PC Big Assignment

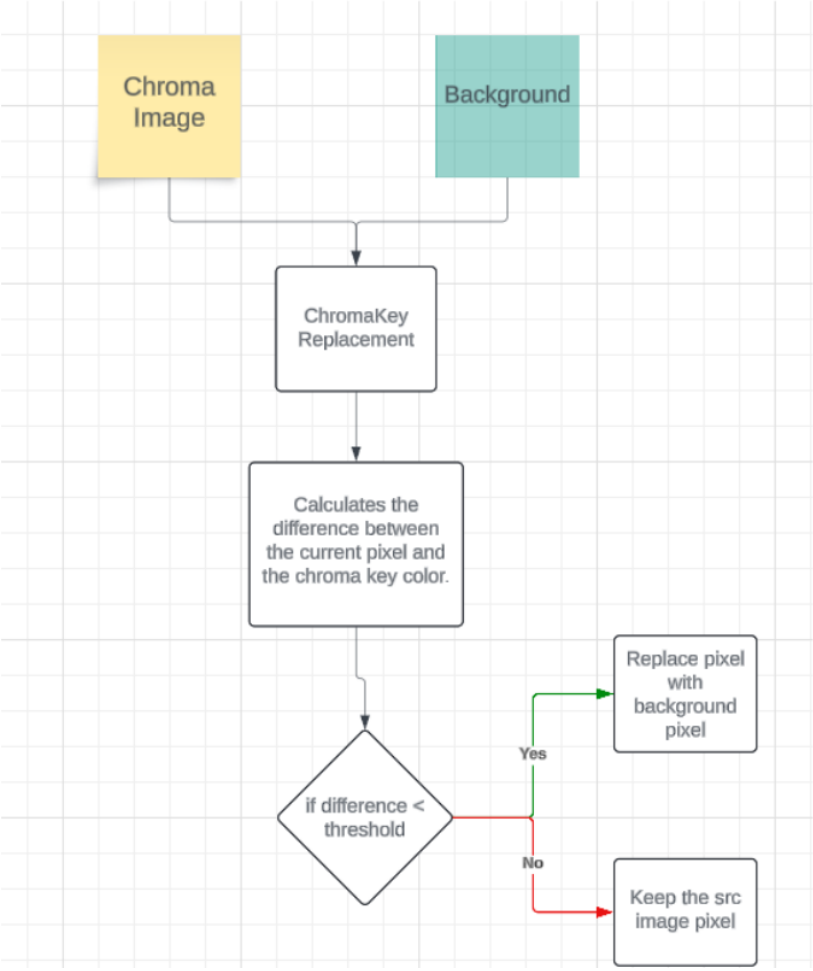
Name	BN	Section
Ahmed Mohmed Ahmed Lotfy	8	1

Introduction

- **Chroma Key Replacement**: is a visual effects technique.
- Used to composite two images. This technique is widely used in video production, filmmaking, and image processing.
- Replace a solid background color (usually green or blue) with a different background, enabling the creation of immersive visual effects and scenes.

How does it work?

Flow



Tests

480x240


+


=


Tests

640x480


+


=


Tests

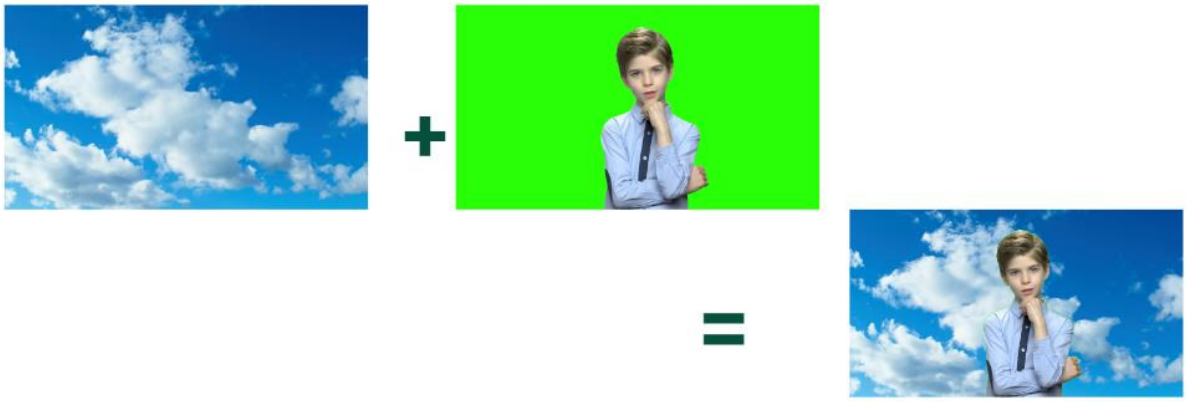
700x900


+


=


Tests

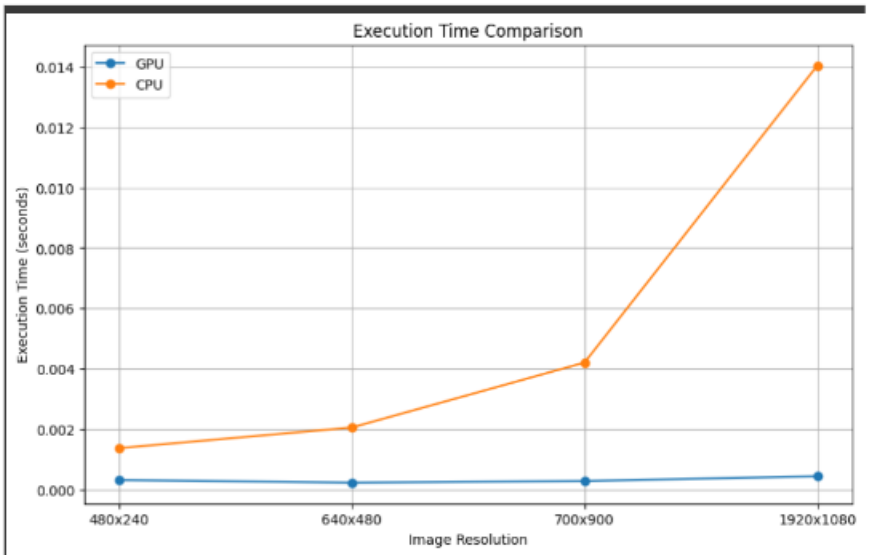
1920x1080



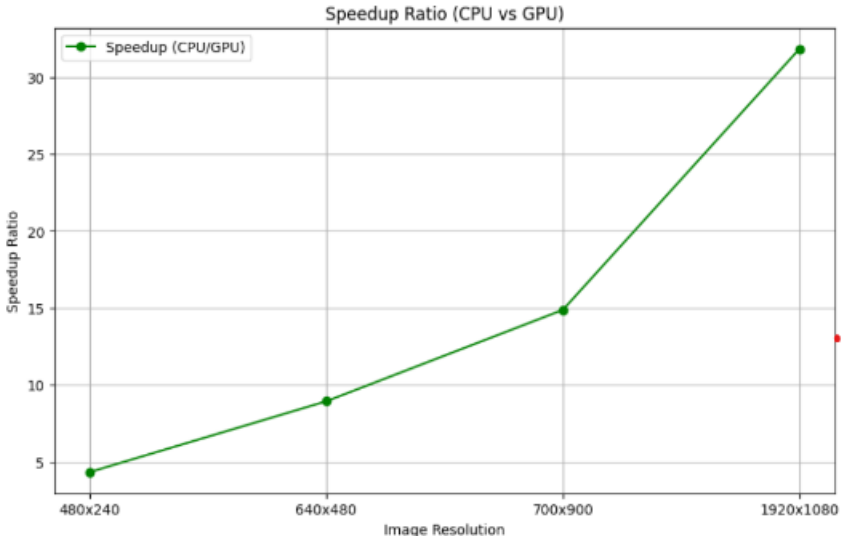
Performance Analysis

Image Resolution	GPU Time	CPU Time	Speedup= CPU/GPU
480x240	0.00031553 seconds	0.00137312 seconds	4.35
640x480	0.00022975 seconds	0.00205611 seconds	8.94
700x900	0.00028273 seconds	0.00420714 seconds	14.88
1920x1080	0.00044120 seconds	0.0140391 seconds	31.8

