Denoising Filters

Baba C. Vemuri

Professor, Department of Computer & Information Science and Engineering
University of Florida
Image Denoising

Image Processing (Chapter 3 Szeliski; Chapter 6,7,8 from BKP Horn)
Image Denoising

Images invariably contain noise, how can we reduce this noise?

Figure: Take several images and average them (Source: S. Sietz)
How about weighted averaging? How do we do that?

- Replace each pixel with a weighted average of its neighbors.

- Weights define the kernel/mask of the filter. They should sum to 1. Why?

- What does the unweighted averaging filter look like?
Let $f$ be the image and $g$ be the filter kernel. Discrete convolution is given by,

$$(f \otimes g)(m, n) = \sum_{k,l} f(m-k, n-l)g(k, l).$$

Convention, filter is flipped. MATLAB: conv2 (also imfilter).

"box filter"

**Figure:** Averaging filter mask & Convolution
Annoying details

- What is the size of the output?
- MATLAB: `conv2(f, g, shape)
  - `shape = 'full': output size is sum of sizes of `f` and `g`
  - `shape = 'same': output size is same as `f`
  - `shape = 'valid': output size is difference of sizes of `f` and `g`
Gaussian Kernel

\[ G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \]

- Constant factor at front makes volume sum to 1 (can be ignored, as we should re-normalize weights to sum to 1 in any case)

Source: C. Rasmussen
Choosing kernel width

- Rule of thumb: set filter half-width to about 3 \( \sigma \)
Mean vs. Gaussian filtering
Separability of the Gaussian filter

\[ G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \]

\[ = \left( \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) \right) \left( \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right) \right) \]

The 2D Gaussian can be expressed as the product of two functions, one a function of \( x \) and the other a function of \( y \).

In this case, the two functions are the (identical) 1D Gaussian.

Source: D. Lowe
Separability example

2D convolution (center location only)

The filter factors into a product of 1D filters:

Perform convolution along rows:

Followed by convolution along the remaining column:

For MN image, PQ filter: 2D takes MNPQ add/times, while 1D takes MN(P + Q)

Source: K. Grauman
Alternative idea: Median filtering

- A **median filter** operates over a window by selecting the median intensity in the window.

```
<table>
<thead>
<tr>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>90</td>
<td>27</td>
</tr>
<tr>
<td>33</td>
<td>31</td>
<td>30</td>
</tr>
</tbody>
</table>
```

Median value: 27

```
<table>
<thead>
<tr>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>33</td>
<td>31</td>
<td>30</td>
</tr>
</tbody>
</table>
```

Sort: 31, 33, 90
Replace:

- Is median filtering linear?

Source: K. Grauman
Median filter

Replace each pixel by the median over N pixels (5 pixels, for these examples). Generalizes to “rank order” filters.

\[
\text{Median}([1 \ 7 \ 1 \ 5 \ 1]) = 1 \\
\text{Mean}([1 \ 7 \ 1 \ 5 \ 1]) = 2.8
\]

Spike noise is removed

Monotonic edges remain unchanged
Median vs. Gaussian filtering

3x3

Gaussian

5x5

Median

7x7