

Benchmarking numpy / scikit-image / scipy vs clesperanto

```

In [1]: import clesperanto as cle
import numpy as np
import time
import matplotlib.pyplot as plt

num_iterations = 10

# measure execution time of a given method
def benchmark(function, kwargs):
    times = []
    for i in range(0, num_iterations):
        start_time = time.time()
        function(**kwargs)
        delta_time = time.time() - start_time
        times = times + [delta_time]
        # print(delta_time)

    # return median of measurements to ignore warmup-effects
    return np.median(times)

def benchmark_size(method_np, method_cle, method_cle_alloc):
    times_ref = []
    times_cle = []
    times_cle_alloc = []
    sizes = []
    for size in [1, 2, 4, 8, 16, 32, 64]:

        input1 = np.zeros((1024, 1024, size))
        cl_input1 = cle.push(input1)
        cl_input2 = cle.create(cl_input1.shape)

        time_ref = benchmark(method_np, {"image":input1})
        time_cle = benchmark(method_cle, {"image":cl_input1, "output":cl_input
2})
        time_cle_alloc = benchmark(method_cle_alloc, {"image":cl_input1})

        times_ref = times_ref + [time_ref]
        times_cle = times_cle + [time_cle]
        times_cle_alloc = times_cle_alloc + [time_cle_alloc]
        sizes = sizes + [size]

    plt.plot(sizes, times_ref, 'r--', sizes, times_cle, 'g--', sizes, times_c
le_alloc, 'b--');
    plt.ylabel('Time / ms')
    plt.xlabel('Image size / MB')
    plt.legend(("ref", "cle", "cle+alloc"));
    plt.show()

    print("\nSizes (MB)          " + str(sizes))
    print("Times ref (s)          " + str(np.round(times_ref, 4)))
    print("Times cle (s)            " + str(np.round(times_cle, 4)))
    print("Times cle+alloc (s)       " + str(np.round(times_cle_alloc, 4)))

```

Thresholding

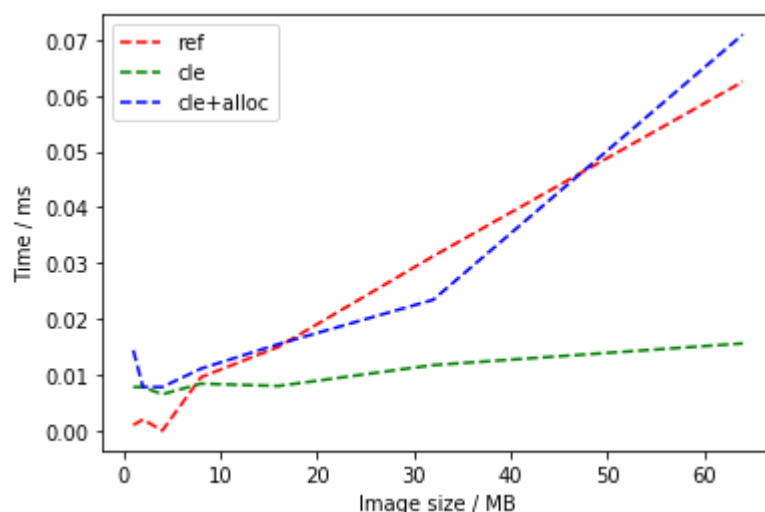
```
In [2]: # RED: thresholding of a numpy array
def threshold_ref(image):
    thresholded = image > 100
    return thresholded

# GREEN: thresholding of a pre-existing opencl array (no push, pull or alloc)
def threshold_cle(image, output):
    cle.greater_constant(image, output, 100)

# BLUE: allocate result memory + thresholding
def threshold_cle_alloc(image):
    thresholded = cle.create(image.shape)
    cle.greater_constant(image, thresholded, 100)

benchmark_size(threshold_ref, threshold_cle, threshold_cle_alloc)
```

C:\Users\rober\Miniconda3\envs\cle\lib\site-packages\pyopencl__init__.py:24
 8: CompilerWarning: Non-empty compiler output encountered. Set the environmen
 t variable PYOPENCL_COMPILER_OUTPUT=1 to see more.
 warn("Non-empty compiler output encountered. Set the "



Sizes (MB)	[1, 2, 4, 8, 16, 32, 64]
Times ref (s)	[0.001 0.002 0. 0.0096 0.015 0.0313 0.0625]
Times cle (s)	[0.0078 0.0078 0.0065 0.0084 0.008 0.0117 0.0156]
Times cle+alloc (s)	[0.0144 0.0078 0.0078 0.0111 0.0155 0.0234 0.071]

