EQUA.
SIMULATION TECHNOLOGY GROUP

Faster Daylight Simulation

Master's Thesis Proposal · Oct 28, 2026

Context

Annual daylight simulation is accurate but slow, and that cost limits how often designers can use it. IDA ICE, Equa's building performance simulation tool, computes daylight with Radiance, the reference ray tracer for lighting simulation. Two kinds of calculation are offered:

- **Point-in-time**: illuminance under one sky, by backward ray tracing. Accurate, but too slow to repeat for every hour of a year.
- **Climate-based (annual)**: daylight coefficients computed once, then combined with a full year of sky conditions (the matrix, or three-phase, method). Faster per hour, but the coefficient step is still the bottleneck, and its cost grows with the number of sensor points, sky patches and ambient bounces.

In both, Radiance's Monte Carlo sampling sets the trade-off: fewer rays mean faster runs and noisier results.

Equa's daylight engine is covered by automated tests. This makes it practical to try new calculation methods and measure them against the current one.

Overall goal

The thesis designs, prototypes and evaluates a method that makes daylight simulation significantly faster while keeping its accuracy within a stated bound. Three principles apply to every direction:

- **Radiance stays the physics.** New methods accelerate or approximate Radiance's results; Radiance at high precision is the reference they are checked against.
- **Speed and accuracy are measured together.** Every result reports run time and error against the reference, on the same cases and the same hardware.
- **Visual AI is a tool, not a replacement.** Learned methods are welcome where they reduce computation, as long as their error is measured and bounded.

The student is expected to identify and test their own improvements to the chosen method. An improvement that does not raise performance is still a valid result, as long as it is tested and its outcome explained.

Prime example: fewer rays, learned denoising

Run Radiance with far fewer samples than usual, then remove the resulting noise with a trained denoiser. Production renderers already do this for images (for example NVIDIA's OptiX denoiser and Intel's Open Image Denoise); the question here is whether it holds for physically meaningful daylight quantities.

Research questions

1. How much faster is a low-sample run plus denoising than a full-precision Radiance run, for the same error?
2. Which representation suits the denoiser: illuminance on sensor grids, daylight-coefficient matrices, or rendered HDR images?
3. How well does a denoiser trained on one set of buildings generalize to new geometry, materials and climates?
4. Where does the method fail, and how can the error be estimated without a reference run?

Possible tasks

- Generate training data with the daylight engine: pairs of low-sample and high-sample results over varied rooms, windows and skies.
- Design and train a denoising model suited to the chosen representation, using the geometry (normals, depth, distance to windows) as extra input channels.
- Benchmark against full-precision Radiance on reference cases: run time, error per sensor point, and errors in standard metrics such as daylight factor and daylight autonomy.

Optional extension: a GPU ray-tracing backend.

Accelerate the sampling step itself by computing daylight coefficients on the GPU with a modern ray-tracing API (OptiX, Vulkan ray tracing or CUDA). Earlier work, notably Accelerad, showed that GPU Radiance can match CPU accuracy; this extension asks how far current hardware ray tracing goes, alone or combined with denoising. The denoising work does not depend on it.

Other possible directions

The prime example is a suggestion, not a requirement. A student may pick another direction that fits their interests better:

Direction	How it saves time	Visual AI technique
Neural surrogate	A network predicts illuminance maps or annual metrics directly from the room geometry; Radiance produces the training data and checks the predictions	Image-to-image networks (U-Net), graph neural networks, neural fields

Direction	How it saves time	Visual AI technique
Annual glare from images	Annual glare needs thousands of fisheye HDR renderings; render a few and predict the rest, or predict glare metrics from low-resolution images	HDR image synthesis, image-based regression
Super-resolution of sensor grids	Compute a coarse grid of sensor points, then learn to upsample it to the detail designers need	Super-resolution
GPU ray tracing	Compute daylight coefficients on the GPU instead of the CPU (see the optional extension above)	Hardware ray tracing

A student's own idea fits if it:

- makes daylight simulation faster, measured on the same cases and hardware as the current method;
- uses Radiance as its reference, with its error stated.

State of the art

The student starts from the published work below and positions their contribution against it; full entries are under References.

Area	Key work
Radiance and annual methods	Ward (1994); Reinhart & Walkenhorst (2001); McNeil & Lee (2013); Subramaniam (2017)
GPU daylight simulation	Jones & Reinhart (2015, 2017); Jones (2017); GPU acceleration framework for climate-based daylight (2024)
Monte Carlo denoising	Zwicker et al. (2015); Kalantari et al. (2015); Bako et al. (2017); Chaitanya et al. (2017); Huo & Yoon (2021); Müller et al. (2021); Intel Open Image Denoise; NVIDIA OptiX denoiser
Machine learning for daylight	Ayoub (2020); He et al. (2021); Liu et al. (2023); Le et al. (2025)
Glare	Wienold & Christoffersen (2006)

Expected outcome

- A working prototype of the chosen method, running on Equa's daylight engine.
- A benchmark comparing the prototype with current Radiance runs: speed-up, error per sensor point and error in standard daylight metrics, on cases of increasing size.

- An analysis of where the method works, where it fails, and why.
- Recommendations for bringing the method into production, including the hardware it needs.
- A Master's thesis report, and optionally a conference or journal paper.

