Better, Faster, Stronger
Super-Fast Glare Analysis and Real-Time Visualization

Nathaniel Jones
August 22, 2019

Massachusetts Institute of Technology
Sustainable Design Lab

Arup San Francisco
Advanced Technology & Research
Problem

How can simulation be an effective design tool?
Problem

How can simulation be an effective design tool?

Create a **stronger** connection between tool and user

Provide **faster** results for annual and spatial glare analysis

Make **better** decisions using real-time glare analysis
Accurate
49 minutes

Fast
1.5 minutes
Unified Theories of Cognition
Allen Newell

**Deliberate Act**
Mouse, Trackpad, Keyboard
< 0.1 s

**Cognitive Operation**
Pointing, Commands, Requests
0.1 – 1.0 s

**Unit Task**
Modeling, Writing, Games
> 1.0 s
Unit Task
- Modeling, Writing, Games

Cognitive Operation
- Pointing, Commands, Requests

Deliberate Act
- Mouse, Trackpad, Keyboard

Time (seconds)
Data up to the year 2010 collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
Data for 2010-2015 collected by K. Rupp
Data up to the year 2010 collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
Data for 2010-2015 collected by K. Rupp
Data up to the year 2010 collected by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
Data for 2010-2015 collected by K. Rupp
CPU

4 cores
Intel Kaby Lake

GPU

4608 cores + 72 RT cores
Nvidia Turing
Accelerad

rpict  100x faster
rtrace  28x faster
rcontrib  45x faster
Problem

How can simulation be an effective design tool?

Create a **stronger** connection between tool and user

Provide **faster** results for annual and spatial glare analysis

Make **better** decisions using real-time glare analysis
Task
Evaluate glare at every point in a room in every viewing direction at every hour of the year for multiple façade designs
Spatial Daylight Autonomy (sDA)

\[ sDA_{300lux,50\%} \]

Fraction of space in which daylighting achieves 300 Lux in at least 50% of occupied hours

Daylight Autonomy (DA)

\[ DA_{300lux} \]

Fraction of occupied time in which daylighting achieves 300 Lux

Daylight Autonomy (DA)

\[ DA_{300\text{lux}} \]
Fraction of occupied time in which daylighting achieves 300 Lux

Spatial Daylight Autonomy (sDA)

\[ sDA_{300\text{lux},50\%} \]
Fraction of space in which daylighting achieves 300 Lux in at least 50% of occupied hours

Glare Autonomy (GA)

\[ GA_{40\%,5\%} \]
Fraction of occupied time in which daylight glare probability is less than 40%

Spatial Glare Autonomy (sGA)

Fraction of space in which daylight glare probability exceeds 40% for no more than 5% of occupied hours
Daylight Glare Probability (DGP)

\[ DGP = 5.87 \times 10^{-5}E_v + 0.0918 \times \log_{10} \left( 1 + \sum_{i=1}^{n} \frac{I_{s,i}^2 \omega_{s,i}}{E_v^{1.87} P_i^2} \right) + 0.16 \]

- Brightness
- Contrast
- Guth position index
819 locations
×
8 directions
×
2080 hours
=
14 million images
Solution

Use matrix-based methods
The diagram illustrates the 2-Phase Method involving the following matrices and operations:

1. **Daylight Coefficient Matrix**: Represents $n$ sensor points and $p$ sky patches.
2. **Sky Matrix**: Depicts $p$ sky patches over $t$ times.
3. **Irradiance Matrix**: Indicates $n$ sensor points over $t$ times.

The formula is represented as:

$$\text{Daylight Coefficient Matrix} \times \text{Sky Matrix} = \text{Irradiance Matrix}$$

Where:
- $n$ sensor points
- $p$ sky patches
- $t$ times
- $r_{\text{contrib}}$
- $\text{gendaymtx}$
- $\text{dctimestep}$
1. Model
2. Render
3. Find Glare Sources

1. Model
2. Find Glare Sources in Scene
Daylight Glare Probability (DGP)

\[ DGP = 5.87 \times 10^{-5} E_v + 0.0918 \times \log_{10} \left( 1 + \sum_{i=1}^{n} \frac{I_{s,i}^2 \omega_{s,i}}{E_v^{1.87} P_i^2} \right) + 0.16 \]
Daylight Glare Probability (DGP)

\[ DGP = 5.87 \times 10^{-5} E_v + 0.0918 \times \log_{10} \left( 1 + \sum_{i=1}^{n} \frac{L_{s,i}^2 \omega_{s,i} \cos \theta}{E_v^{1/3} P_i} \right) + 0.16 \]

\[ E_v = 179 \times D_{total} S \]

\[ L_s = 179 \times \frac{d_{direct} S_i}{\omega \cos \theta} \]

if patch \( i \) in field of view and \( L_s > \) threshold

# of sky patches

Brightness

Contrast

## Spatial Glare Autonomy (sGA)

![Image of sGA visualization]