GPU Code vs CPU Code



GPU Code vs CPU Code



GPU Profiling

480x240

```
==1074== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   58.86%  61.344us         2  30.672us  30.592us  30.752us  [CUDA memcpy HtoD]
                   29.51%  30.752us         1  30.752us  30.752us  30.752us  [CUDA memcpy DtoH]
                   11.64%  12.128us         1  12.128us  12.128us  12.128us  replaceChromaBackground(uchar3*, uchar3*, uchar3*
      API calls:   99.49%  224.85ms         3  74.950ms  3.4910us  224.84ms  cudaMalloc
                    0.16%  357.49us         3  119.16us  99.595us  139.79us  cudaMemcpy
                    0.13%  297.30us         1  297.30us  297.30us  297.30us  cudaLaunchKernel
                    0.11%  250.94us         3  83.645us  5.0590us  203.50us  cudaFree
                    0.09%  202.09us       114  1.7720us     183ns  78.079us  cuDeviceGetAttribute
                    0.01%  14.505us         1  14.505us  14.505us  14.505us  cuDeviceGetName
                    0.01%  13.260us         1  13.260us  13.260us  13.260us  cudaDeviceSynchronize
                    0.00%  7.4750us         1  7.4750us  7.4750us  7.4750us  cuDeviceGetPCIBusId
                    0.00%  6.6750us         1  6.6750us  6.6750us  6.6750us  cuDeviceTotalMem
                    0.00%  2.3170us         3     772ns     331ns  1.6310us  cuDeviceGetCount
                    0.00%  1.2080us         2     604ns     320ns     888ns  cuDeviceGet
                    0.00%     523ns         1     523ns     523ns     523ns  cuModuleGetLoadingMode
                    0.00%     416ns         1     416ns     416ns     416ns  cuDeviceGetUuid
```

640x480

```
==3737== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   58.93%  154.14us         2  77.071us  77.056us  77.087us  [CUDA memcpy HtoD]
                   30.35%  79.391us         1  79.391us  79.391us  79.391us  [CUDA memcpy DtoH]
                   10.72%  28.032us         1  28.032us  28.032us  28.032us  replaceChromaBackground(uchar3*, uchar3*, uchar3*
      API calls:   99.17%  192.76ms         3  64.252ms  3.4160us  192.67ms  cudaMalloc
                    0.41%  804.93us         3  268.31us  210.73us  314.57us  cudaMemcpy
                    0.22%  421.35us         3  140.45us  51.044us  221.63us  cudaFree
                    0.10%  197.65us         1  197.65us  197.65us  197.65us  cudaLaunchKernel
                    0.07%  140.56us       114  1.2320us     140ns  55.878us  cuDeviceGetAttribute
                    0.01%  27.929us         1  27.929us  27.929us  27.929us  cudaDeviceSynchronize
                    0.01%  11.789us         1  11.789us  11.789us  11.789us  cuDeviceGetName
                    0.00%  5.3080us         1  5.3080us  5.3080us  5.3080us  cuDeviceGetPCIBusId
                    0.00%  4.6690us         1  4.6690us  4.6690us  4.6690us  cuDeviceTotalMem
                    0.00%  1.3940us         3     464ns     238ns     912ns  cuDeviceGetCount
                    0.00%     919ns         2     459ns     176ns     743ns  cuDeviceGet
                    0.00%     547ns         1     547ns     547ns     547ns  cuModuleGetLoadingMode
                    0.00%     215ns         1     215ns     215ns     215ns  cuDeviceGetUuid
```

700x900

```
==9166== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   69.52%  515.10us         2  257.55us  254.75us  260.35us  [CUDA memcpy HtoD]
                   24.28%  179.94us         1  179.94us  179.94us  179.94us  [CUDA memcpy DtoH]
                    6.20%  45.951us         1  45.951us  45.951us  45.951us  replaceChromaBackground(uchar3*, uchar3*, ucha
      API calls:   98.68%  183.08ms         3  61.027ms  74.562us  182.93ms  cudaMalloc
                    0.80%  1.4879ms         3  495.96us  461.86us  555.00us  cudaMemcpy
                    0.28%  516.38us         3  172.13us  144.78us  224.76us  cudaFree
                    0.12%  229.99us         1  229.99us  229.99us  229.99us  cudaLaunchKernel
                    0.08%  144.31us       114  1.2650us     136ns  61.923us  cuDeviceGetAttribute
                    0.03%  48.483us         1  48.483us  48.483us  48.483us  cudaDeviceSynchronize
                    0.01%  11.181us         1  11.181us  11.181us  11.181us  cuDeviceGetName
                    0.00%  7.7870us         1  7.7870us  7.7870us  7.7870us  cuDeviceGetPCIBusId
                    0.00%  4.2420us         1  4.2420us  4.2420us  4.2420us  cuDeviceTotalMem
                    0.00%  2.5060us         2  1.2530us     175ns  2.3310us  cuDeviceGet
                    0.00%  1.5970us         3     532ns     205ns  1.1190us  cuDeviceGetCount
                    0.00%     627ns         1     627ns     627ns     627ns  cuModuleGetLoadingMode
                    0.00%     224ns         1     224ns     224ns     224ns  cuDeviceGetUuid
```

