

[◀ Back to Week 3](#)[✕ Lessons](#)[Prev](#)[Next](#)

- Implement a histogram equalization function. If using Matlab, compare your implementation with Matlab's built-in function.
- Implement a median filter. Add different levels and types of noise to an image and experiment with different sizes of support for the median filter. As before, compare your implementation with Matlab's.
- Implement the non-local means algorithm. Try different window sizes. Add different levels of noise and see the influence of it in the need for larger or smaller neighborhoods. (Such block operations are easy when using Matlab, see for example the function at <http://www.mathworks.com/help/images/ref/blockproc.html>). Compare your results with those available in IPOL as demonstrated in the video lectures.

1. Consider an image and add to it random noise. Repeat this N times, for different values of N, and add the resulting images. What do you observe?

- Implement the basic color edge detector. What happens when the 3 channels are equal?
- Take a video and do frame-by-frame histogram equalization and run the resulting video. Now consider a group of frames as a large image and do histogram equalization for all of them at once. What looks better? See this example on how to read and handle videos in Matlab:

```

1  xyloObj = VideoReader('xylophone.mp4');
2  nFrames = xyloObj.NumberOfFrames;
3
4  vidHeight = xyloObj.Height;
5  vidWidth = xyloObj.Width;
6
7  % Preallocate movie structure.
8  mov(1:nFrames) = struct('cdata', zeros(vidHeight, vidWidth, 3, 'uint8'),
9      'colormap', []);
10
11 % Read one frame at a time.
12 for k = 1 : nFrames
13     im = read(xyloObj, k);
14     % here we process the image im
15     mov(k).cdata = im;
16 end
17
18 % Size a figure based on the video's width and height.
19 hf = figure;
20 set(hf, 'position', [150 150 vidWidth vidHeight])
21
22 % Play back the movie once at the video's frame rate.
23 movie(hf, mov, 1, xyloObj.FrameRate);

```

- Take a video and do frame-by-frame non-local means denoising. Repeat but now using a group of frames as a large image. This allows you for example to find more matching blocks (since you are searching across frames). Compare the results. What happens if now you use 3D spatio-temporal blocks, e.g., $5 \times 5 \times 3$ blocks and consider the group of frames as a 3D image? Try this and compare with previous results.
- Search for “camouflage artist Liu Bolin.” Do you think you can use the tools you are learning to detect him?

Mark as completed

