

### OpenCL median filter code execution time for a kernel size of 3.

```
C:\Python27\Scripts>med.py
[<pyopencl.Device 'Intel(R) Core(TM) i7-3770S CPU @ 3.10GHz' on 'Intel(R) OpenCL' at 0x17a398c0>, <pyopencl.Device 'Intel(R) HD Graphics 4000' on 'Intel(R) OpenCL' at 0x11a6b80>]
C:\Python27\lib\site-packages\pyopencl\cffi_cl.py:1521: CompilerWarning: Non-empty compiler output encountered. Set the environment variable PYOPENCL_COMPILER_OUTPUT=1 to see more.
  "to see more.", CompilerWarning)
('Time: ', 0.015000104904174805)
Now try Numpy
C:\Python27\Scripts>
```

OpenCL time: 0.015 seconds.

The OpenCL (open computing language) framework uses the GPU for hardware acceleration, and, in our case, it improves the execution time of the median filter. Comparing the execution time of the OpenCL algorithm vs the NumPy algorithm one can see that the OpenCL algorithm is almost 42 times faster. NumPy uses vectorised computation resulting in fast execution on the CPU.

### NumPy median filter code execution time for different kernel sizes. `scipy.signal.medfilt()`

```
C:\Python27\Scripts>med.py
[<pyopencl.Device 'Intel(R) Core(TM) i7-3770S CPU @ 3.10GHz' on 'Intel(R) OpenCL' at 0x178498c0>, <pyopencl.Device 'Intel(R) HD Graphics 4000' on 'Intel(R) OpenCL' at 0xf73fb0>]
C:\Python27\lib\site-packages\pyopencl\cffi_cl.py:1521: CompilerWarning: Non-empty compiler output encountered. Set the environment variable PYOPENCL_COMPILER_OUTPUT=1 to see more.
  "to see more.", CompilerWarning)
('Time: ', 0.625)
Now try Numpy
C:\Python27\Scripts>
```

Kernel size=3: 0.625 seconds

```
C:\Python27\Scripts>med.py
[<pyopencl.Device 'Intel(R) Core(TM) i7-3770S CPU @ 3.10GHz' on 'Intel(R) OpenCL' at 0x177698c0>, <pyopencl.Device 'Intel(R) HD Graphics 4000' on 'Intel(R) OpenCL' at 0xf1a9c0>]
C:\Python27\lib\site-packages\pyopencl\cffi_cl.py:1521: CompilerWarning: Non-empty compiler output encountered. Set the environment variable PYOPENCL_COMPILER_OUTPUT=1 to see more.
  "to see more.", CompilerWarning)
('Time: ', 2.0470001697540283)
Now try Numpy
C:\Python27\Scripts>
```

Kernel size=5: 2.04 seconds

```
C:\Python27\Scripts>med.py
[<pyopencl.Device 'Intel(R) Core(TM) i7-3770S CPU @ 3.10GHz' on 'Intel(R) OpenCL' at 0x172098c0>, <pyopencl.Device 'Intel(R) HD Graphics 4000' on 'Intel(R) OpenCL' at 0xa17af0>]
C:\Python27\lib\site-packages\pyopencl\cffi_cl.py:1521: CompilerWarning: Non-empty compiler output encountered. Set the environment variable PYOPENCL_COMPILER_OUTPUT=1 to see more.
  "to see more.", CompilerWarning)
('Time: ', 3.6559998989105225)
Now try Numpy
C:\Python27\Scripts>
```

Kernel size=7: 3.66 seconds

```
C:\Python27\Scripts>med.py
[<pyopencl.Device 'Intel(R) Core(TM) i7-3770S CPU @ 3.10GHz' on 'Intel(R) OpenCL' at 0x172098c0>, <pyopencl.Device 'Intel(R) HD Graphics 4000' on 'Intel(R) OpenCL' at 0x74fef0>]
C:\Python27\lib\site-packages\pyopencl\cffi_cl.py:1521: CompilerWarning: Non-empty compiler output encountered. Set the environment variable PYOPENCL_COMPILER_OUTPUT=1 to see more.
  "to see more.", CompilerWarning)
('Time: ', 30.233999967575073)
Now try Numpy
C:\Python27\Scripts>
```

Kernel size=23: 30.23 seconds

### General purpose computing on GPU? And the role of OpenCL?

General purpose computing on a GPU is the execution of a programs using a GPU to accelerate computation speed. This is achieved by utilising the GPU over the CPU for some tasks that are computationally intensive and time consuming. This means both the CPU and the GPU run in parallel resulting in faster overall execution times of the program. OpenCL is the framework that allows the user to execute their program on the GPU.

Source: [What Is GPU Computing? \(boston.co.uk\)](http://www.boston.co.uk)

The median filter kernel has a size of 3.

### Median filter algorithms.

Title	Methodology	Time	UQI(50%)
SMF	Replace each pixel with a median of window.	$O(n^2)$	0.63
DBA	Uses a median of fixed-sized 3x3 window.	$O(n^2)$	0.72
NASMF	SMF and fuzzy weighted median filters.	$O(n^2)$	0.48
FRMF	Arakawa's fuzzy based median filter.	$O(n^2)$	0.23

Source: <https://www.sciencedirect.com/science/article/pii/S131915782030327X>

**UQI: Universal image quality index.**