1920x1080

```
==10896== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   66.79%  2.9128ms         2  1.4564ms  1.4423ms  1.4705ms  [CUDA memcpy HtoD]
                   29.14%  1.2710ms         1  1.2710ms  1.2710ms  1.2710ms  [CUDA memcpy DtoH]
                    4.07%  177.28us         1  177.28us  177.28us  177.28us  replaceChromaBackground(uchar3*, uchar3*, ucha
      API calls:   97.02%  216.28ms         3  72.095ms  110.75us  216.06ms  cudaMalloc
                    2.29%  5.1145ms         3  1.7048ms  1.6910ms  1.7148ms  cudaMemcpy
                    0.38%  848.68us         3  282.89us  235.59us  307.76us  cudaFree
                    0.11%  248.98us         1  248.98us  248.98us  248.98us  cudaLaunchKernel
                    0.09%  208.61us       114  1.8290us     281ns  79.256us  cuDeviceGetAttribute
                    0.08%  180.18us         1  180.18us  180.18us  180.18us  cudaDeviceSynchronize
                    0.01%  14.298us         1  14.298us  14.298us  14.298us  cuDeviceGetName
                    0.00%  8.2830us         1  8.2830us  8.2830us  8.2830us  cuDeviceGetPCIBusId
                    0.00%  6.1030us         1  6.1030us  6.1030us  6.1030us  cuDeviceTotalMem
                    0.00%  2.4570us         3     819ns     378ns  1.6740us  cuDeviceGetCount
                    0.00%  1.2830us         2     641ns     362ns     921ns  cuDeviceGet
                    0.00%     711ns         1     711ns     711ns     711ns  cuModuleGetLoadingMode
                    0.00%     392ns         1     392ns     392ns     392ns  cuDeviceGetUuid
```

1920x1080 (with streams)

```
GPU Time with Streams: 0.00633823 seconds
==1366== Profiling application: ./kern10
==1366== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   60.47%  4.0462ms         1  4.0462ms  4.0462ms  4.0462ms  [CUDA memcpy DtoH]
                   37.58%  2.5148ms         2  1.2574ms  1.2531ms  1.2617ms  [CUDA memcpy HtoD]
                    1.95%  130.66us         1  130.66us  130.66us  130.66us  chromaKeyKernel(unsigned char*, unsigned char*, u
      API calls:   95.65%  230.46ms         3  76.822ms  124.85us  230.21ms  cudaMalloc
                    3.71%  8.9470ms         3  2.9823ms  1.4111ms  6.0886ms  cudaMemcpyAsync
                    0.37%  890.62us         3  296.87us  248.51us  333.41us  cudaFree
                    0.11%  268.88us       114  2.3580us     231ns  128.50us  cuDeviceGetAttribute
                    0.09%  212.01us         1  212.01us  212.01us  212.01us  cudaLaunchKernel
                    0.02%  51.970us         2  25.985us  3.6310us  48.339us  cudaStreamCreate
                    0.01%  35.811us         2  17.905us  17.722us  18.089us  cudaStreamDestroy
                    0.01%  29.293us         2  14.646us  2.7430us  26.550us  cudaStreamSynchronize
                    0.01%  15.027us         1  15.027us  15.027us  15.027us  cuDeviceGetName
                    0.00%  8.4320us         1  8.4320us  8.4320us  8.4320us  cuDeviceGetPCIBusId
                    0.00%  6.1540us         1  6.1540us  6.1540us  6.1540us  cuDeviceTotalMem
                    0.00%  2.4230us         3     807ns     390ns  1.6390us  cuDeviceGetCount
                    0.00%  1.2360us         2     618ns     270ns     966ns  cuDeviceGet
                    0.00%     532ns         1     532ns     532ns     532ns  cuModuleGetLoadingMode
                    0.00%     464ns         1     464ns     464ns     464ns  cuDeviceGetUuid
```

Theoretical Complexity

GPU: O(1) as each thread operates over one pixel

CPU: O(n) as it loops over rows and columns so it's the number of pixels in the image

Theoretical Speedup = CPU/GPU = O(n)/O(1)

For 640x480 speedup=307200
For 700x900 speedup= 630000
For 1920x1080 speedup= 2073600

Compare between Theoretical and actual speedup

For 640x480 speedup= 34362
For 700x900 speedup= 42338
For 1920x1080 speedup= 65207

Why the speedup is less than the theoretical one?

- **Memory Transfer Overhead**: Data transfer between CPU and GPU adds latency (use streams).
- **Parallel Efficiency**: Not all code sections are efficiently parallelized.
- **Resource Contention**: Multiple threads may contend for resources.

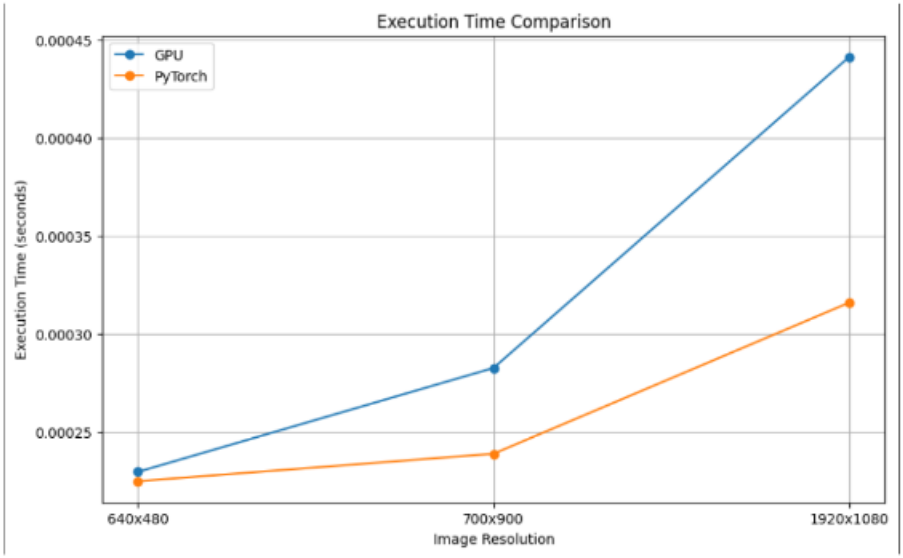
How to achieve better speed up?

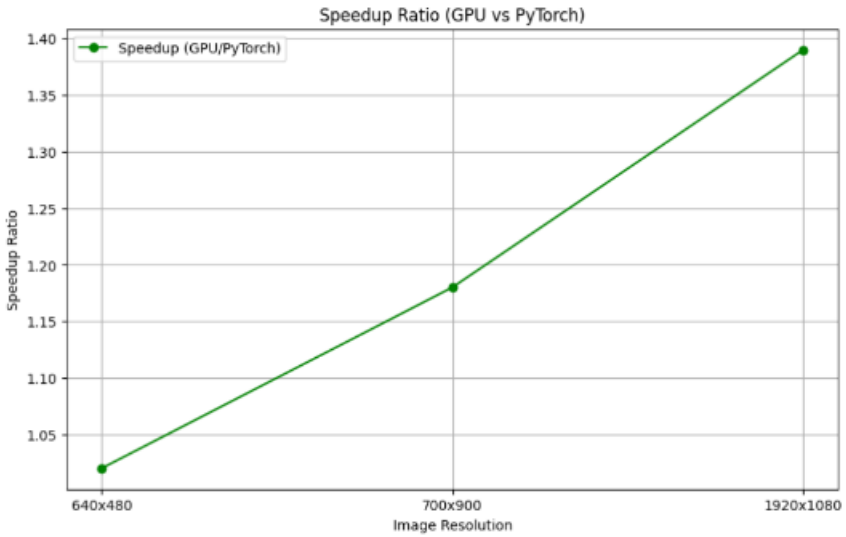
- **Coalesced Memory Access:** meaning that threads in a warp access contiguous memory locations
- **Use Shared Memory**
- **Fine-Tune Kernel Configuration:** Adjust the block and grid sizes for better performance based on the specific GPU architecture.

Benchmarking

Our GPU Code vs Pytorch with CUDA

Image Resolution	GPU	Pytorch	Speedup= GPU/Pytorch
640x480	0.00022975 seconds	**0.000225 seconds**	1.02
700x900	0.00028273 seconds	**0.000239 seconds**	1.18
1920x1080	0.00044120 seconds	**0.000316 seconds**	1.39




Speedup Ratio (GPU vs PyTorch)
Speedup (GPU/PyTorch)
Speedup Ratio
1.40
1.35
1.30
1.25
1.20
1.15
1.10
1.05
640x480
700x900
1920x1080
Image Resolution

Thank You