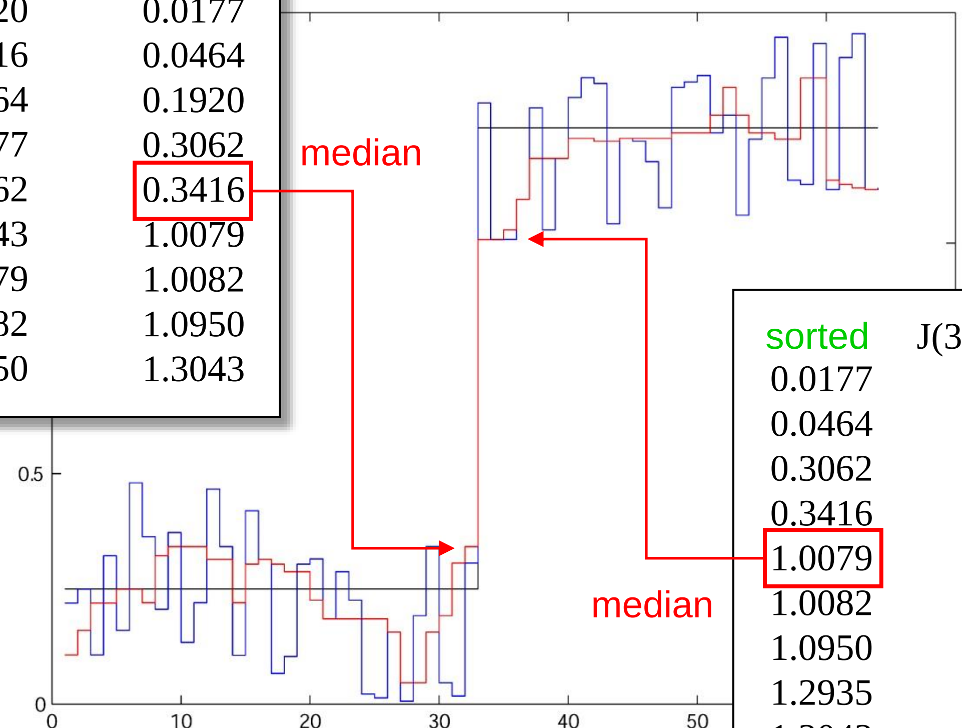
Median Filtered Noisy 1D Step Edge

$$h(n) = \mathrm{med}\left\{ h(n+k) \right\}_{k=-4}^{4}$$



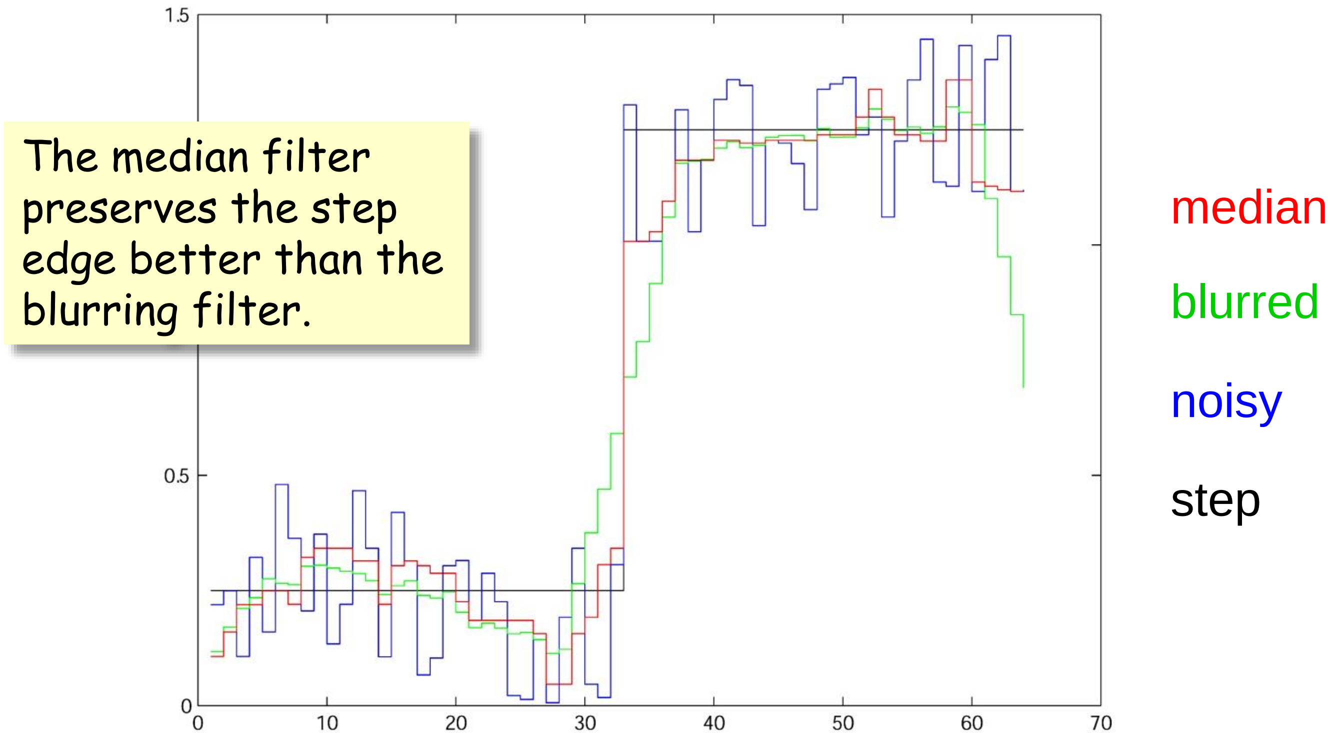
Median Filtered Noisy 1D Step Edge

J(32-4:32+4)=	sorted
0.1920	0.0177
0.3416	0.0464
0.0464	0.1920
0.0177	0.3062
0.3062	$\boxed{0.3416}$
1.3043	1.0079
1.0079	1.0082
1.0082	1.0950
1.0950	1.3043



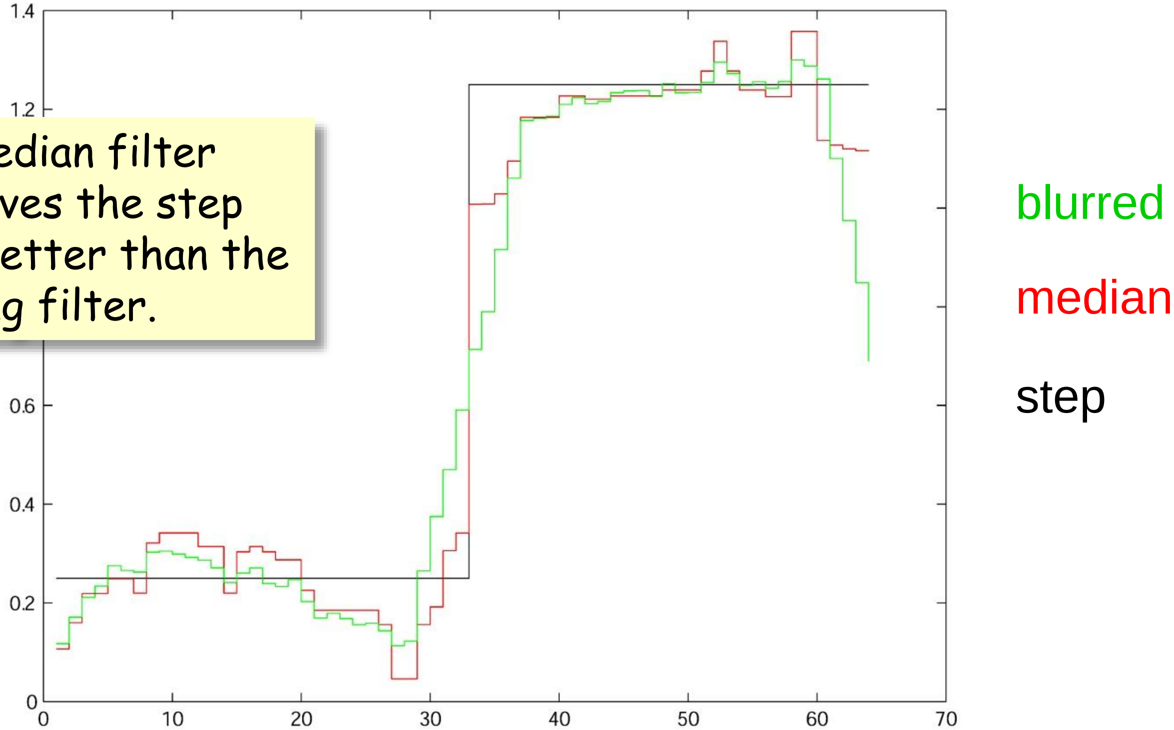
sorted	J(33-4:33+4)=
0.0177	0.3416
0.0464	0.0464
0.3062	0.0177
0.3416	0.3062
$\boxed{1.0079}$	1.3043
1.0082	1.0079
1.0950	1.0082
1.2935	1.0950
1.3043	1.2935

Median vs. Blurred



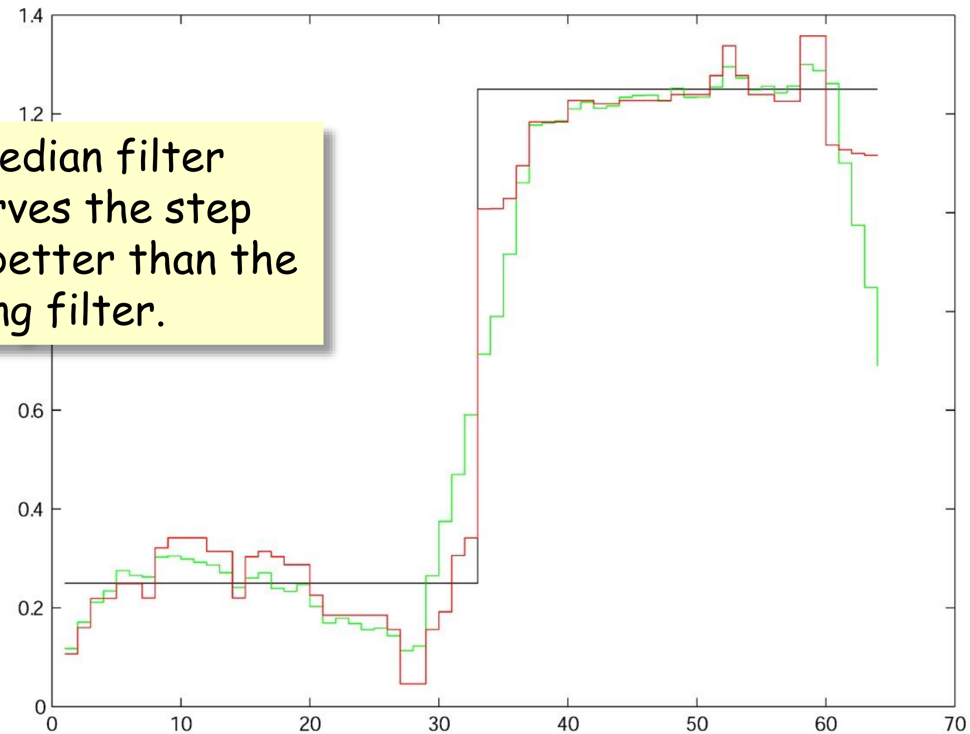
Median vs. Blurred

The median filter preserves the step edge better than the blurring filter.