Gaussian blur radius 2

```
In [3]: from skimage.filters import gaussian

radius = 2

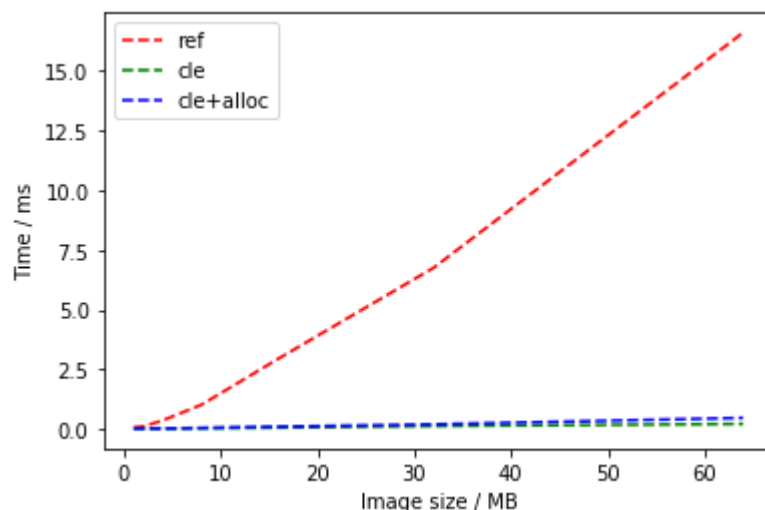
def gaussian_blur_filter_ref(image):
    filtered = gaussian(image, sigma=radius)
    return filtered

def gaussian_blur_filter_cle(image, output):
    cle.gaussian_blur(image, output, radius, radius, radius)

def gaussian_blur_filter_cle_alloc(image):
    filtered = cle.create(image.shape)
    cle.gaussian_blur(image, filtered, radius, radius, radius)

benchmark_size(gaussian_blur_filter_ref, gaussian_blur_filter_cle, gaussian_blur_filter_cle_alloc)
```

C:\Users\rober\Miniconda3\envs\cle\lib\site-packages\pyopencl__init__.py:248: CompilerWarning: Non-empty compiler output encountered. Set the environment variable PYOPENCL_COMPILER_OUTPUT=1 to see more.
warn("Non-empty compiler output encountered. Set the "

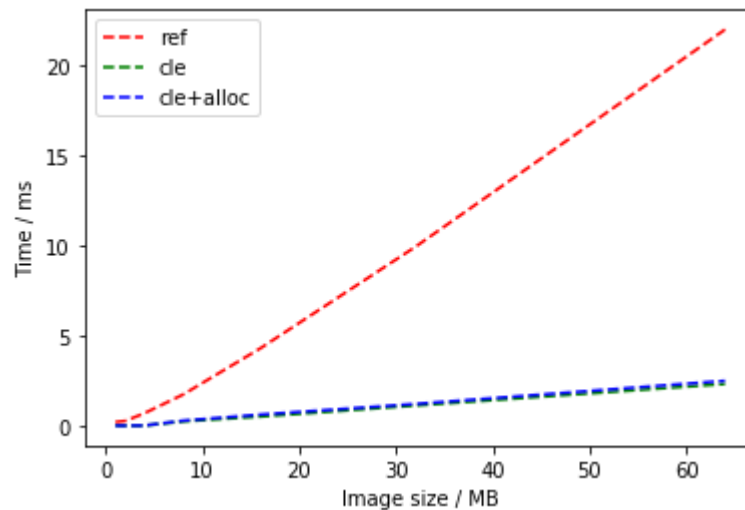


Sizes (MB)	[1, 2, 4, 8, 16, 32, 64]
Times ref (s)	[0.0727 0.1102 0.3626 1.018 2.9589 6.7538 16.6034]
Times cle (s)	[0.0302 0.0234 0.0234 0.0313 0.0612 0.127 0.2153]
Times cle+alloc (s)	[0.0184 0.0234 0.0234 0.0498 0.1016 0.1978 0.4688]

Gaussian blur radius 10

```
In [4]: radius = 10
benchmark_size(gaussian_blur_filter_ref, gaussian_blur_filter_cle, gaussian_blur_filter_cle_alloc)
```

C:\Users\rober\Miniconda3\envs\cle\lib\site-packages\pyopencl__init__.py:248: CompilerWarning: Non-empty compiler output encountered. Set the environment variable PYOPENCL_COMPILER_OUTPUT=1 to see more.
 warn("Non-empty compiler output encountered. Set the "



Sizes (MB)	[1, 2, 4, 8, 16, 32, 64]
Times ref (s)	[0.2203 0.2774 0.7055 1.7437 4.3068 9.9177 21.9466]
Times cle (s)	[0.0313 0.0218 0.0221 0.2537 0.5198 1.1307 2.3259]
Times cle+alloc (s)	[0.0224 0.0189 0.0259 0.2931 0.6308 1.2177 2.4975]

