

Visualisation with Open DX

Experiences on the use of OpenDX in the Radiance workflow

OpenDX



Open source version of IBM Visualization Data Explorer.



Compiles on Unix, Linux, Windows & Mac



Free!

<http://www.opendx.org/>



OpenDX



Easy to use



Easy to configure/customize

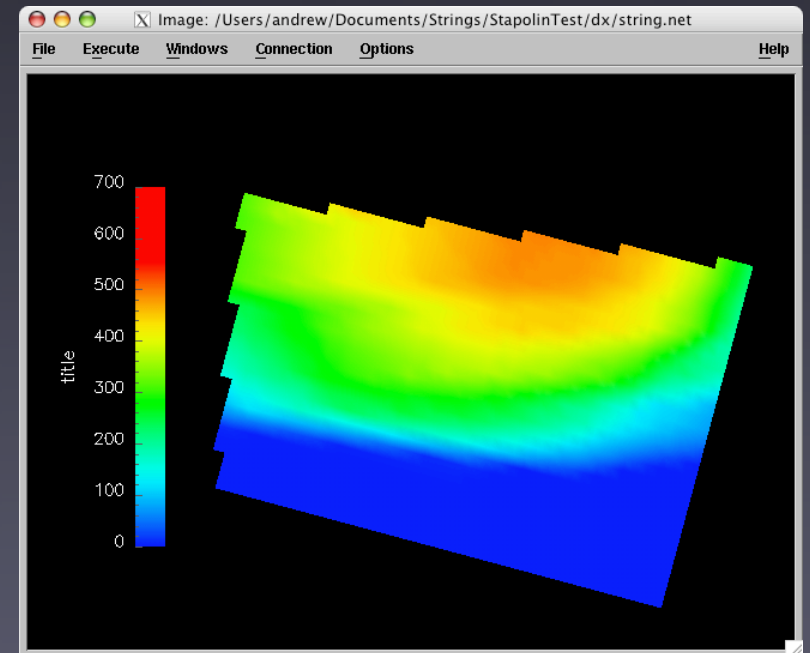
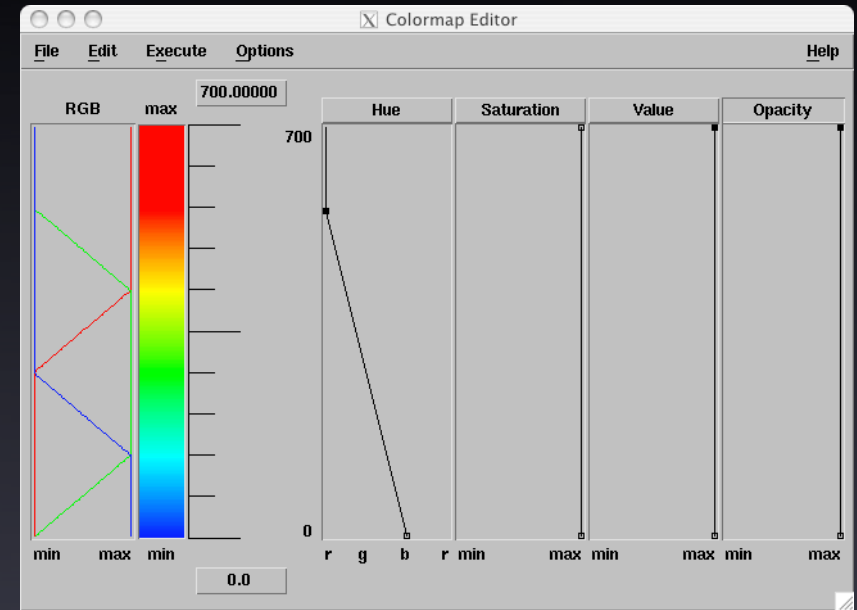
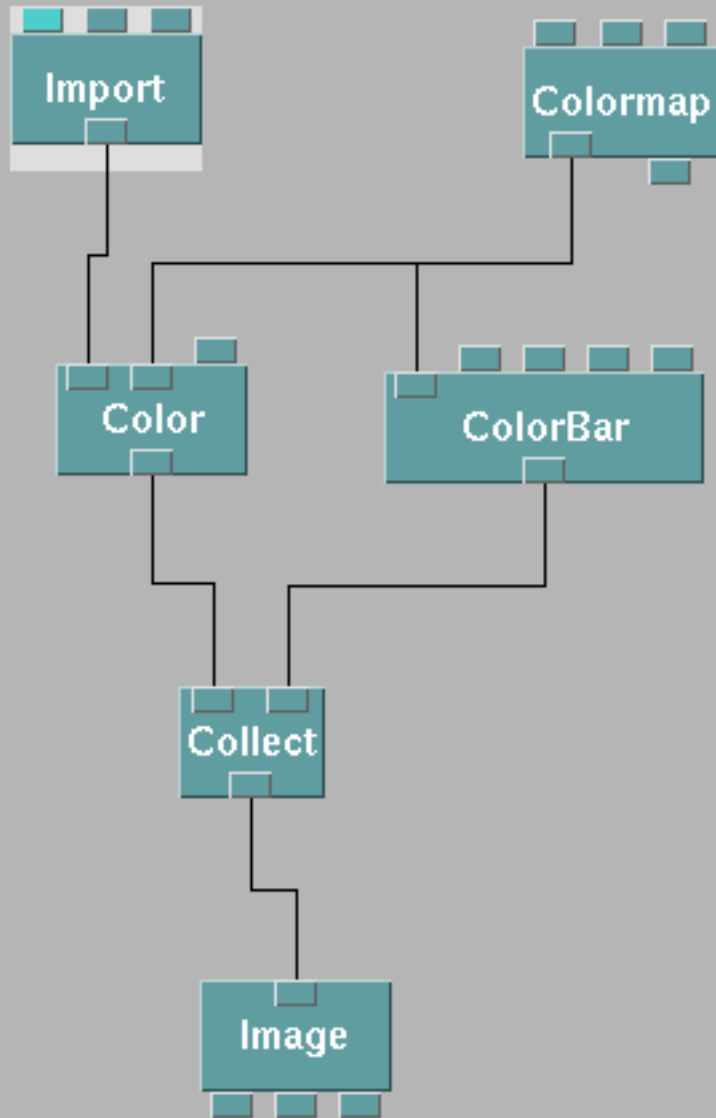


Versatile

<http://www.opendx.org/>



Simple Visualization



Importing data

data file structure:

header

x = 26			
y = 8			
z = 1			
0.05	-0.05	0.5	0.0902690198
0.05	-0.05	1	0.0882096427
0.05	-0.05	1.5	0.101252317
0.05	-0.05	2	0.10056587
0.05	-0.05	2.5	0.105371071
0.05	-0.05	3	0.092328379
0.05	-0.05	3.5	0.0739656819
0.05	-0.05	4	0.0547448946
0.05	-0.56	0.5	0.098506492
...

X Y Z datum



Data structure is totally free. For our convenience we typically use the following scheme:

x y z datum

Custom import GUI

File Options Help





Data file name

I

Select the format of your data:

- ☐ Data Explorer file
- ☐ CDF format
- ☐ NetCDF format file
- ☐ HDF format
- ☐ Image file
- ☒ Grid or Scattered file (General Array Format)

Grid type

Number of variables

☐ Positions in data file

☒ Single time step

Data organization: ☒ Block ☐ Columnar

Browse Data... Test Import...

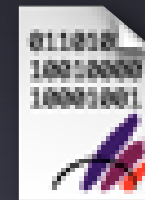
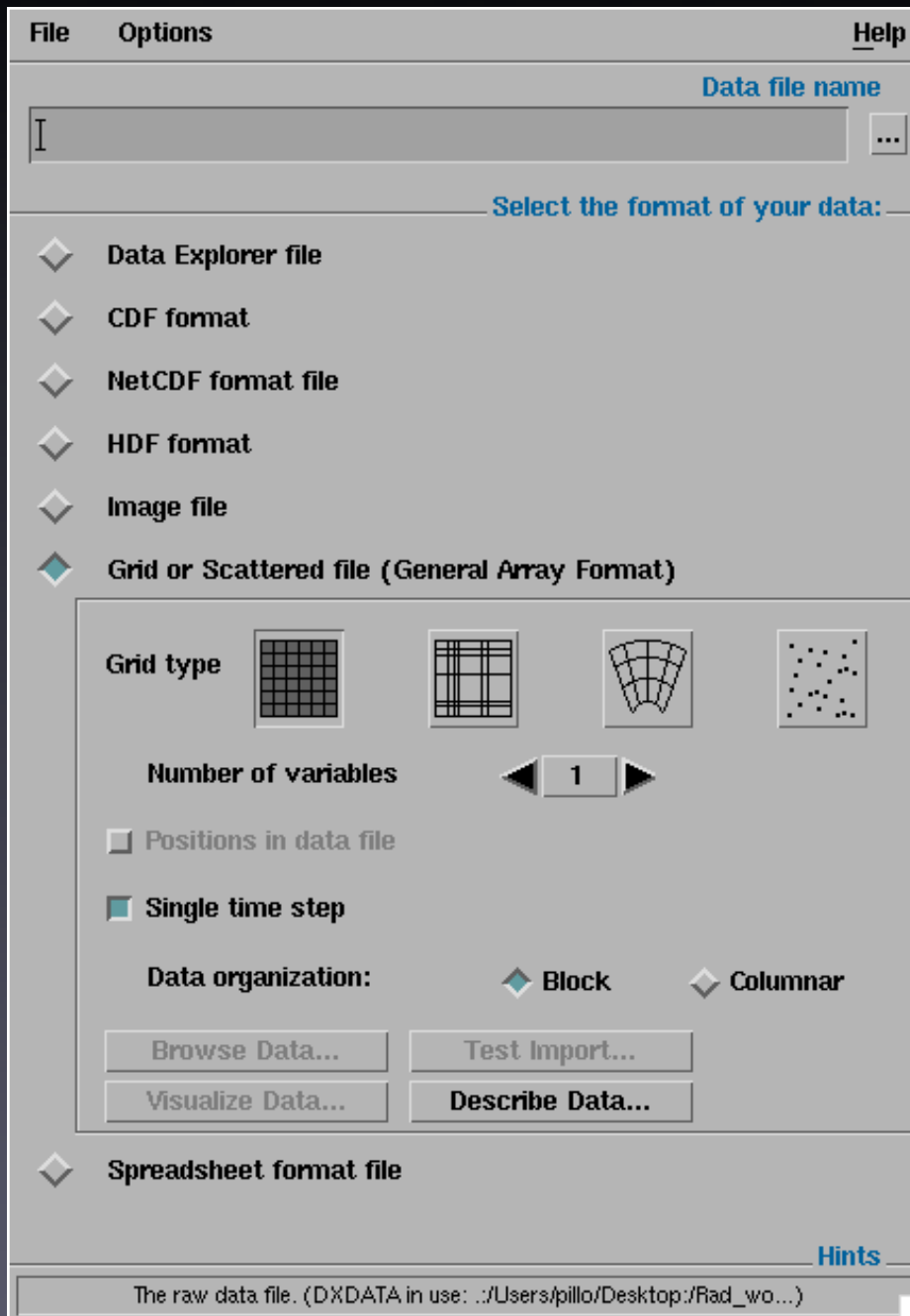
Visualize Data... Describe Data...

☐ Spreadsheet format file

Hints

The raw data file. (DXDATA in use: ..\Users\pillo\Desktop\Rad_wo...)

Custom import GUI



Saves a data
structure template.

Importing the 3d model

Two ways to import geometry:



via **.stl* file format with the command `admesh`
(Anthony D. Martin 1995)



via **.rad* file format with `rad2odx`
(Andrew Mc Neil 2005)

rad2odx



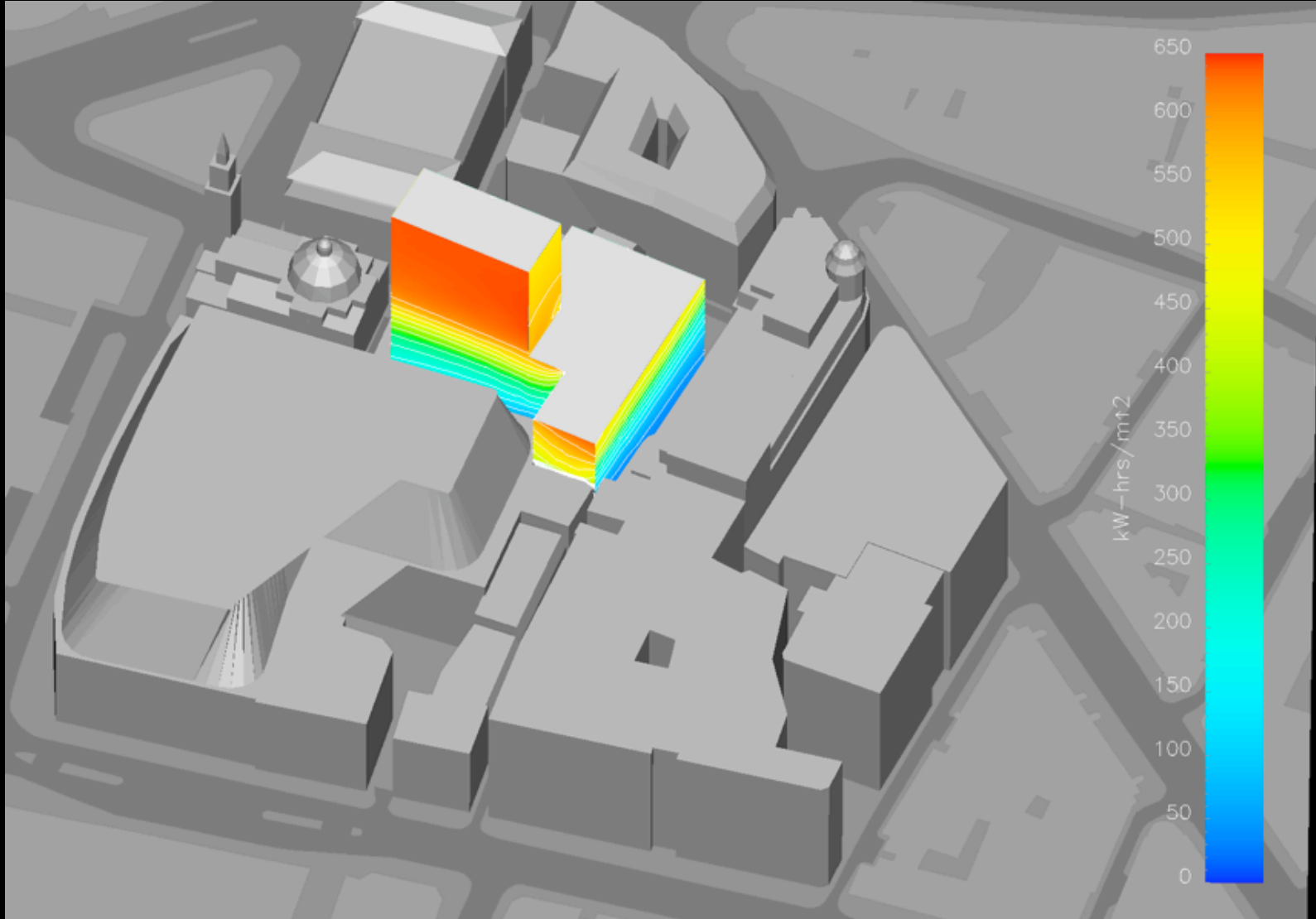
Supports n-sided polygons only.



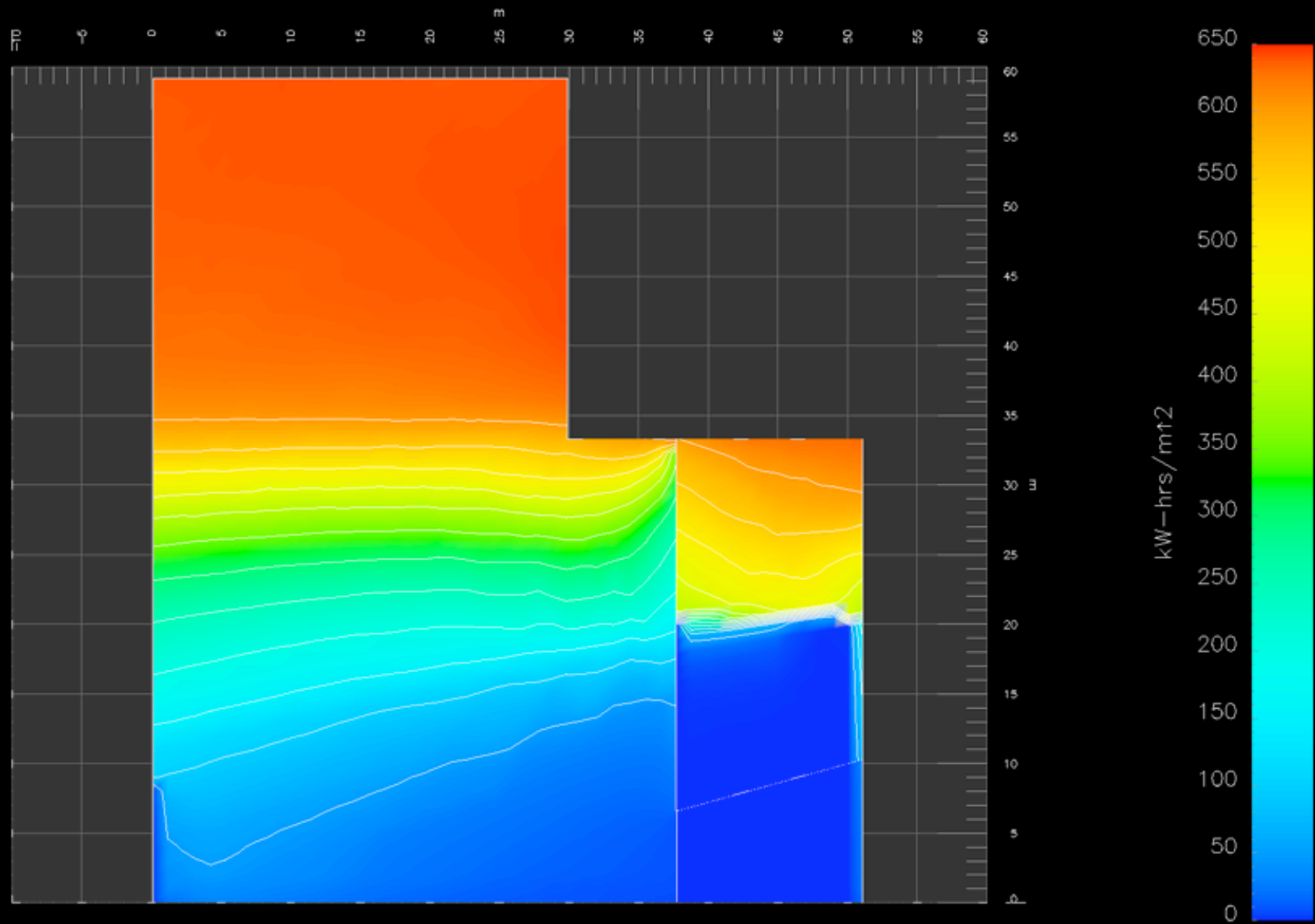
Maintains object colour and transparency.

Case studies

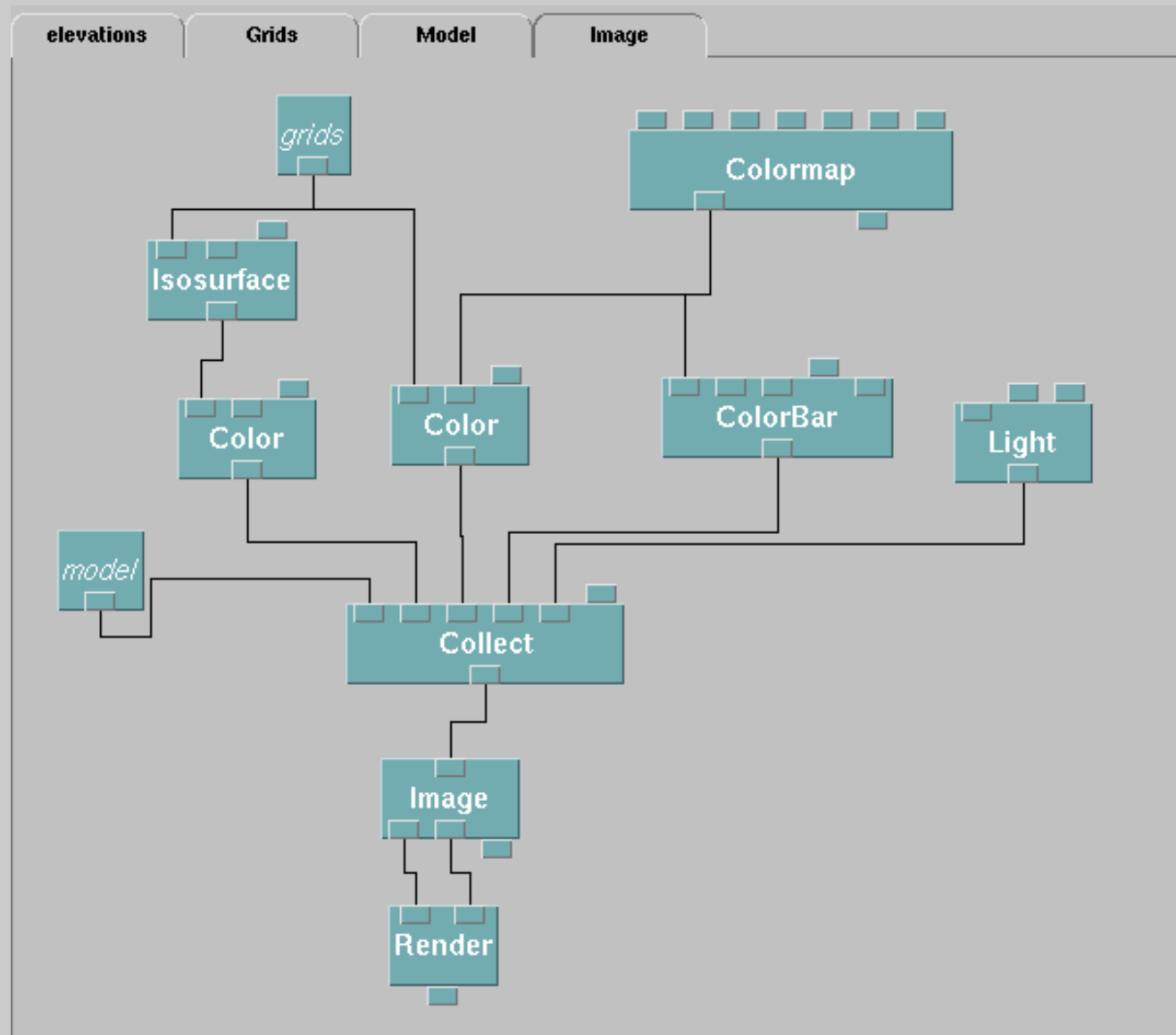
Bank Facade Irradiance



Bank

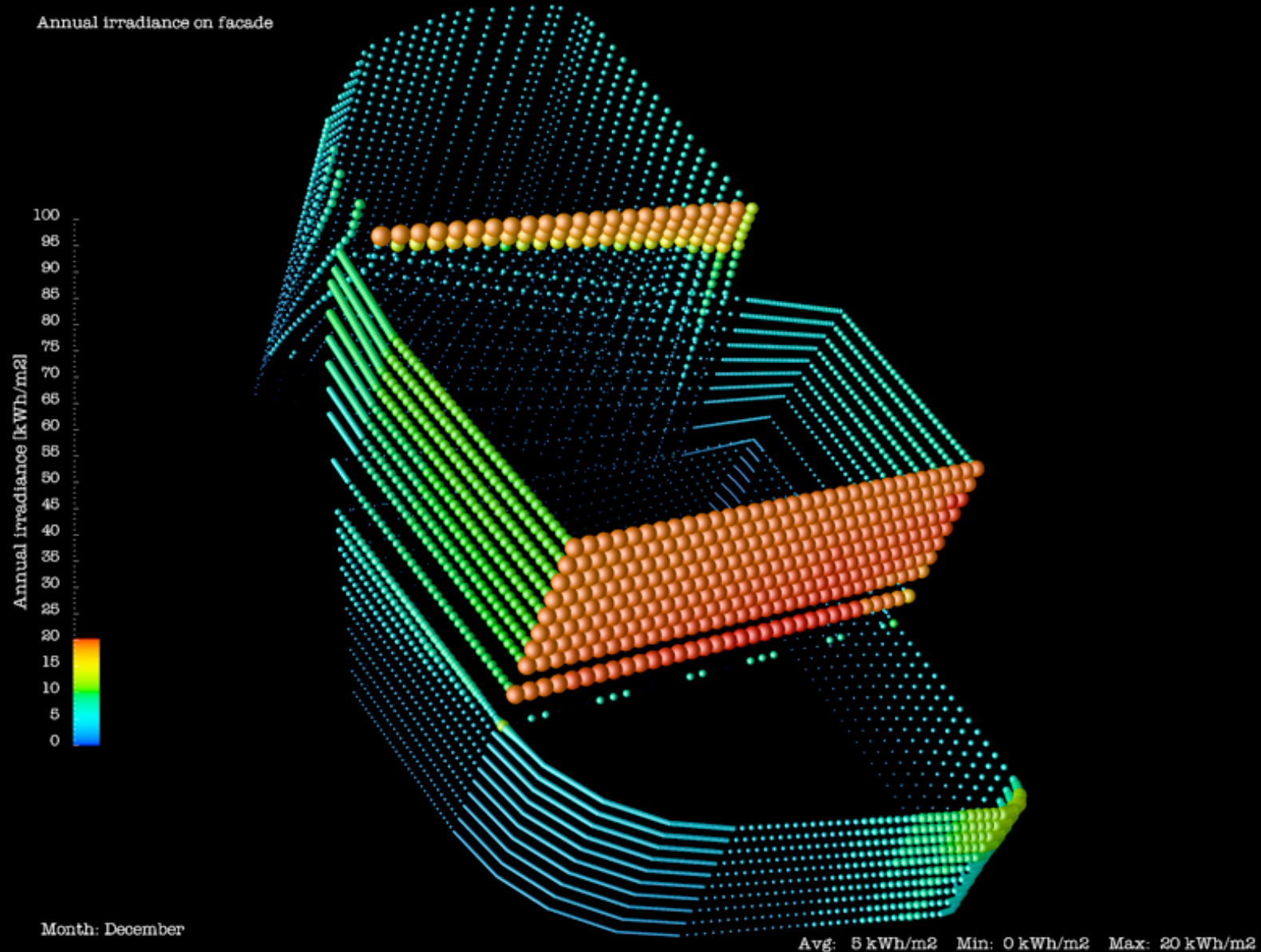


OpenDX map

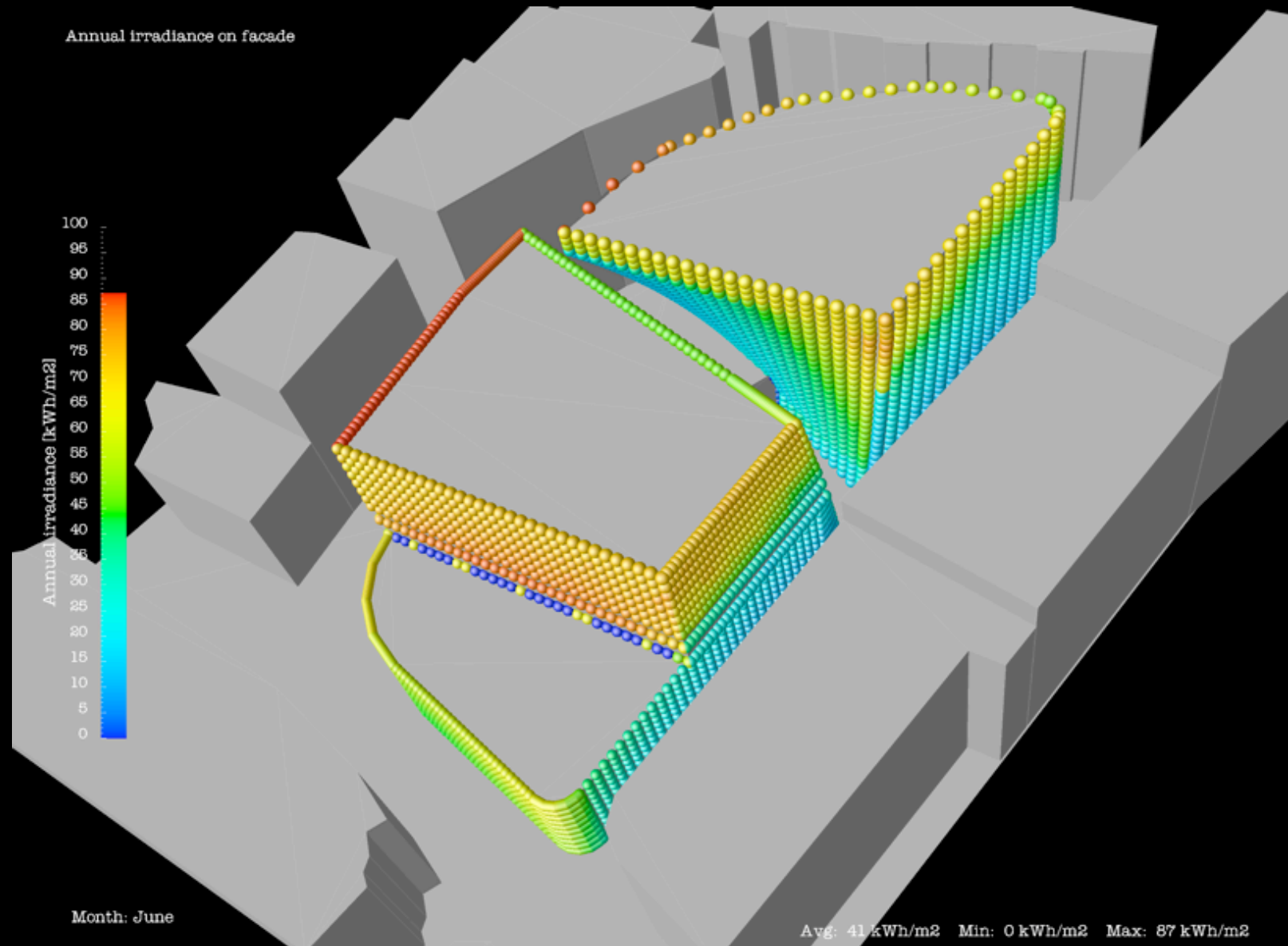