Median vs. Blurred

The median filter preserves the step edge better than the blurring filter.

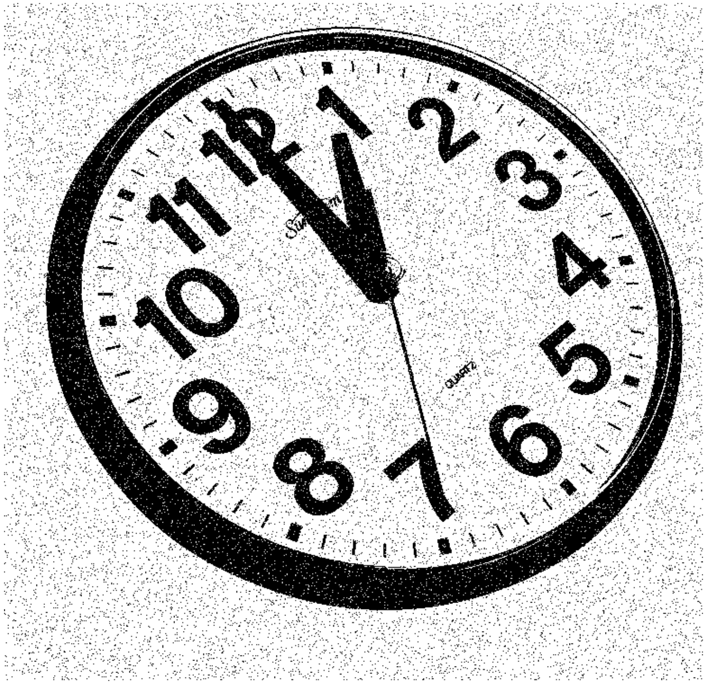


median

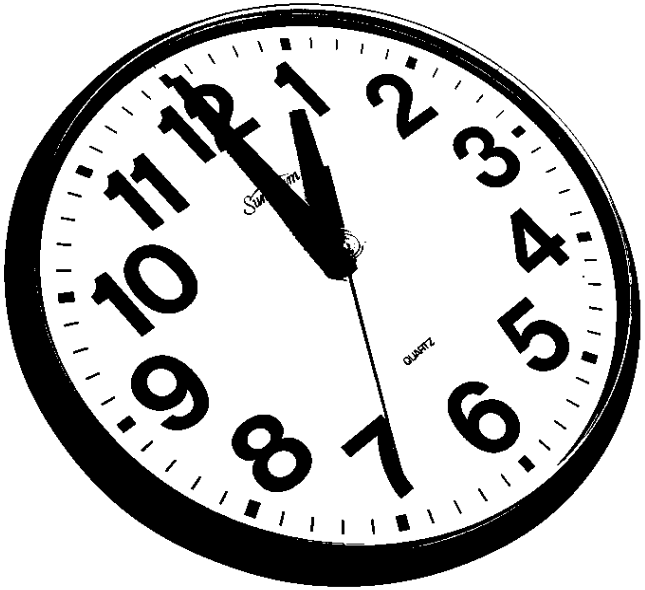
blurred

step

Median Filtering of Binary Images

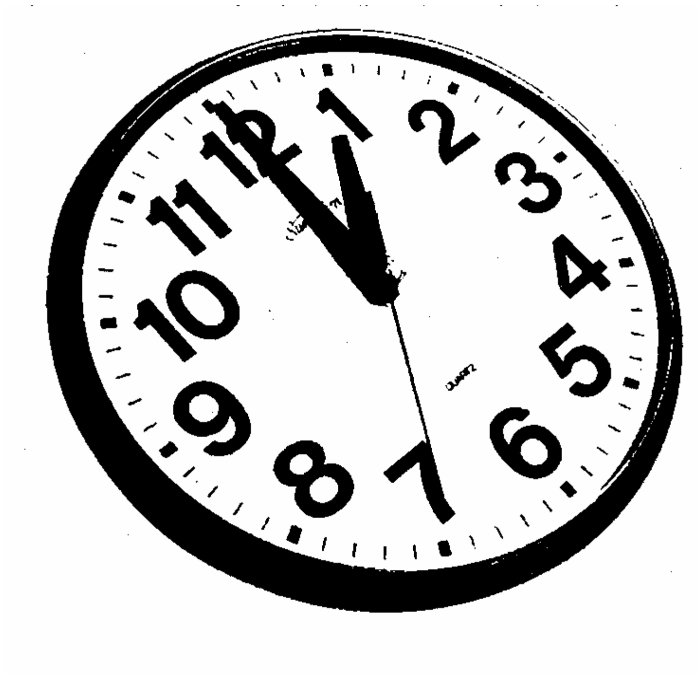


Noisy

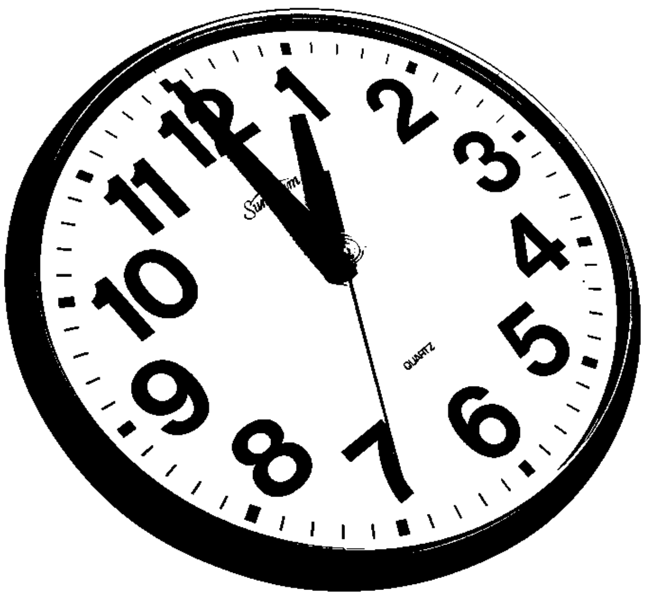


Original

Median Filtering of Binary Images

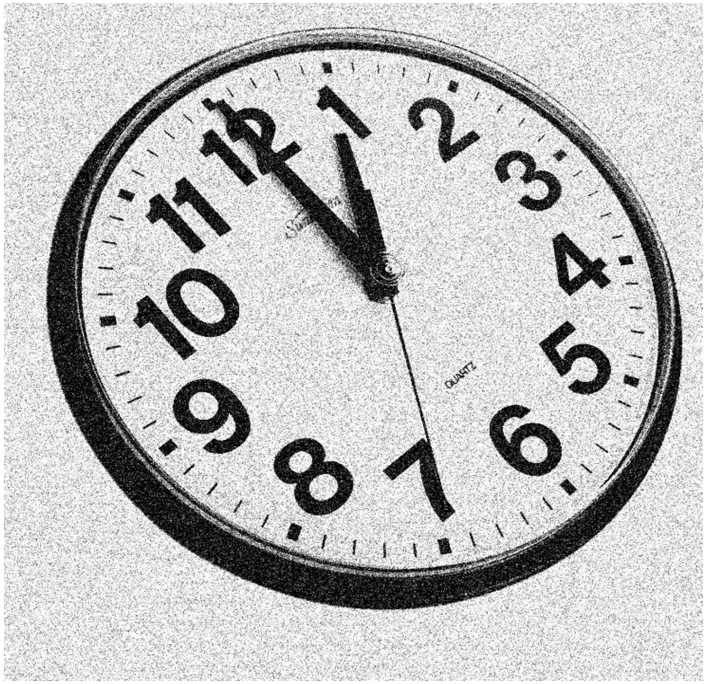


Median Filtered Noisy



Original

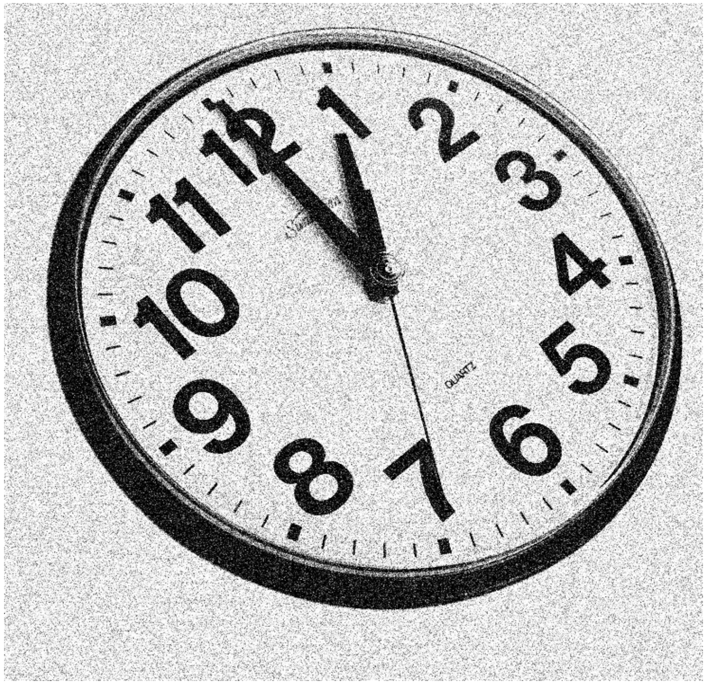
Filtering of Grayscale Images



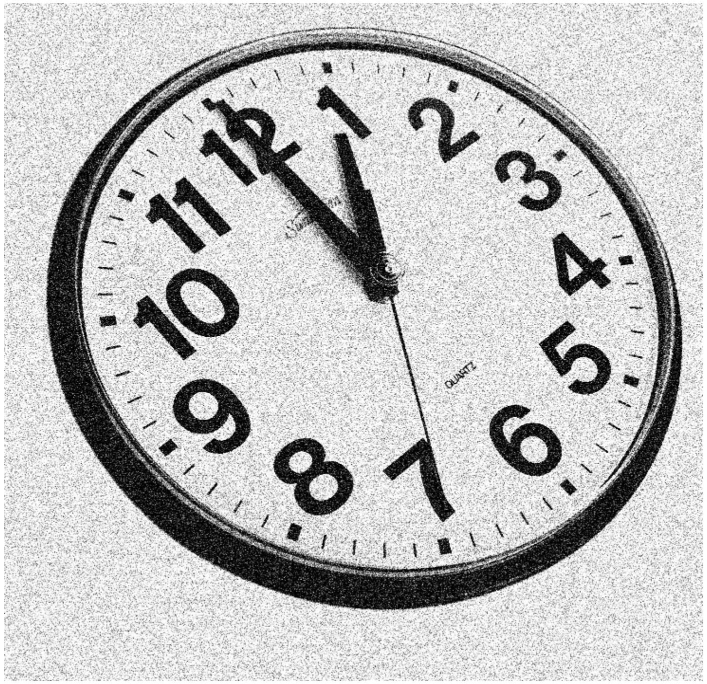
Noisy

Original

Filtering of Grayscale Images

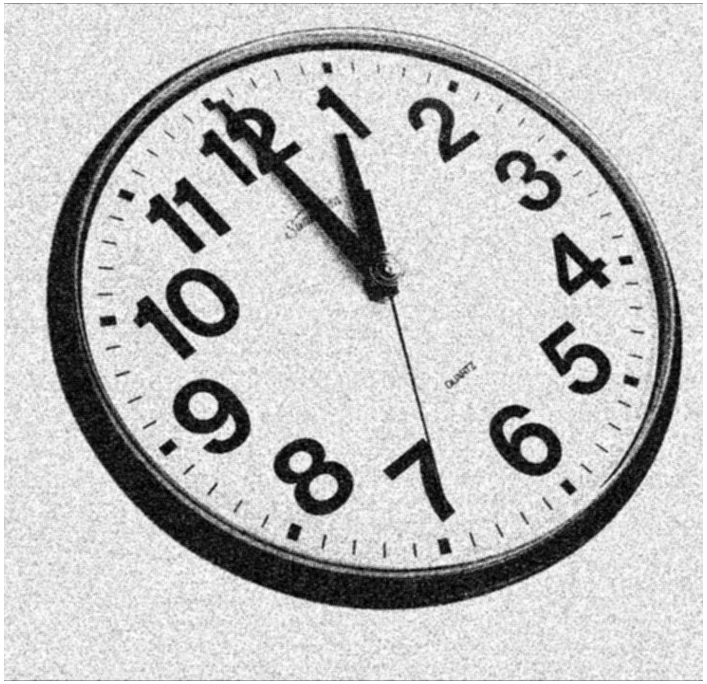


Noisy

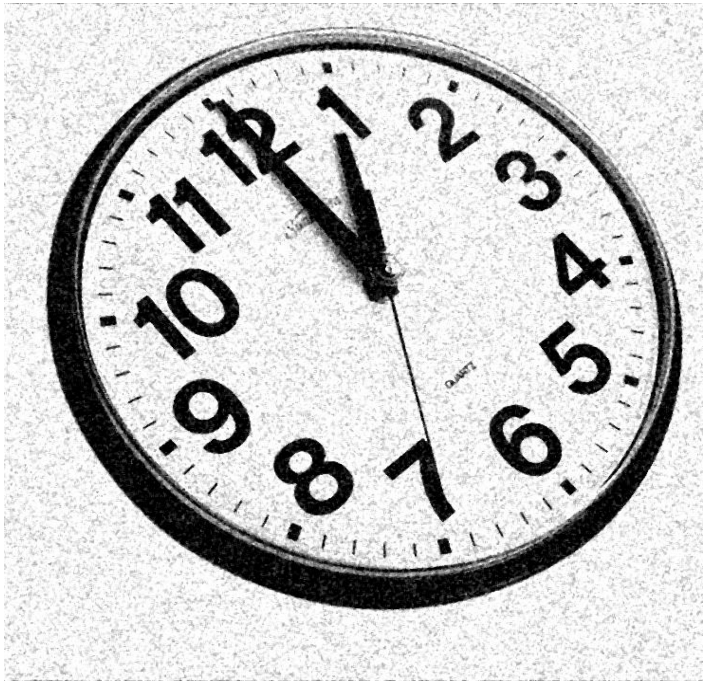


Noisy

Filtering of Grayscale Images

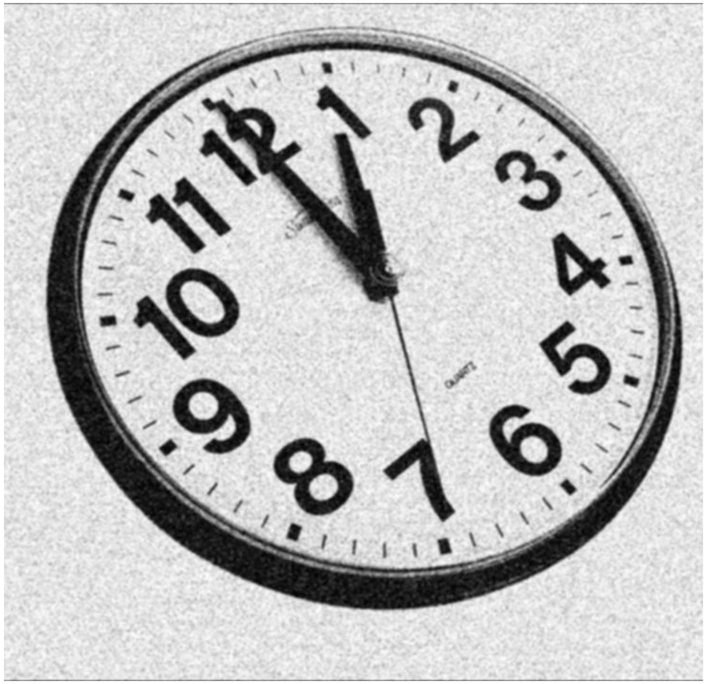


3x3-blur x 1

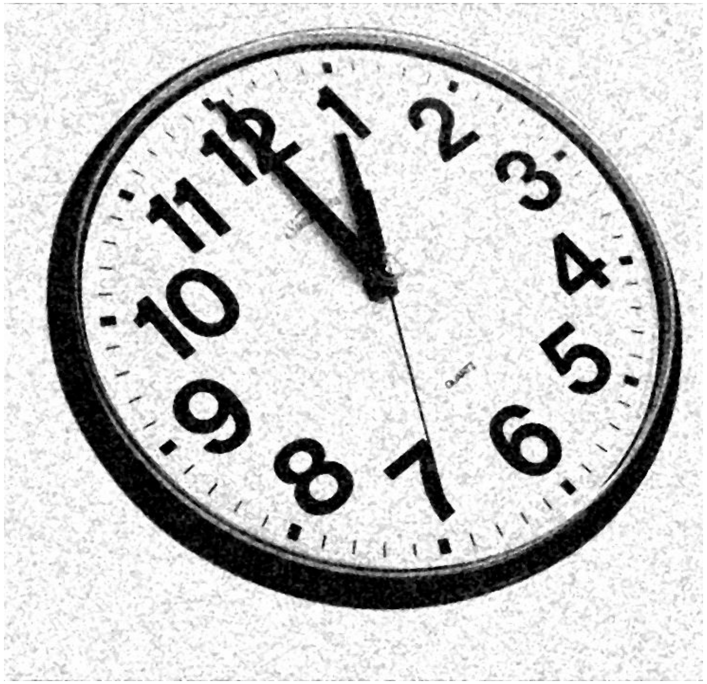


3x3-median x 1