Binary erosion

```
In [5]: from skimage.morphology import binary_erosion

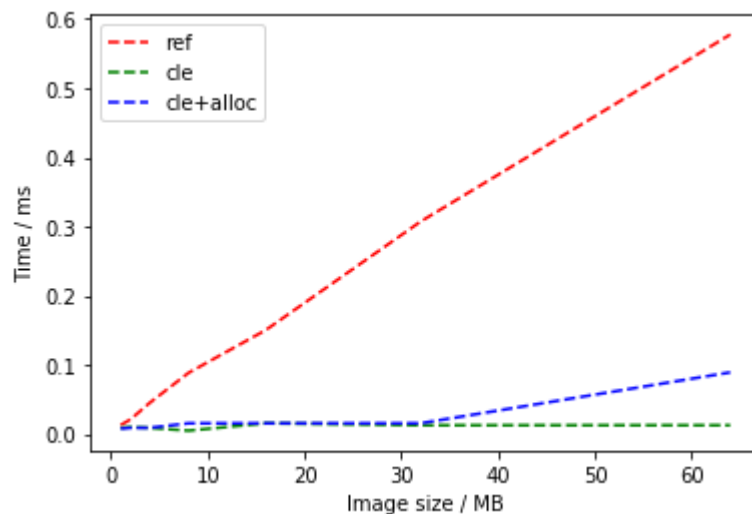
def binary_erosion_ref(image):
    filtered = binary_erosion(image)
    return filtered

def binary_erosion_cle(image, output):
    cle.erode_box(image, output)

def binary_erosion_cle_alloc(image):
    filtered = cle.create(image.shape)
    cle.erode_box(image, filtered)

benchmark_size(binary_erosion_ref, binary_erosion_cle, binary_erosion_cle_alloc)
```

C:\Users\rober\Miniconda3\envs\cle\lib\site-packages\pyopencl__init__.py:248: CompilerWarning: Non-empty compiler output encountered. Set the environment variable PYOPENCL_COMPILER_OUTPUT=1 to see more.
 warn("Non-empty compiler output encountered. Set the "



Sizes (MB)	[1, 2, 4, 8, 16, 32, 64]
Times ref (s)	[0.013 0.0212 0.0452 0.0885 0.1504 0.3073 0.5775]
Times cle (s)	[0.009 0.01 0.009 0.005 0.0156 0.0128 0.0128]
Times cle+alloc (s)	[0.008 0.009 0.009 0.0156 0.0156 0.0156 0.0892]

Mean filter radius=2

```
In [6]: import scipy.ndimage.filters as spf

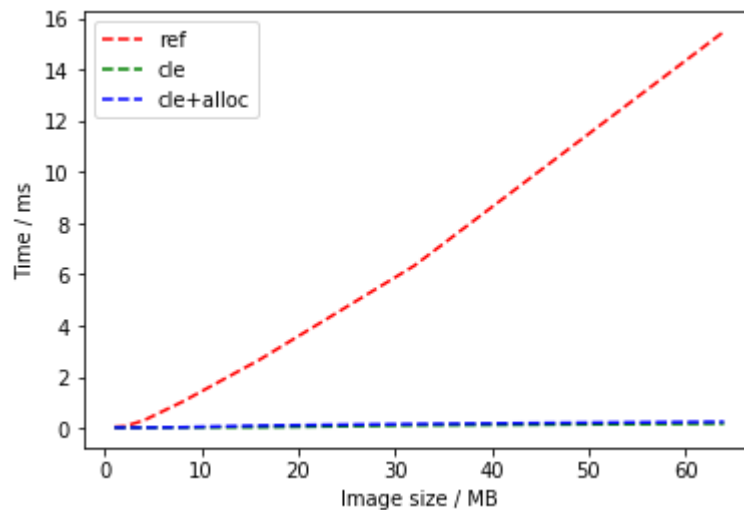
radius = 2
def mean_filter_ref(image):
    # todo: not sure if size is a radius or a diameter. Check documentation
    # https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.unifo
    rm_filter.html#scipy.ndimage.uniform_filter
    filtered = spf.uniform_filter(image, size=radius)
    return filtered

def mean_filter_cle(image, output):
    cle.mean_box(image, output, radius, radius, radius)

def mean_filter_cle_alloc(image):
    filtered = cle.create(image.shape)
    cle.mean_box(image, filtered, radius, radius, radius)

benchmark_size(mean_filter_ref, mean_filter_cle, mean_filter_cle_alloc)
```

C:\Users\rober\Miniconda3\envs\cle\lib\site-packages\pyopencl__init__.py:24
 8: CompilerWarning: Non-empty compiler output encountered. Set the environmen
 t variable PYOPENCL_COMPILER_OUTPUT=1 to see more.
 warn("Non-empty compiler output encountered. Set the "



Sizes (MB)	[1, 2, 4, 8, 16, 32, 64]
Times ref (s)	[0.0433 0.0669 0.2949 1.0248 2.6696 6.3408 15.4764]
Times cle (s)	[0.0246 0.0189 0.0189 0.0267 0.0313 0.108 0.1739]
Times cle+alloc (s)	[0.0241 0.0181 0.0214 0.0326 0.0984 0.1614 0.2473]

In []: