Multi Core CPU Processing in VisionLab

10 April 2018

jaap@vdlmv.nl

Overview

- **Introduction**
- **Demonstration**
- **Automatic operator parallelization**
- **Calibrating for optimal performance**
- **Overview script commands**

Introduction

Observations:

- **The last years CPUs do not much become faster than about 4 GHz**
- **Multi-core PCs are now widely available at low costs**

Idea:

- **By using multi-core implementations of the vision algorithms the executing time of scripts become faster without modifying the scripts**

At the moment for more than 170 operators of VisionLab have a multi-core implementation

It is to be expected that in the near future more algorithms will be enabled for Multi Core Processing (MCP)

27-8-2018 Multi Core Processing in VisionLab 3

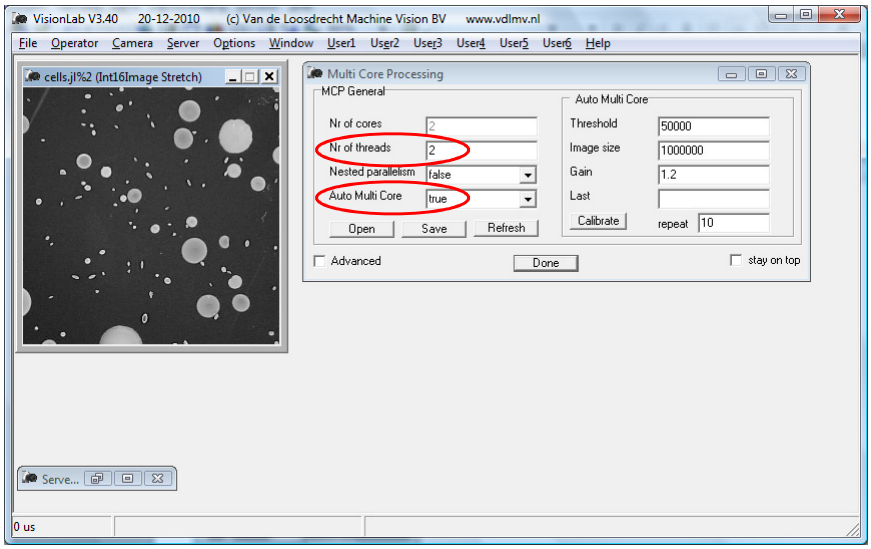
Demonstration Multi Core Processing (MCP)

- **Open image cells.jl**
- **Open Multi Core Processing form in Server main menu**
  Set Nr of threads to 2
  Set Auto Multi Core parameter to true
- **Execute a Convolution with Gaussian9x9 mask on image**
- **Set Nr of threats to 1 and execute Convolution again**
- **Crop image with ROI operator to 20 x 20 pixels**
  Set Nr of threads to 2
  Set Auto Multi Core parameter to false
  Execute Convolution on cropped image
- **Set Nr of threads to 1 and execute Convolution on cropped image**
- **Set Nr of threads to 2**
  Set Auto Multi Core parameter to true
  Execute Convolution on cropped image

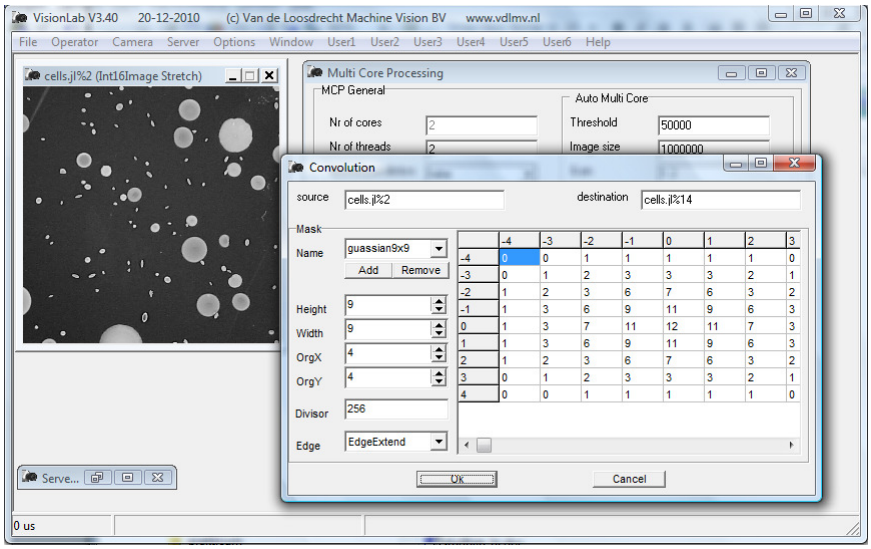
- **Note 1: this demonstration can only be run on a multi core computer**
- **Note 2: the time of executing op the operators can fluctuate due to other processes using the cores and caches.**