Filtering of Grayscale Images

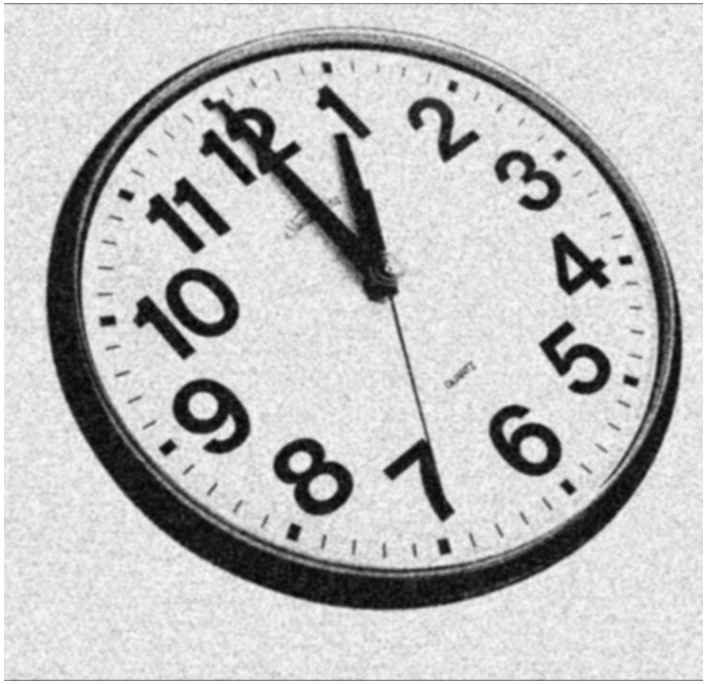


3x3-blur x 2

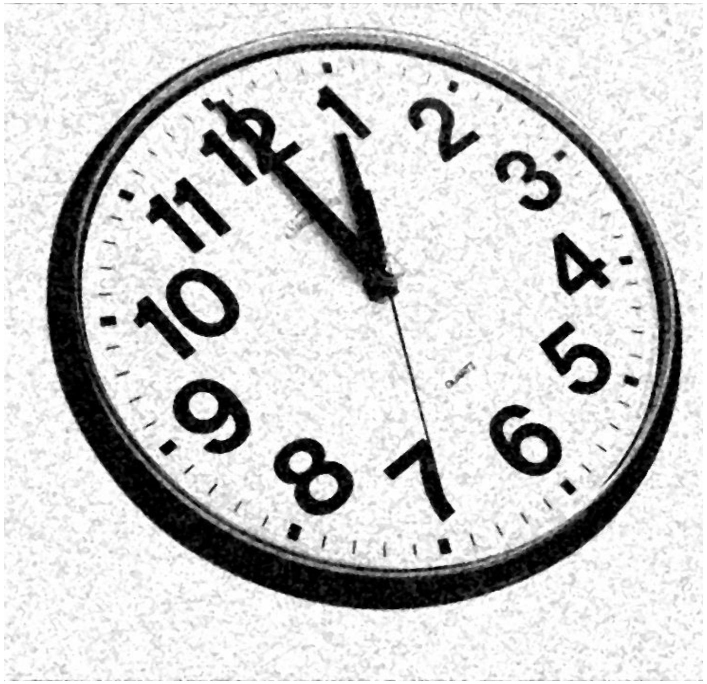


3x3-median x 2

Filtering of Grayscale Images

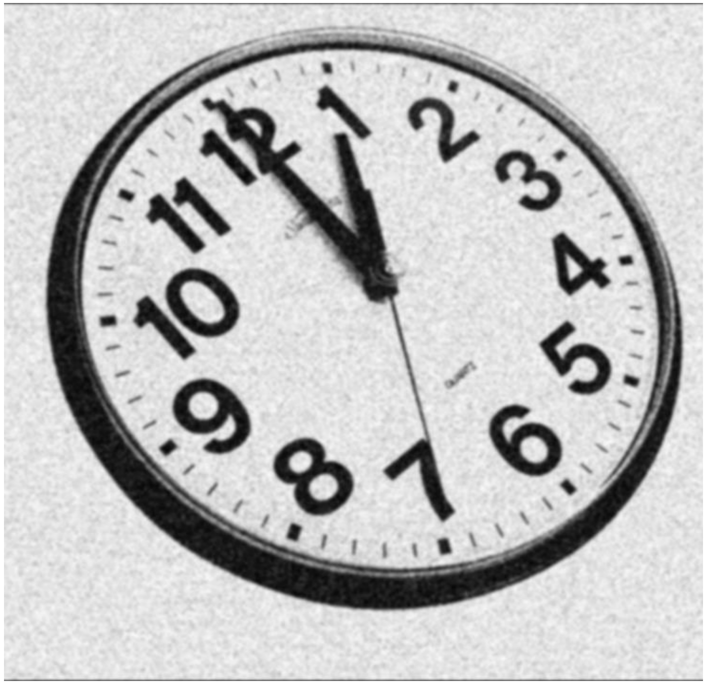


3x3-blur x 3

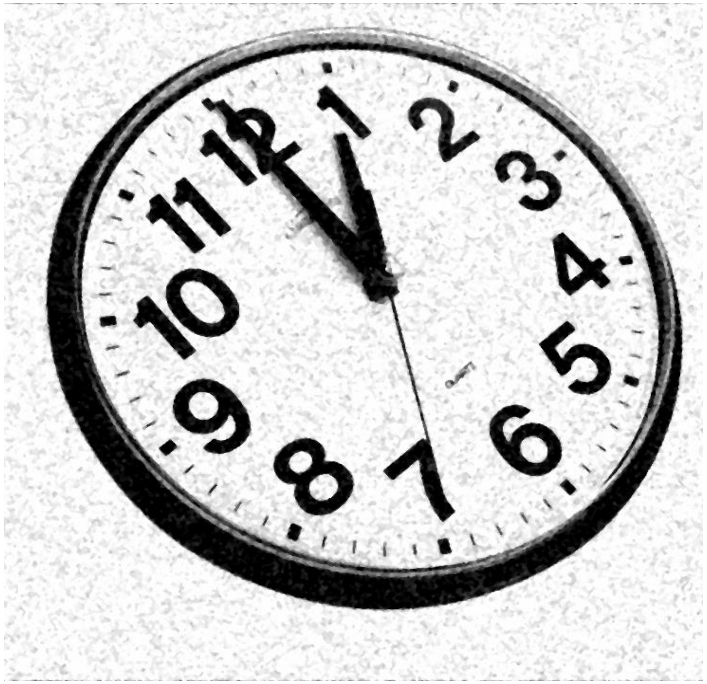


3x3-median x 3

Filtering of Grayscale Images

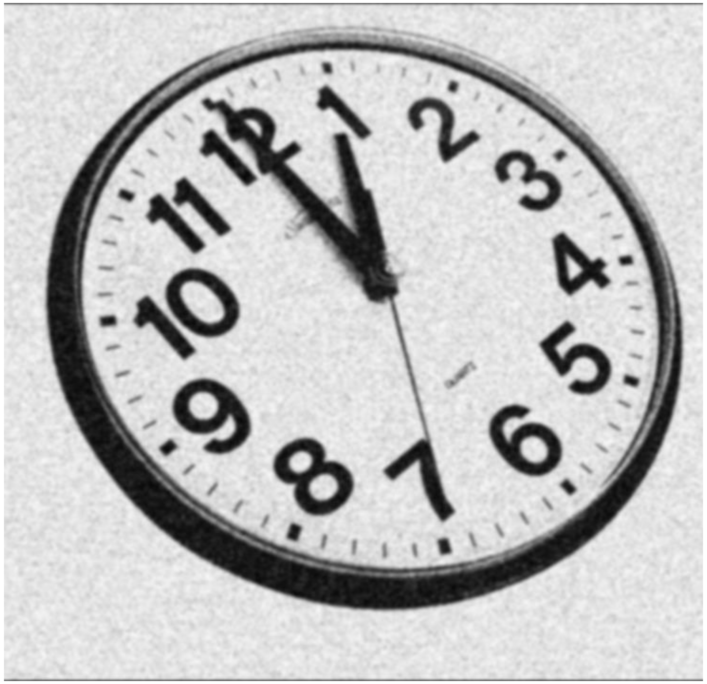


3x3-blur x 4

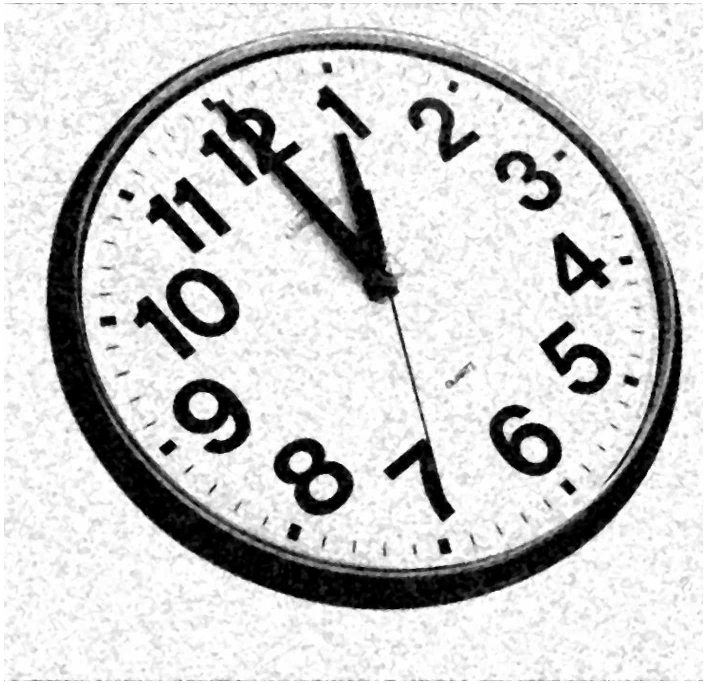


3x3-median x 4

Filtering of Grayscale Images



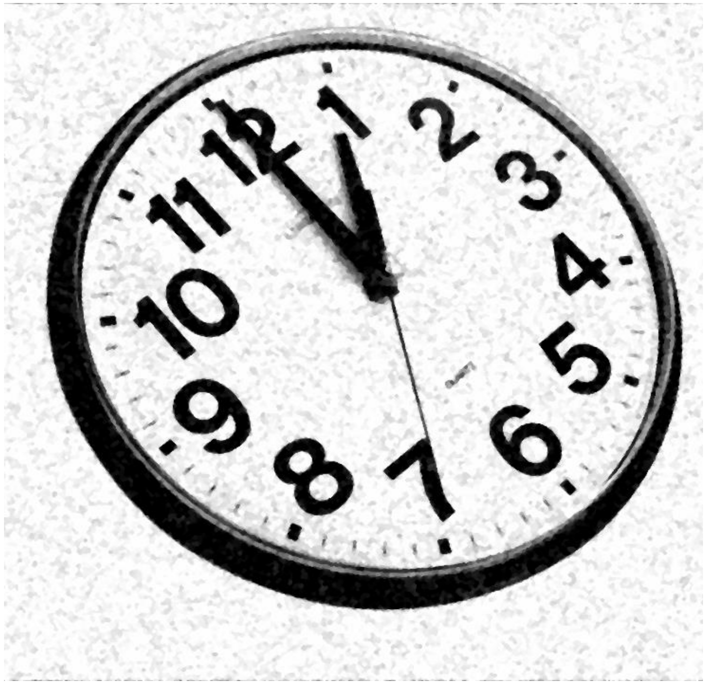
3x3-blur x 5



3x3-median x 5

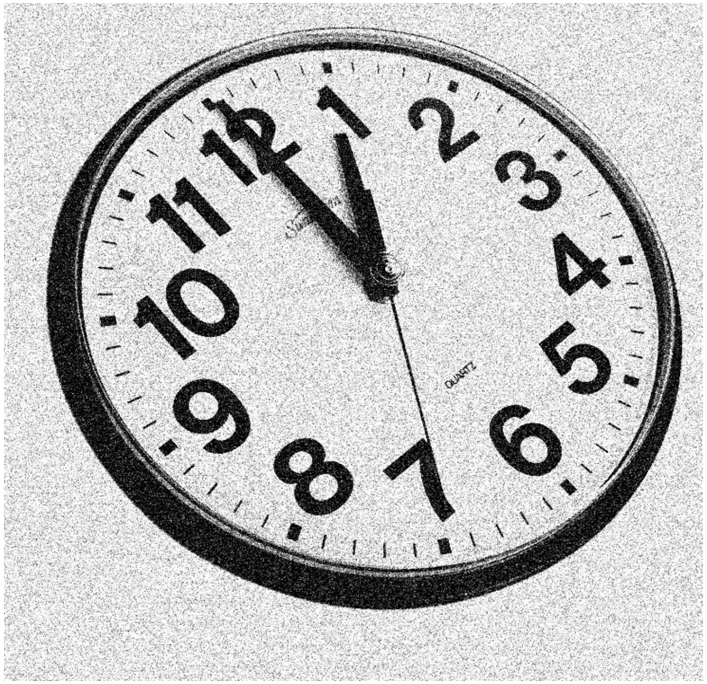
Filtering of Grayscale Images

3x3-blur x 10

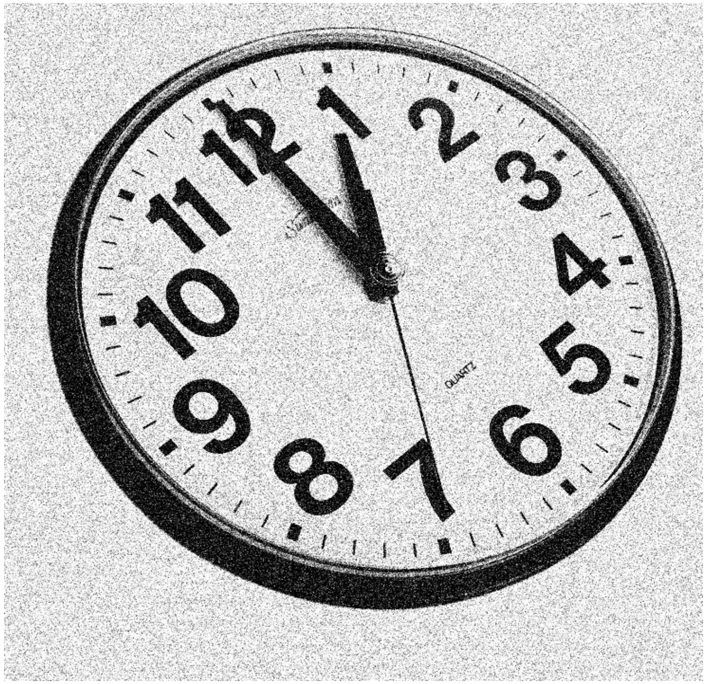


3x3-median x 10

Filtering of Grayscale Images



Noisy



Noisy

Limit and Root Images

Fact: if you repeatedly filter an image with the same blurring filter or median filter, eventually the output does not change. That is, let

$$\mathbf{I}[*\mathbf{h}]^k \equiv \left(\left(\left(\mathbf{I} * \mathbf{h}\right) * \mathbf{h}\right) \cdots * \mathbf{h}\right), \quad k \text{ times, and}$$

$$\mathbf{I}[\text{med } \mathbf{Z}]^k \equiv \left(\left(\left(\mathbf{I} \text{ med } \mathbf{Z}\right) \text{ med } \mathbf{Z}\right) \cdots \text{med } \mathbf{Z}\right), \quad k \text{ times.}$$

Then

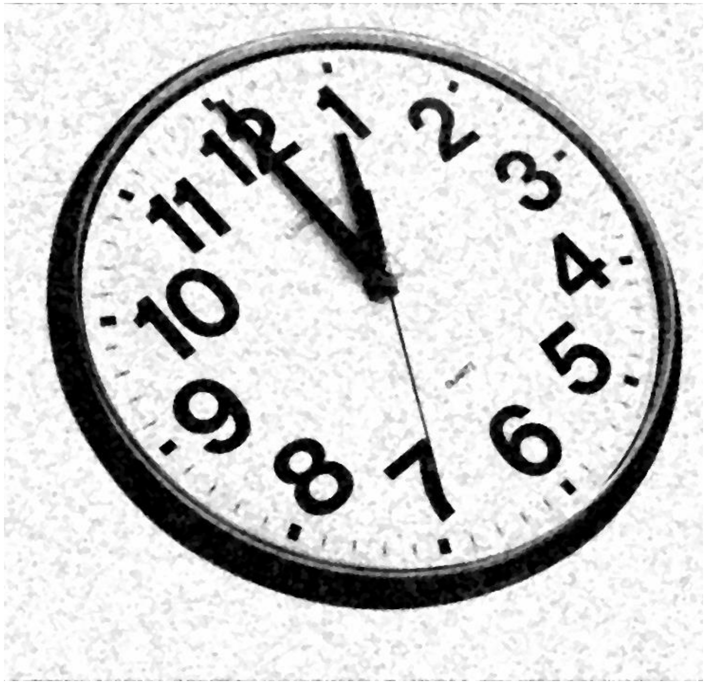
$$\lim_{k\to\infty} \mathbf{I}[*\mathbf{h}]^k = \mathbf{I}[*\mathbf{h}]^n = \mathbf{I}_0, \quad \text{and}$$

$$\lim_{k\to\infty} \mathbf{I}[\text{med } \mathbf{Z}]^k = \mathbf{I}[\text{med } \mathbf{Z}]^m = \mathbf{I}_\mathrm{r},$$

where n and m are integers $(< \infty)$, $\mathbf{I}_0$ is a single-valued image and $\mathbf{I}_\mathrm{r}$ is called the *median root* of $\mathbf{I}$.

Limit and Root Images

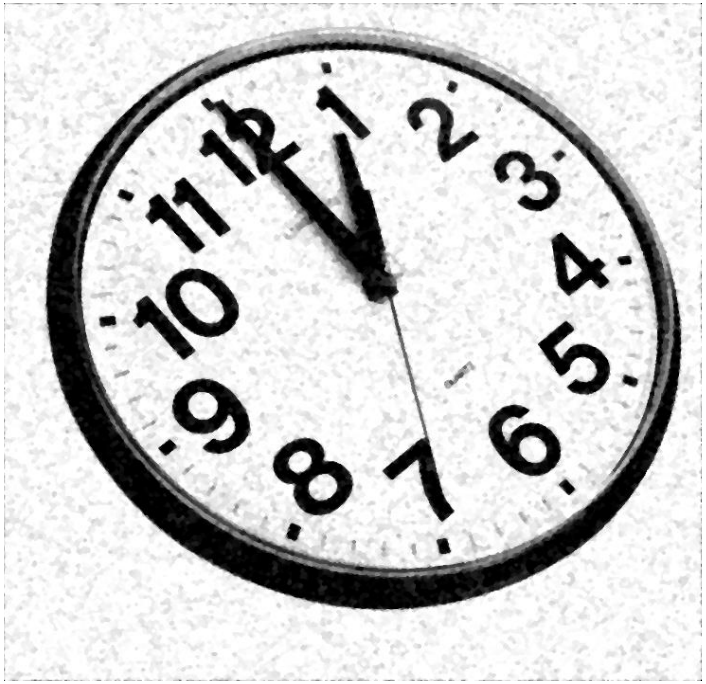
3x3-blur x 10



3x3-median x 10

Limit and Root Images

3x3-blur x n→∞



3x3-median root

Median Filter Algorithm in Matlab

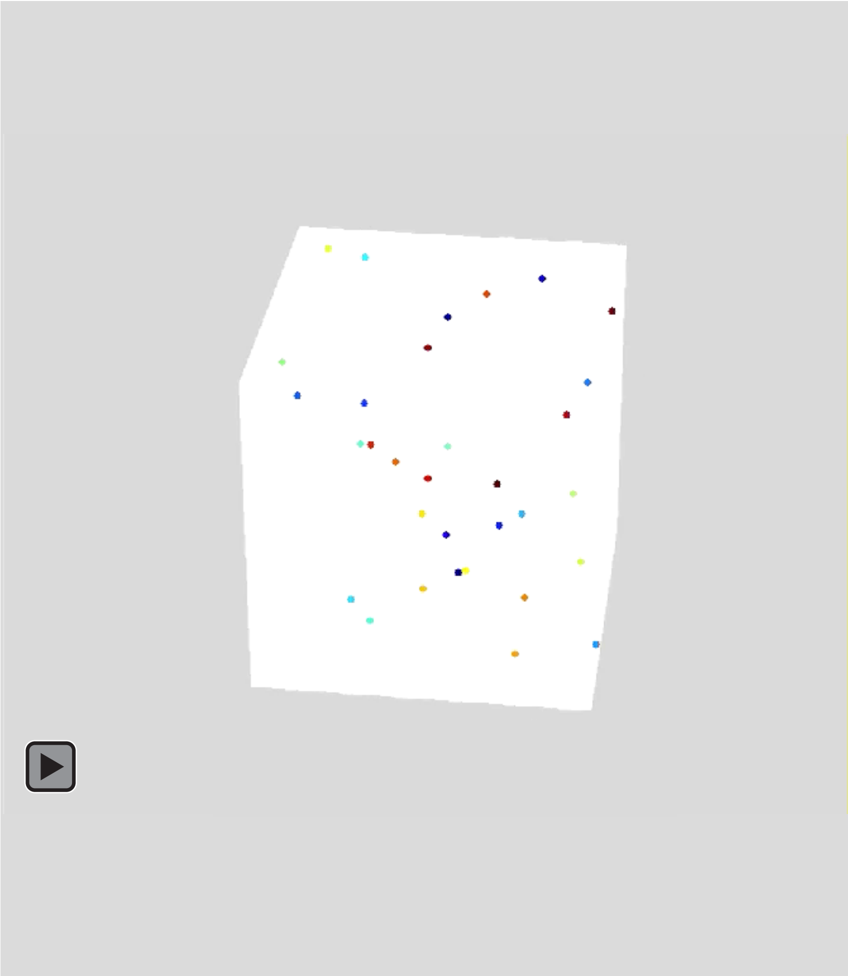
```
function D = median_filt(I,SE,origy,origx)
[R,C] = size(I);      % assumes 1-band image
[SER,SEC] = size(SE); % SE < 0 ⇒ not in nbhd

N = sum(sum(SE>=0));  % no. of pixels in nbhd
A = -ones(R+SER-1,C+SEC-1,N); % accumulator
n=1; % copy I into band n of A for nbhd pix n
for j = 1 : SER % neighborhood is def’d in SE
   for i = 1 : SEC
      if SE(j,i) >= 0 % then is a nbhd pixel
         A(j:(R+j-1),i:(C+i-1),n) = I;
         n=n+1; % next accumulator band
      end
   end
end
% pixel-wise median across the bands of A
A = shiftdim(median(shiftdim(A,2)),1);
D = A( origy:(R+origy-1) , origx:(C+origx-1) );
return;
```

Vector Median Filter

A vector median filter selects from among a set of vectors, the one vector that is closest to all the others.

That is, if S is a set of vectors, in $\mathbb{F}^n$ the median, $\bar{\mathbf{v}}$, is♠

$$\bar{\mathbf{v}} = \arg\min_{k \neq j} \left\{ \left\| \mathbf{v}_k - \mathbf{v}_j \right\| \, \middle| \, \mathbf{v}_k, \mathbf{v}_j \in S \right\}.$$



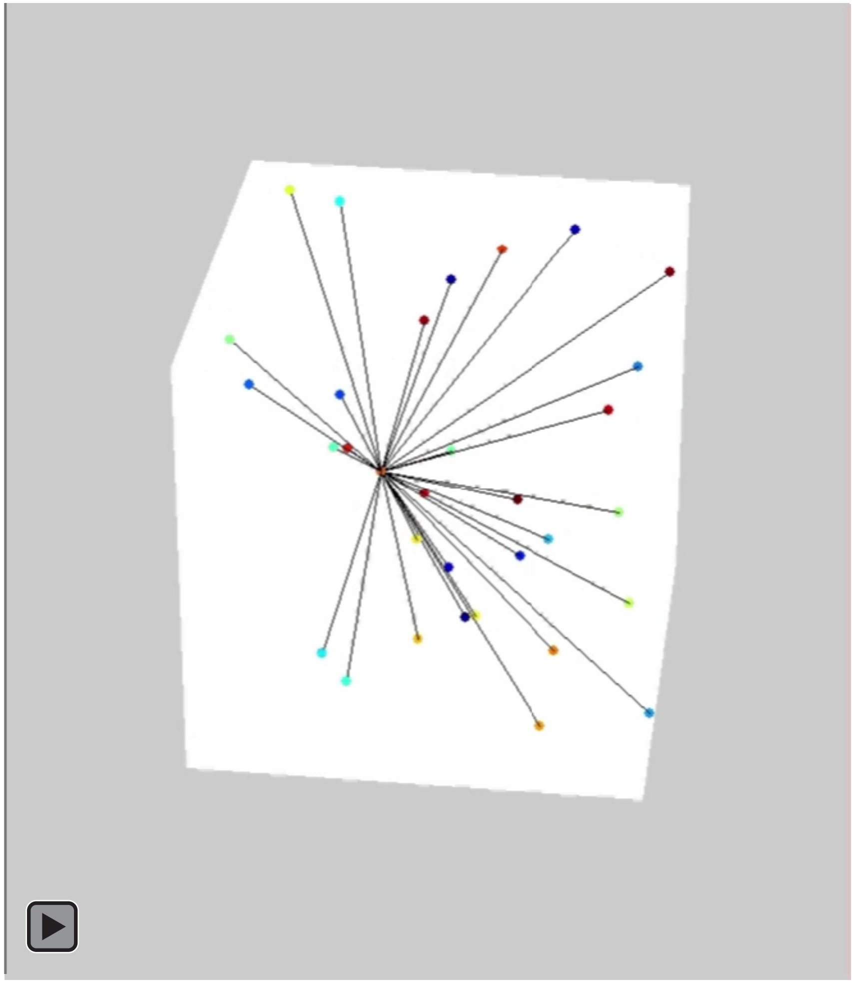
♠$\mathbb{F}^n$ is an n-dimensional linear vector space over the field, $\mathbb{F}$.)

Vector Median Filter

A vector median filter selects from among a set of vectors, the one vector that is closest to all the others.

That is, if S is a set of vectors, in $\mathbb{F}^n$ the median, $\bar{\mathbf{v}}$, is♠

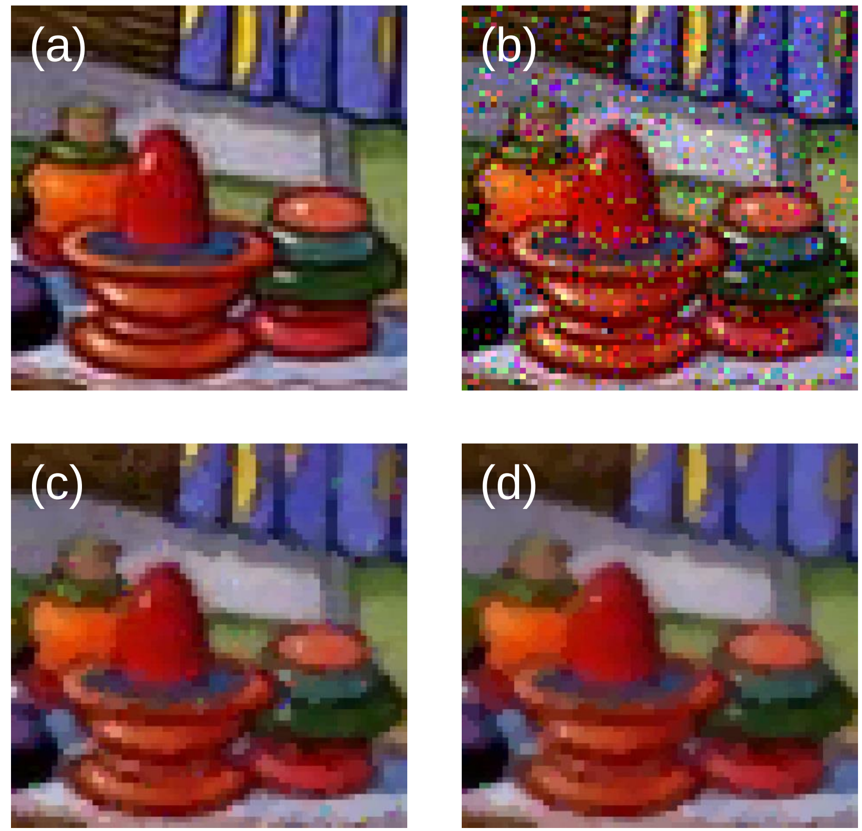
$$\bar{\mathbf{v}} = \underset{k \neq j}{\arg\min} \left\{ \left\| \mathbf{v}_k - \mathbf{v}_j \right\| \,\middle|\, \mathbf{v}_k, \mathbf{v}_j \in S \right\}.$$



♠$\mathbb{F}^n$ is an n-dimensional linear vector space over the field, $\mathbb{F}$.)

Color Median Filter

If we let $\mathbb{F}^n = \mathbb{R}^3$ then the vector median can be used as a color median filter.



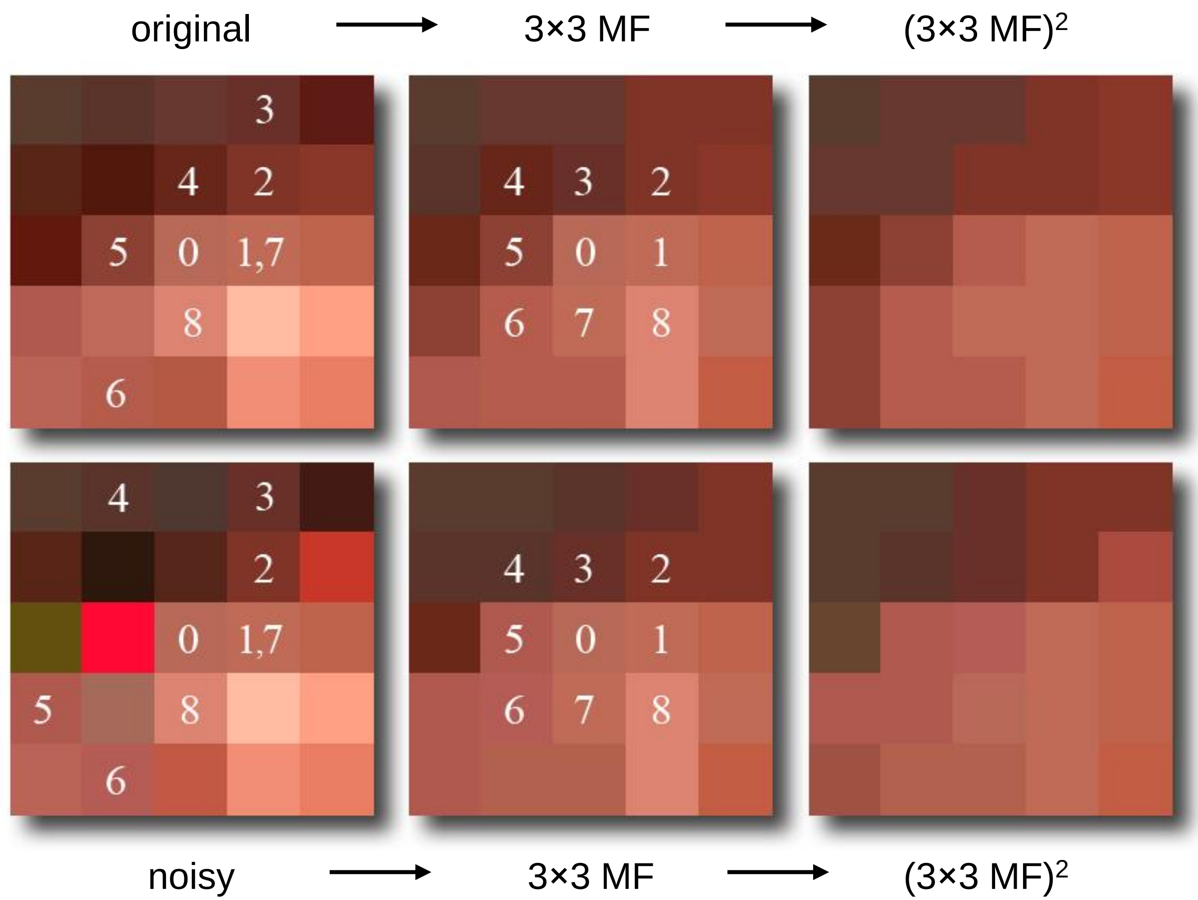
(a) original image
(b) image (a) with sparse noise
(c) image (b) color median filtered
(d) image (c) color median filtered

Median filter performed on 3×3 nbhd.

original image: Jim Woodring
www.jimwoodring.com

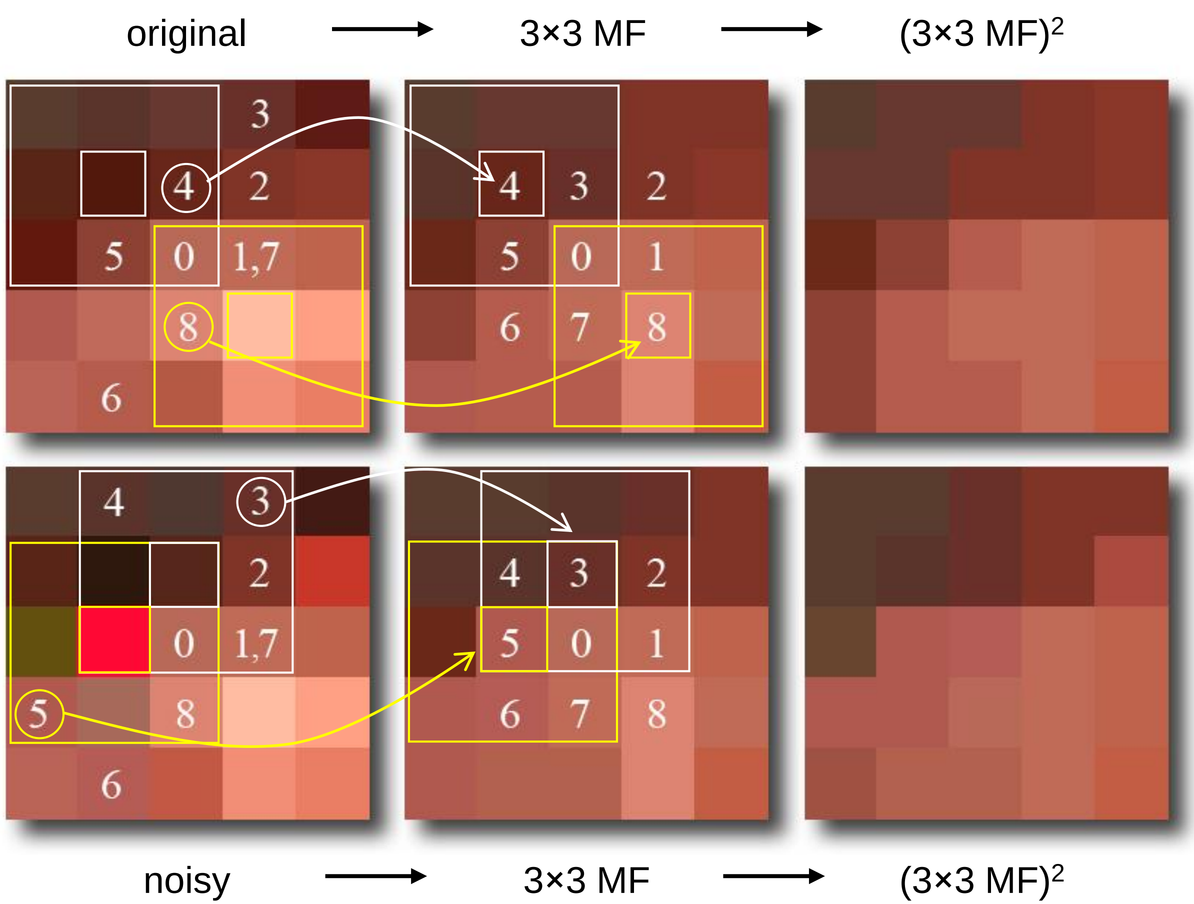
Color Median Filter

The output color at (r,c) is always selected from a nbhd of (r,c) in the input image.



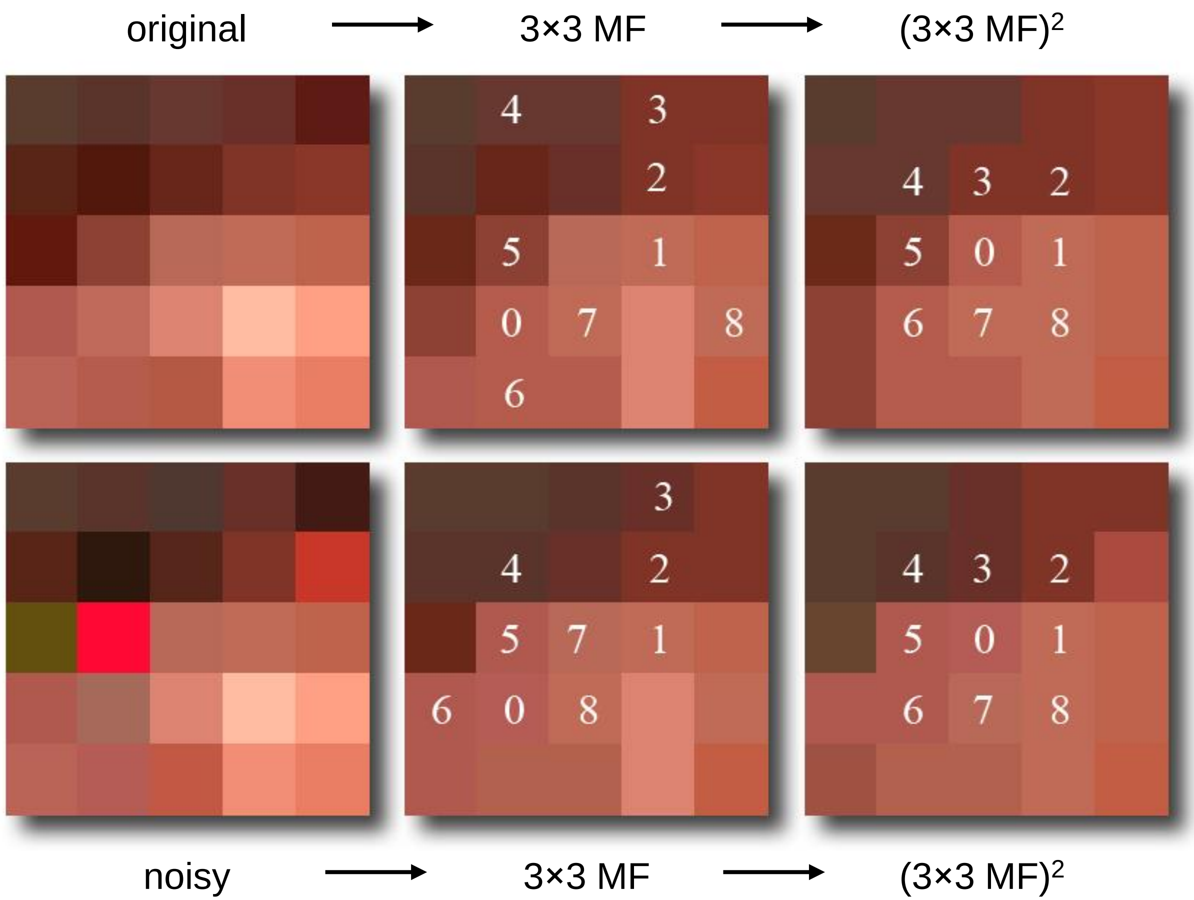
The output color at (r,c) is always selected from a nbhd of (r,c) in the input image.

Color Median Filter



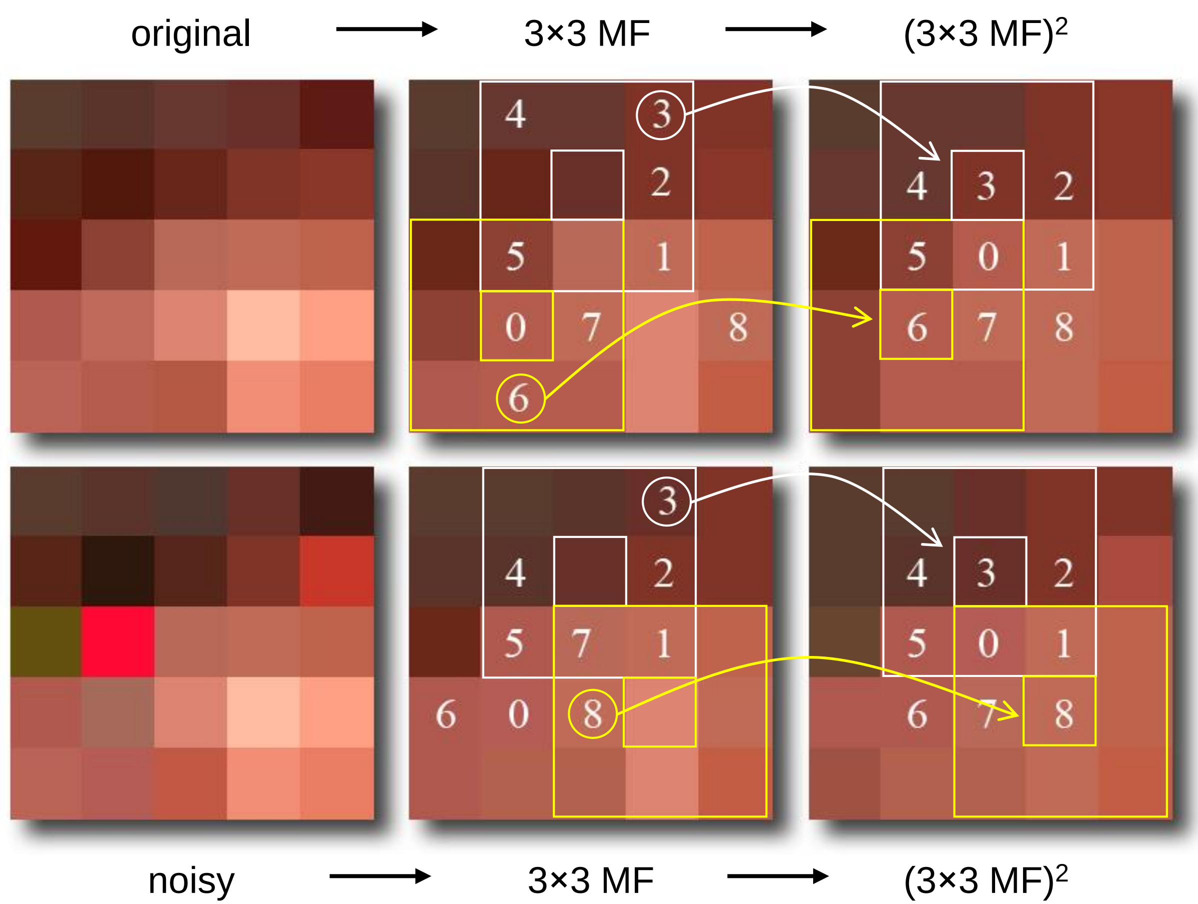
The output color at (r,c) is always selected from a nbhd of (r,c) in the input image.

Color Median Filter



Color Median Filter

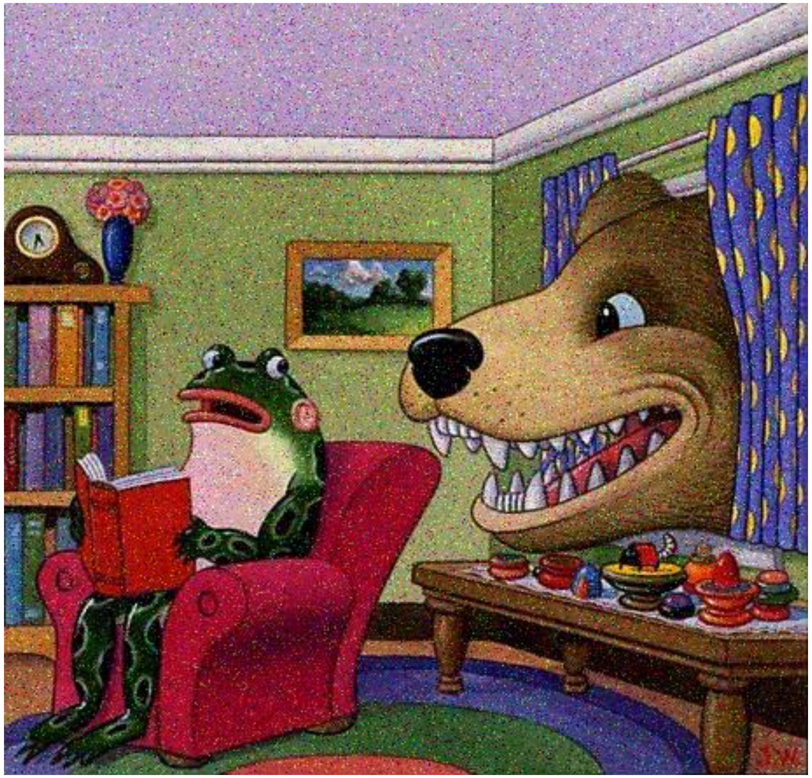
The output color at (r,c) is always selected from a nbhd of (r,c) in the input image.



Color Median Filter

Jim Woodring – A Warm Shoulder

www.jimwoodring.com



Sparse noise, 32% coverage in each band

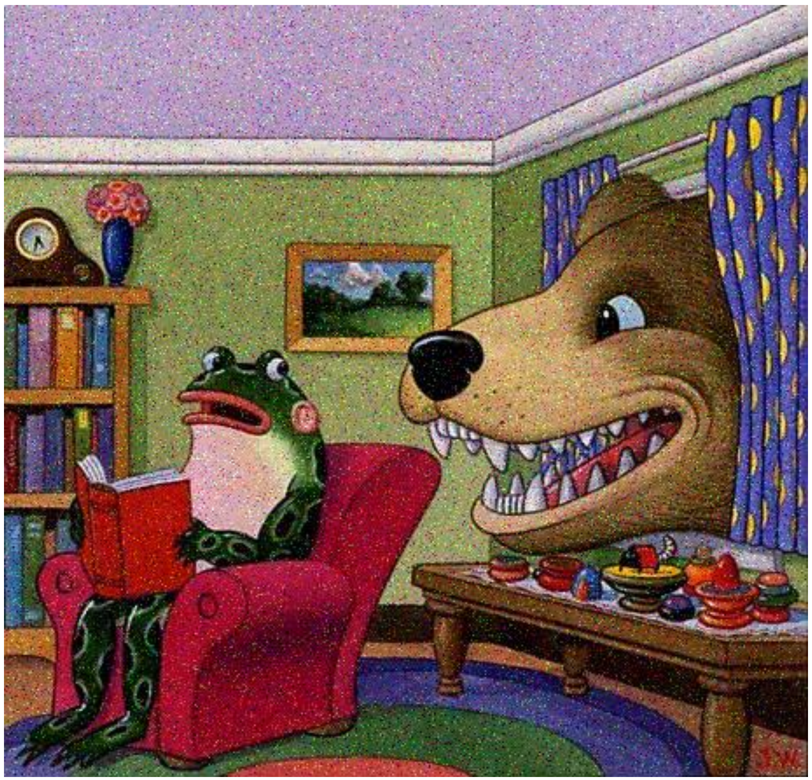
Color Median Filter

3×3 color median filter applied once

3×3 color median filter applied twice

original image: Jim Woodring
www.jimwoodring.com

Color Median Filter



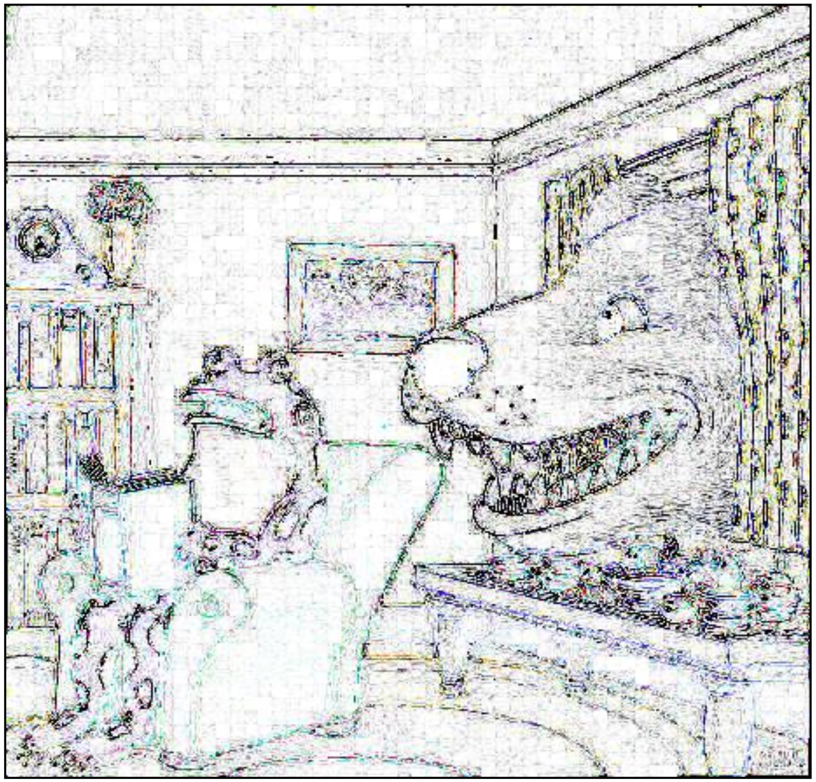
Sparse noise, 32% coverage in each band

Jim Woodring – A Warm Shoulder

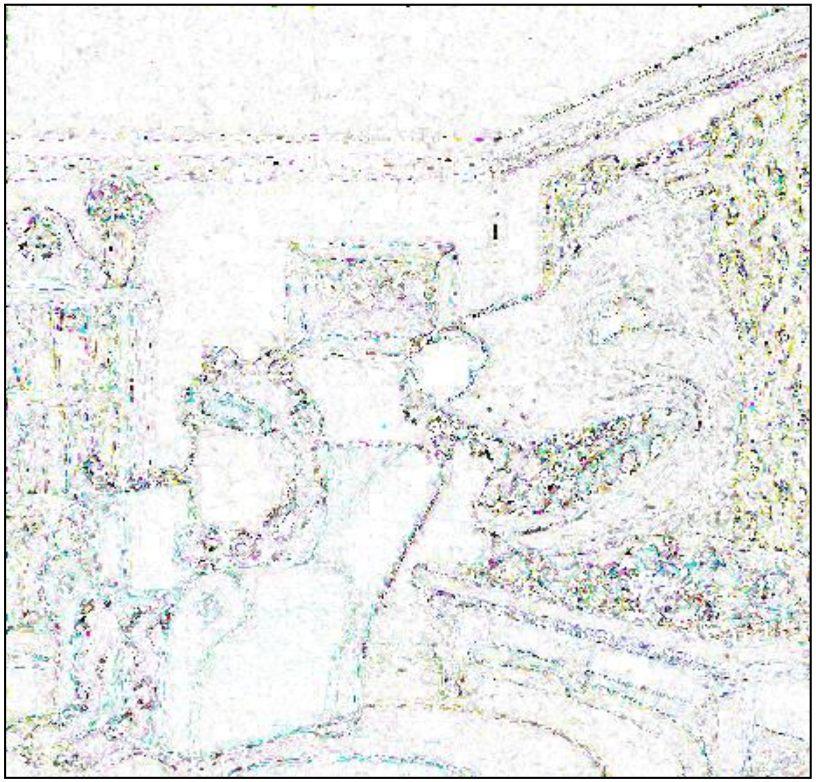
www.jimwoodring.com

Color Median Filter

Absolute differences displayed as negatives to enhance visibility



(3×3 CMF² of noisy) – original



(3×3 CMF² of noisy) – (3×3 CMF² of original)

original image: Jim Woodring
www.jimwoodring.com

CMF vs. Standard Median on Individual Bands

A color median filter has to compute the distances between all the color vectors in the neighborhood of each pixel. That's expensive computationally.

Q: Why not simply take the 1-band median of each color band individually?

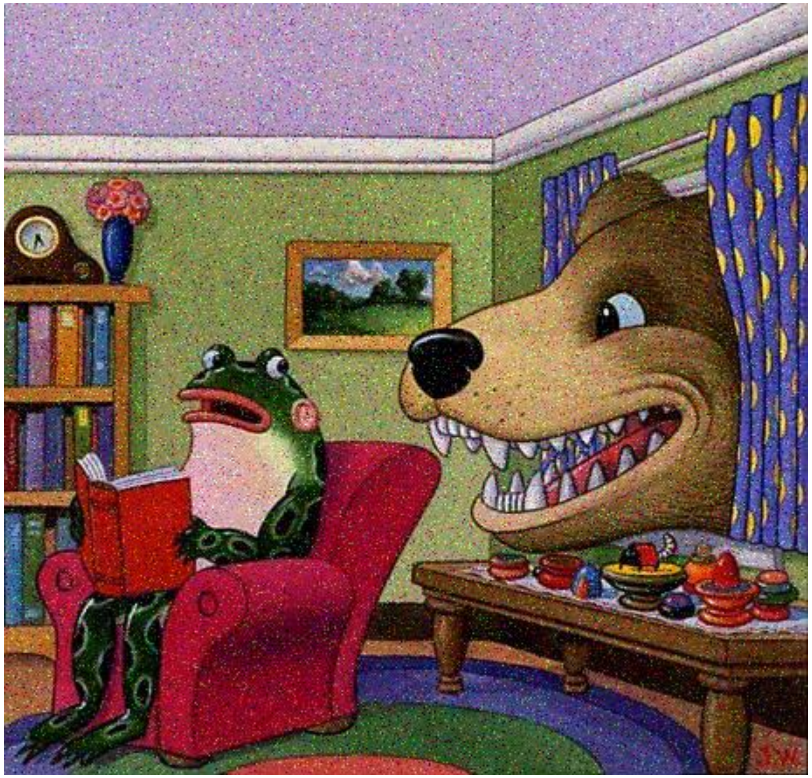
A: The result at a pixel could be a color that did not exist in the pixel's neighborhood in the input image. The result is not the median of the colors – it is the median of the intensities of each color band treated independently.

Q: Is that a problem?

A: Maybe. Maybe not. It depends on the application. It may make little difference visually. If the colors need to be preserved, it could be problematic.

CMF vs. Standard Median on Individual Bands

Jim Woodring – A Warm Shoulder
www.jimwoodring.com



Sparse noise, 32% coverage in each band

CMF vs. Standard Median on Individual Bands

3×3 color median filter applied once

3×3 color median filter applied twice

original image: Jim Woodring
www.jimwoodring.com

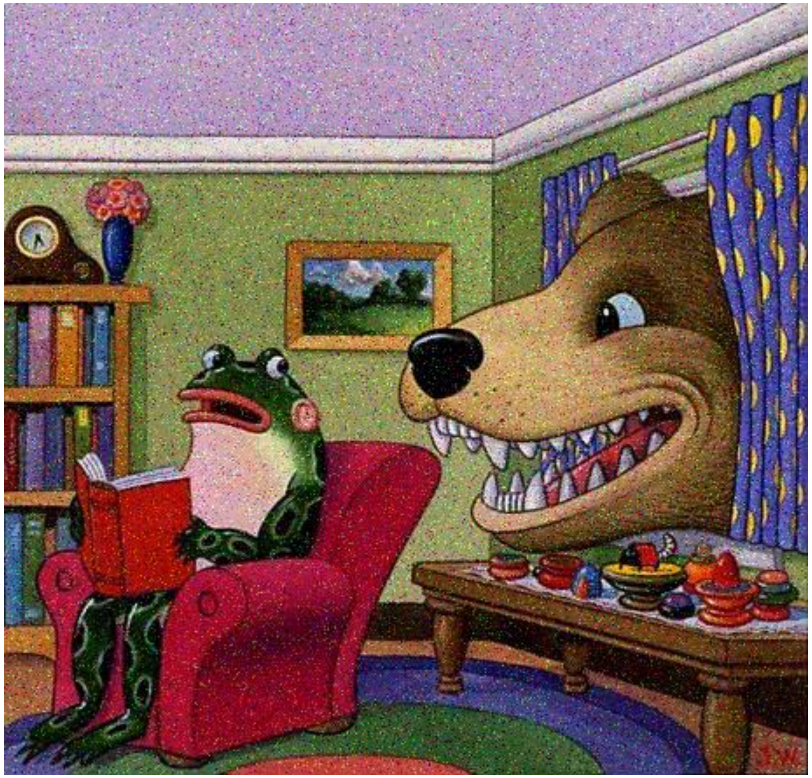
CMF vs. Standard Median on Individual Bands

3×3 median filter applied to each band once

3×3 median filter applied to each band twice

original image: Jim Woodring
www.jimwoodring.com

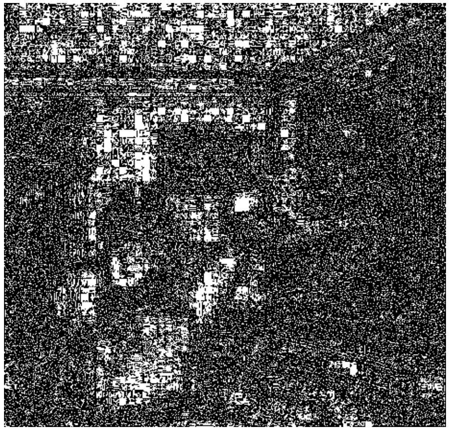
CMF vs. Standard Median on Individual Bands



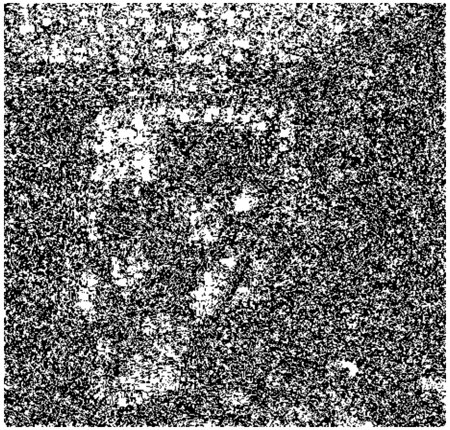
Sparse noise, 32% coverage in each band

Jim Woodring – A Warm Shoulder
www.jimwoodring.com

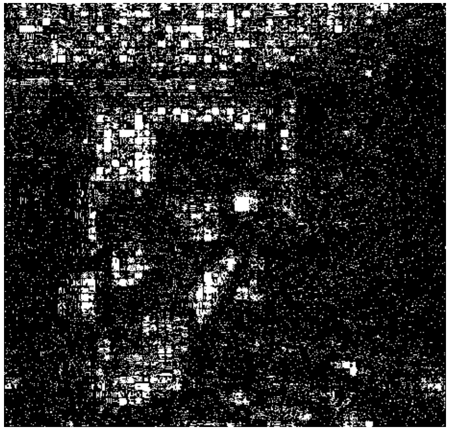
CMF vs. Standard Median on Individual Bands



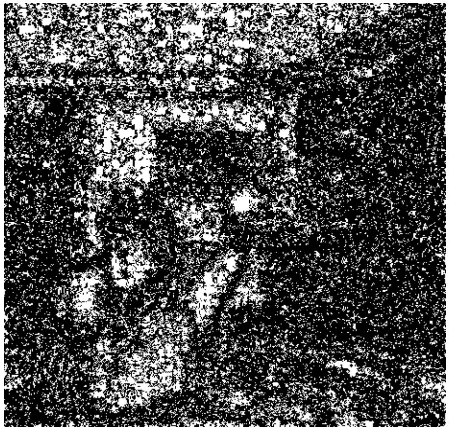
Fraction of pixels in CMF^2 noisy image identical to original: 0.29



Fraction of pixels in CMF^2 noisy image identical to CMF^2 original: 0.43



Fraction of pixels in MF^2 noisy image identical to original: 0.14



Fraction of pixels in MF^2 noisy image identical to MF^2 original: 0.28

original image: Jim Woodring
www.jimwoodring.com