**sGA: 73%**

<table>
<thead>
<tr>
<th>Method</th>
<th>Calculation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual Renderings</td>
<td>1600 years</td>
</tr>
<tr>
<td>2-Phase Rendering</td>
<td>6 years</td>
</tr>
<tr>
<td>Batch Rendering</td>
<td>600 days</td>
</tr>
<tr>
<td>Batch Rendering with eDGPs</td>
<td>164 days</td>
</tr>
<tr>
<td>Imageless DGP</td>
<td>25 minutes</td>
</tr>
<tr>
<td>Imageless DGP on GPU</td>
<td>2 minutes</td>
</tr>
</tbody>
</table>
Accelerad

rpict  100x faster
rtrace  28x faster
rcontrib  45x faster
dcglare  133,000x faster
RTX Speedup

Radiance (GTX 1080 Ti) 24x
Accelerad 0.6 (GTX 1080 Ti) 3.7x
Accelerad 0.7 (GTX 1080 Ti) 9x
Accelerad 0.7 (RTX 8000) 14x

dcglare
rcontrib direct only
rcontrib daylight coefficient

RTX Technology

Time (minutes)
Calculating Annual DGP

1. Calculate $S$ for each hour of the year

   `gendaymtx -of NYCity.wea > sky.smx`

2. Calculate $D_{direct}$ for each view position and direction using the two-phase method

   `rcontrib -e MF:1 -f reinhartb.cal -b rbin -bn Nrbins -m sky_mat -I+ -ab 1 -ad 50000 -lw .00002 -lr -10 -faf scene.oct < views.vf > dc1.mtx`

3. Calculate $D_{total}$ for the same view positions and directions using the two-phase method or a higher-order multi-phase method

   `rcontrib -e MF:1 -f reinhartb.cal -b rbin -bn Nrbins -m sky_mat -I+ -ab 8 -ad 50000 -lw .00002 -lr -10 -faf scene.oct < views.vf > dc8.mtx`

4. Calculate DGP for each hour and view

   `dcglare -vf views.vf dc1.mtx dc8.mtx sky.smx > dgp.txt`
Calculating Glare Autonomy

1–3. As before

4. Calculate GA using a schedule and glare limit

```
dcglare -vf views.vf -sf 8to6withDST.60min.occ.csv -l .4 dc1.mtx dc8.mtx sky.smx > dgp.txt
```

8760-hour occupancy schedule (compatible with Daysim schedules)

DGP Limit

*i.e. GA*$_{40\%}$
Limitations

- Only sun and sky as glare sources
- No specular reflections (e.g. polished floors, reflective ground surfaces, or bodies of water)
- No light-redirecting fenestration systems
- No electrochromic glazing
- Still not real-time
Problem

How can simulation be an effective design tool?

Create a **stronger** connection between tool and user

Provide **faster** results for annual and spatial glare analysis

Make **better** decisions using real-time glare analysis
Problem

How do we achieve real-time rendering?
Distribution
Ray Tracing
Path Tracing

Distribution Ray Tracing

Path Tracing
Direct

Path Tracing
Diffuse
Experiment

Do faster tools make a difference?
Discomfort Glare
- Sun in the field of view

Veiling Glare
- Reflections obscure screen

Dim Lighting
- Insufficient task illumination
Design Goals

Daylight Glare Probability: < 35%
Veiling Glare: < 50 cd/m² reflected
Work Surface Illuminance: > 300 lux (~48 cd/m²)
Comfortable View

- DGP: 25%
- Screen: 25 cd/m²
- Desk: 100 cd/m²

Uncomfortable View

- DGP: 100%
- Screen: 400 cd/m²
- Desk: 150 cd/m²
Results

How do tools affect user behavior?
Average Number of Interactions

- **Time and Date**: AcceleradRT (low), DIVA-for-Rhino (low)
- **Shading**: AcceleradRT (medium), DIVA-for-Rhino (low)
- **View**: AcceleradRT (high), DIVA-for-Rhino (medium)

Results

How do tools affect design quality?
States viewed for two seconds

Pareto optimal vs. Not Pareto optimal

- AcceleradRT
- DIVA-for-Rhino
Background

Experience

Architecture  Non-Architecture

None (n = 13)  1-2 years (n = 11)  3-6 years (n = 11)  7-13 years (n = 5)

Fraction on Pareto frontier

Acceleration

DIVA-for-Rhino
Results

How do tools affect user satisfaction?
More confident in glare assessment: 58% for AcceleradRT, 13% for DIVA-for-Rhino
More confident in final design performance: 58% for AcceleradRT, 23% for DIVA-for-Rhino
More familiar tool: 18% for AcceleradRT, 55% for DIVA-for-Rhino

preferred overall: AcceleradRT
Accelerad

rpict  100x faster
rtrace  28x faster
rcontrib  45x faster
dcglare  133,000x faster

AcceleradRT  real-time
AcceleradRT from the command line

1. False color hemispherical view

   ```
   AcceleradRT -vp 85 21 46 -vd 0 -1 0 -vu 0 0 1 -vta-vv 180 -vh 180 -ab 3
                -aa 0 -ad 1 -x 512 -y 512 -s 10000 -log 3 -m 0.1 scene.oct
   ```

2. Cinematic view

   ```
   AcceleradRT -vp 85 21 46 -vd 0 -1 0 -vu 0 0 1 -vtv-vv 40 -vh 60 -ab 3
                -aa 0 -ad 1 -x 1920 -y 1080 -s 10000 -log 0 scene.oct
   ```
Real-Time Daylighting Model with
1 Billion Polygons
nathaniel.jones@arup.com

https://nljones.github.io/

https://nljones.github.io/Accelerad/