27-8-2018 Multi Core Processing in VisionLab 4

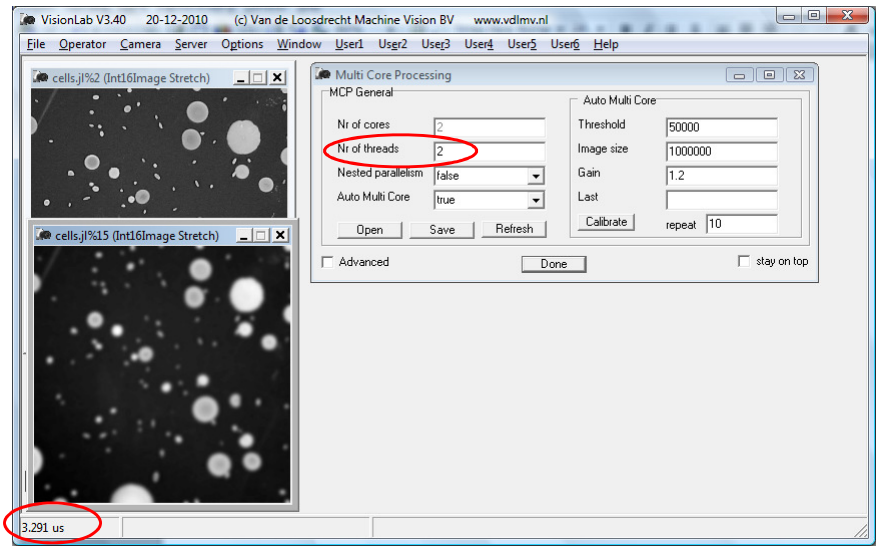
Set parameters Multi Core Processing



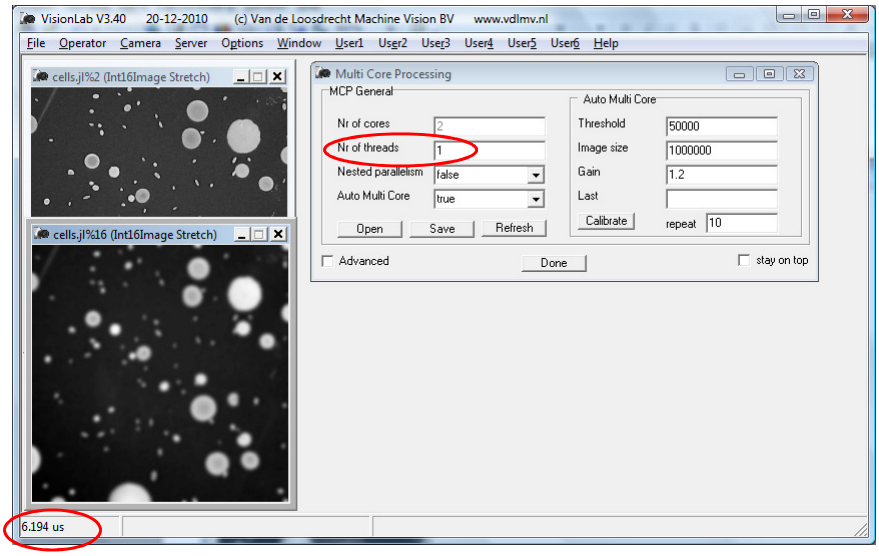
Select Convolution



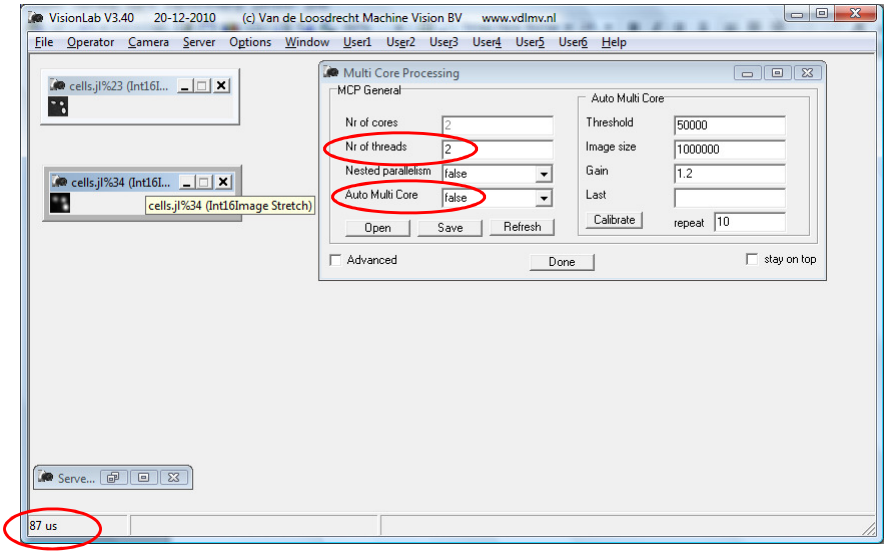
Convolution with Gaussian9x9 mask on 2 cores