What Equa provides

- **The daylight engine**: Equa's Python package built on Radiance, with its tests.
- **Reference cases**: IDA ICE daylight cases with reference results, from simple rooms to real buildings.
- **GPU hardware**: one NVIDIA GB10 AI workstation for development, AI training and experiments.
- **Supervision**: regular meetings with Equa's daylight and software developers, with access to their knowledge of IDA ICE and the platform.

Background and required skills

Required

- Strong programming in Python; C/C++ or CUDA is a plus.
- Deep learning for images: convolutional networks, training, evaluation (PyTorch or similar).
- Comfort with 3D geometry and numerical computation.

Welcome, not required

- Ray tracing and Monte Carlo rendering.
- GPU programming (CUDA, OptiX, Vulkan).
- Interest in lighting, buildings or sustainability. Daylight physics and Radiance are learned during the thesis, with support from Equa.

Practicalities

- **Location**: at Equa's offices in Solna or Göteborg, or remotely with planned physical meetings.
- **Start**: open question, to be agreed.
- **Contact**: Sylvain Tourancheau (sylvain.tourancheau@equa.se)

References

Radiance and annual methods

1. Ward, G. J. (1994). The RADIANCE lighting simulation and rendering system. *Proceedings of SIGGRAPH '94*, 459–472. LBL
2. Reinhart, C. F., & Walkenhorst, O. (2001). Validation of dynamic RADIANCE-based daylight simulations for a test office with external blinds. *Energy and Buildings*, 33(7), 683–697. https://doi.org/10.1016/S0378-7788(01)00058-5
3. McNeil, A., & Lee, E. S. (2013). A validation of the Radiance three-phase simulation method for modelling annual daylight performance of optically complex fenestration systems. *Journal of Building Performance Simulation*, 6(1), 24–37. https://doi.org/10.1080/19401493.2012.671852
4. Subramaniam, S. (2017). *Daylighting simulations with Radiance using matrix-based methods* (tutorial). radiance-online.org

GPU daylight simulation

5. Jones, N. L., & Reinhart, C. F. (2015). Fast daylight coefficient calculation using graphics hardware. *Proceedings of Building Simulation 2015*. https://doi.org/10.26868/25222708.2015.2462
6. Jones, N. L., & Reinhart, C. F. (2017). Speedup potential of climate-based daylight modelling on GPUs. *Proceedings of Building Simulation 2017*. PDF
7. Jones, N. L. (2017). *Validated interactive daylighting analysis for architectural design* (PhD thesis, MIT). DSpace@MIT
8. Exploring GPU acceleration framework for climate-based daylight modeling (2024). *Building Simulation*. https://doi.org/10.1007/s12273-024-1207-5

Monte Carlo denoising

9. Zwicker, M., Jarosz, W., Lehtinen, J., Moon, B., Ramamoorthi, R., Rousselle, F., Sen, P., Soler, C., & Yoon, S.-E. (2015). Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. *Computer Graphics Forum*, 34(2), 667–681. https://doi.org/10.1111/cgf.12592
10. Kalantari, N. K., Bako, S., & Sen, P. (2015). A machine learning approach for filtering Monte Carlo noise. *ACM Transactions on Graphics*, 34(4). https://doi.org/10.1145/2766977
11. Bako, S., et al. (2017). Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Transactions on Graphics*, 36(4), 97. https://doi.org/10.1145/3072959.3073708
12. Chaitanya, C. R. A., Kaplanyan, A. S., Schied, C., Salvi, M., Lefohn, A., Nowrouzezahrai, D., & Aila, T. (2017). Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics*, 36(4), 98. https://doi.org/10.1145/3072959.3073601

13. Huo, Y., & Yoon, S.-E. (2021). A survey on deep learning-based Monte Carlo denoising. *Computational Visual Media*, 7(2), 169–185. https://doi.org/10.1007/s41095-021-0209-9
14. Müller, T., Rousselle, F., Novák, J., & Keller, A. (2021). Real-time neural radiance caching for path tracing. *ACM Transactions on Graphics*, 40(4), 36. https://doi.org/10.1145/3450626.3459812
15. Intel Open Image Denoise (software). openimagedenoise.github.io
16. NVIDIA OptiX AI denoiser (software). developer.nvidia.com

Machine learning for daylight

17. Ayoub, M. (2020). A review on machine learning algorithms to predict daylighting inside buildings. *Solar Energy*, 202, 249–275. https://doi.org/10.1016/j.solener.2020.03.104
18. He, Q., Li, Z., Gao, W., Chen, H., Wu, X., Cheng, X., & Lin, B. (2021). Predictive models for daylight performance of general floorplans based on CNN and GAN: a proof-of-concept study. *Building and Environment*, 206, 108346. https://doi.org/10.1016/j.buildenv.2021.108346
19. Liu, Q., Chen, Y., Liu, Y., Lei, Y., Wang, Y., & Hu, P. (2023). A review and guide on selecting and optimizing machine learning algorithms for daylight prediction. *Building and Environment*, 110822. https://doi.org/10.1016/j.buildenv.2023.110822
20. Le, T. L., Chong, H., Le, B. M., Nguyen-Xuan, H., & Kim, S. A. (2025). From hand sketches to daylight performance: a mixed-input neural prediction framework. *Journal of Building Performance Simulation*. https://doi.org/10.1080/19401493.2025.2499689

Glare

21. Wienold, J., & Christoffersen, J. (2006). Evaluation methods and development of a new glare prediction model for daylight environments with the use of CCD cameras. *Energy and Buildings*, 38(7), 743–757. https://doi.org/10.1016/j.enbuild.2006.03.017