



# Accelerating Radiance Using OpenCL on GPU

Wangda Zuo, Ph.D.

Simulation Research Group  
Energy and Environmental Technologies Division  
Lawrence Berkeley National Laboratory

EETD Seminar, 06/22/2011



# Acknowledgement

## **Project Members:**

**Andrew McNeil, Michael Wetter, Eleanor Lee**

## **Funding Agency:**

**California Energy Commission**

# Objective

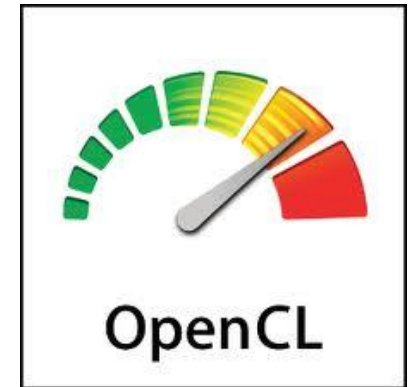
**To accelerate annual daylighting simulations for fenestration systems in Radiance ray-tracing program.**

# Why OpenCL?

**Parallel programming language**

**Cross-platforms (personal computers, servers, handheld/embedded devices)**

**Open and royalty free**



# Step 1: Optimizations of Algorithms

Annual daylighting simulation is done by Radiance program  
`dctimestep.c`

**Original:**

*For  $i=1$  to 8760*

*Call `dctimestep.c` :  $V_I(t_i) = M_V M_T M_D V_S(t_i)$*

**New:**

*`Dctimestep_new.c`:  $M_{VI} = M_V M_T M_D M_{VS}$ ,*

*$M_{VS} = [V_S(t_1), \dots, V_S(t_{8760})]$ ,  $M_{VI} = [V_I(t_1), \dots, V_I(t_{8760})]$ .*

$V_I(t)$  is the illuminance vector for a specific time step,  $t_i$

$V_S(t)$  is the sky vector defining sky patch radiance at  $t_i$

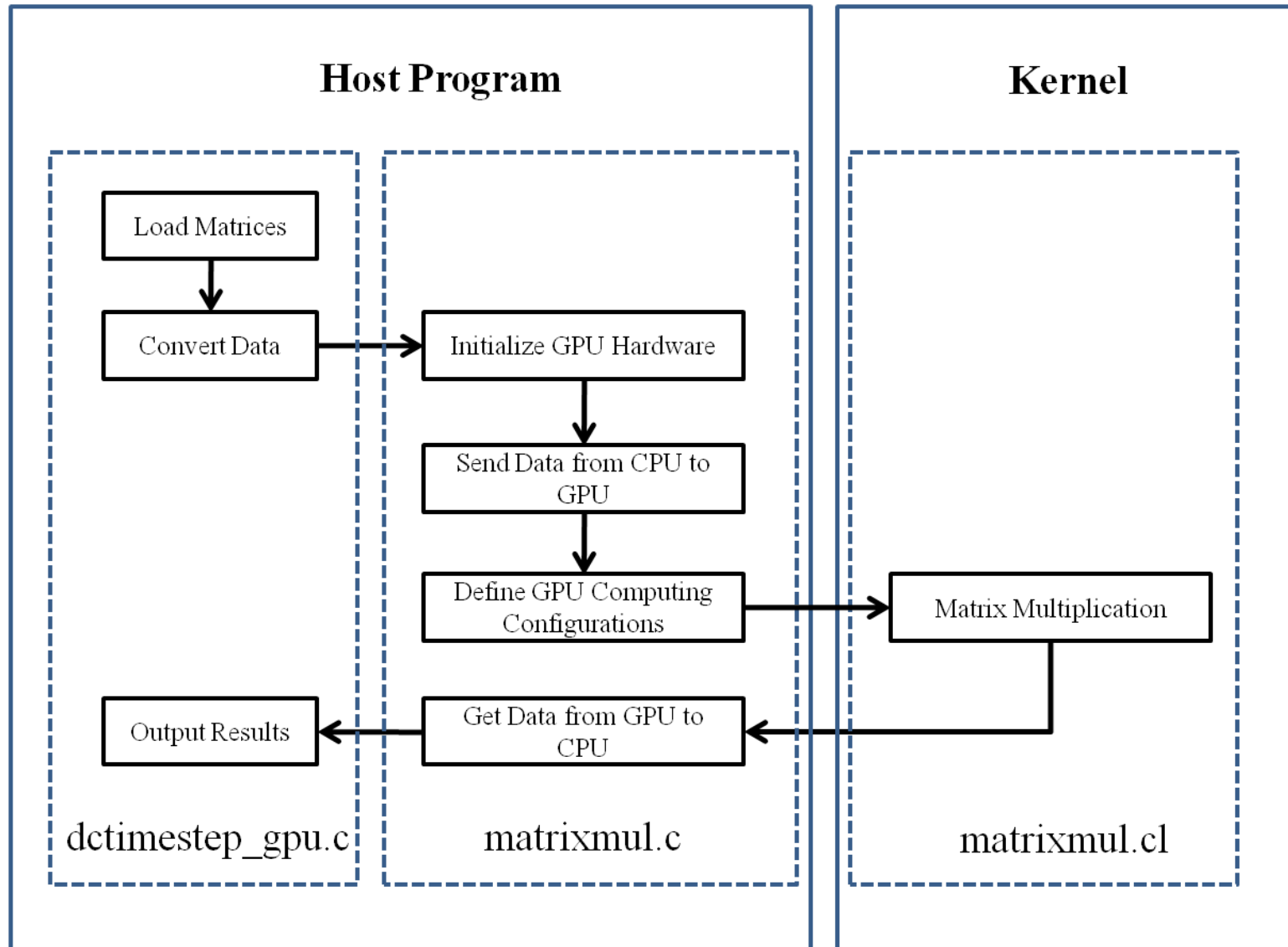
$M_V$  is a view matrix defining lighting connection from the exiting directions of the windows to the sensors.

$M_T$  is a matrix converted from bidirectional transmittance distribution function (BTDF), which describes transmission of the light passes through the surface of studied windows.

$M_D$  is a daylighting matrix defining coefficients between incoming directions for the windows and sky patches.

# Step 2: Acceleration by GPU

NVIDIA GeForce GTX 460, 336 Cores, 1335 MHz, 1GB DDR5 Memory , ~\$200

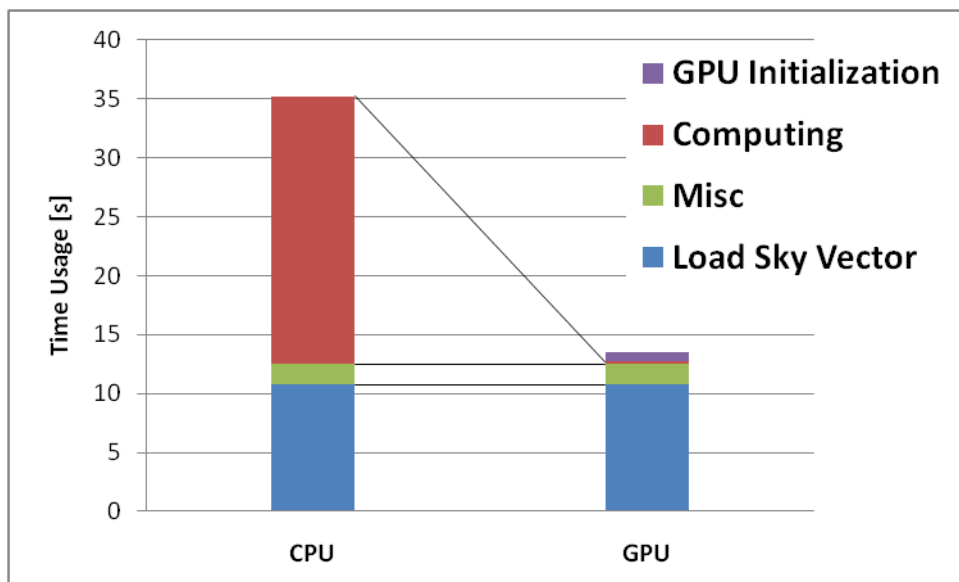


# Results

## Speedup of Annual Daylighting Simulation

Sky Vectors	$V_S = 146 \times 3$	$V_S = 2306 \times 3$
Algorithm Optimization	86.9 ×	11.0 ×
Switching to GPU	1.3 ×	2.6 ×
Total	<b>101.7 ×</b>	<b>28.6 ×</b>

## Usage of Time by Optimized Codes ( $V_S = 2306 \times 3$ )



### Loading Sky Vectors:

- 404 MB file
- 10.76s = 79% of total time for GPU code

### Computing Time:

- CPU: 22.74s vs GPU: 0.27s
- **85.7 × Reduction**

# Expectations for NERSC

**Provide CPU/GPU clusters to run parametric study of fenestration systems for annual daylighting simulations**

**Provide remote access to computers with different configurations in software (OS, compilers) & hardware (CPUs, GPUs, CPUs/GPUs) to test compatibility & measure performance**



**Thank You!**