Convolution with Gaussian9x9 mask on 1 core

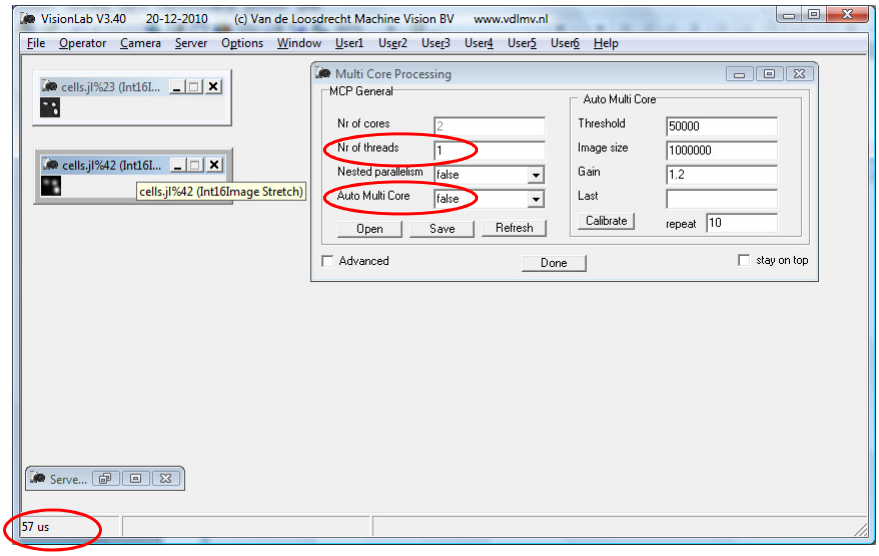


Nr of threats = 2, Auto Multi Core = false
Execute Convolution on cropped image



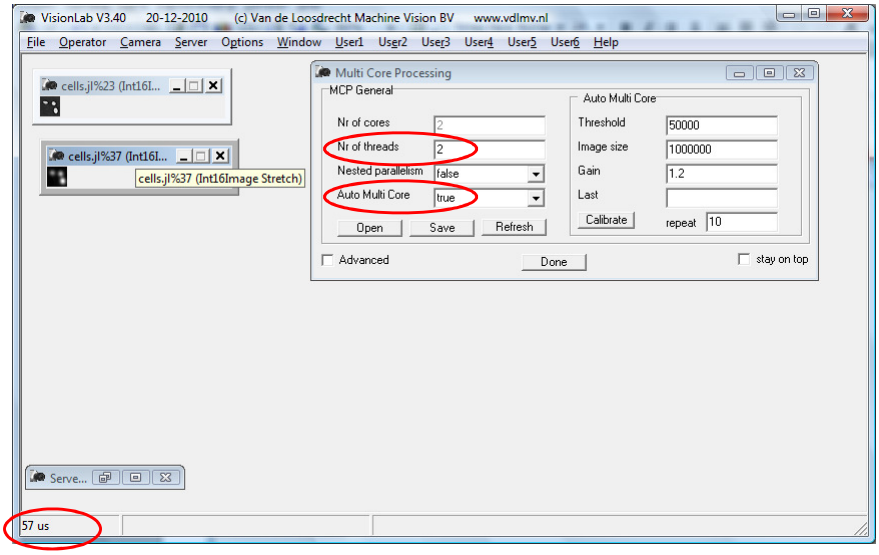
27-8-2018 Multi Core Processing in VisionLab 9

Nr of threats = 1, Auto Multi Core = false
Execute Convolution on cropped image



27-8-2018 Multi Core Processing in VisionLab 10

Nr of threats = 2, Auto Multi Core = true
Execute Convolution on cropped image



Processor affinity

The processor affinity can be controlled by starting the server manually. Example for Windows from the command prompt:

```
start /AFFINITY 0xF0 vissvr.exe 2066 NativeByteOrder
10 10000 EchoOff noautostartscript nodebug
```

Will start the server with an affinity with processors [4..7]
Note that the affinity mask is hexadecimal

Automatic operator parallelization for Multi Core Processing

On large images MCP can give a significant performance benefit

Due to the overhead involved with MCP the use of MCP on small images can lead to a performance lost compared to running on one core

If VisionLab MCP is run in the auto multi core mode VisionLab will decide automatically to run in MCP or single core. This is called automatic operator parallelization for MCP.

An important parameter in making this decision is the MultiCoreThreshold value. With the CalibrateAutoMultiCore VisionLab can calibrate the MultiCoreThreshold value.

Calibrating for optimal performance MCP

MCP performance will depend on hardware used, like processor type, memory speed, number of memory access channels, etc

For optimal performance VisionLab must be calibrated to the hardware
The settings will be saved in file: JL_MCP_Calib.txt
This file will be read at startup

Calibration process

- **Phase 1**
  basic calibration, takes only small amount of time
  performs a general calibration for all MCP enabled operators
  the MultiCoreThreshold value will be calculated
- **Phase 2**
  can optionally be performed AFTER phase 1
  performs for each MCP enabled operator a specific calibration

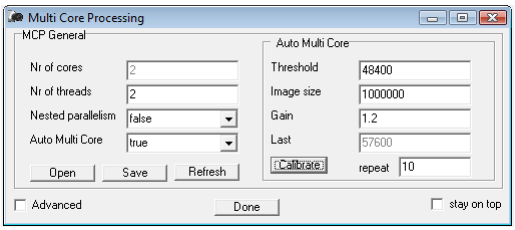
Demonstration MCP calibration phase 1

- **Open Multi Core Processing form in Server main menu**
- **Set Nr of threats to desired number of cores to use**
  Advice: use number of physical cores,
  On most system hyperthreading will not give extra performance
- **Set Image Size to: 1000000**
  Start size for MultiCoreThreshold value
- **Set Gain to 1.2**
  this how much profit MCP must give at least compared to running with one thread, example: 1.2 means 20% profit
- **Set repeat count to 10**
- **Click Calibrate button to start calibration process**
- **Save the result to file JL_MCP_Calib.txt**
  Store this file in the startup directory

- **NOTE for good calibration:**
  - **shut down all other applications**
  - **disconnect from network**
  - **do not run background tasks**

27-8-2018 Multi Core Processing in VisionLab 15

Demonstration MCP calibration phase 1



27-8-2018 Multi Core Processing in VisionLab 16

Demonstration MCP calibration phase 2

- **Open Multi Core Processing form in Server main menu**
- **First perform MCP calibration phase 1**
- **Check the Advanced option**
- **Click on the Zeros button to initialise previous settings**
- **Set repeat count to 10**
- **Click OptimzeAll to start calibration process**
- **Take a break, this will take some time**

- **Save the result to file JL_MCP_Calib.txt**
  Store this file in the startup directory

- **NOTE: The list of operators displayed are the "basic" MCP operators who can be calibrated. "Basic" MCP operators can be combined to more complex operators which can execute multi core.**
  So the number of operators that can run MCP is much greater then the number of basic" MCP operators.

Demonstration MCP calibration phase 2