Old Baileys development

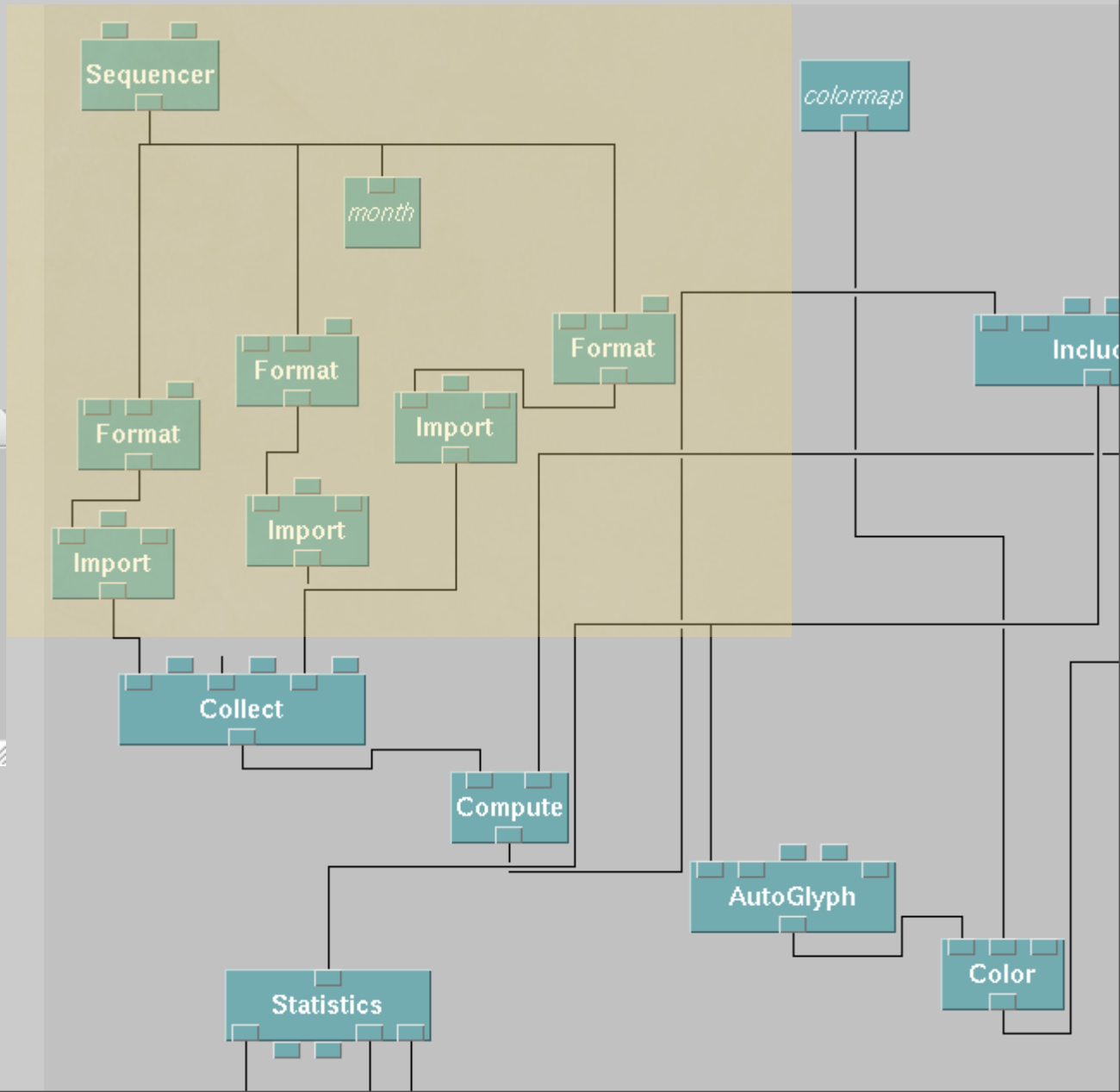
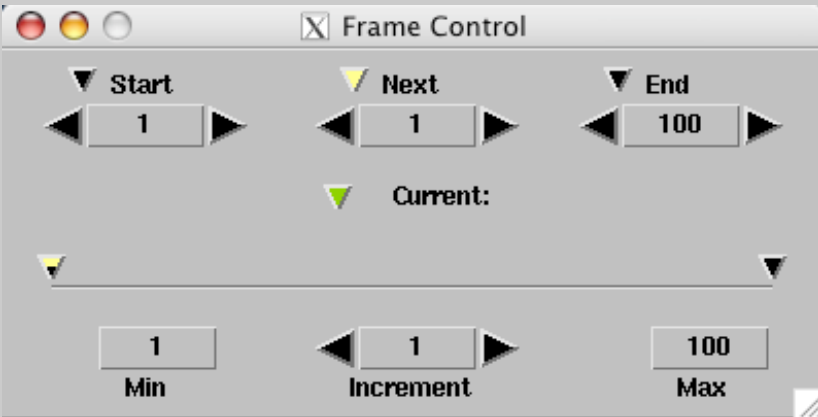
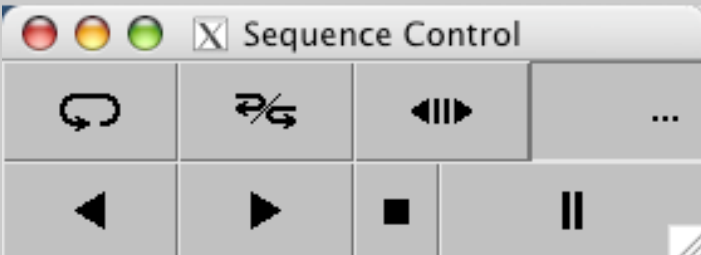
Experimenting with sequencer module



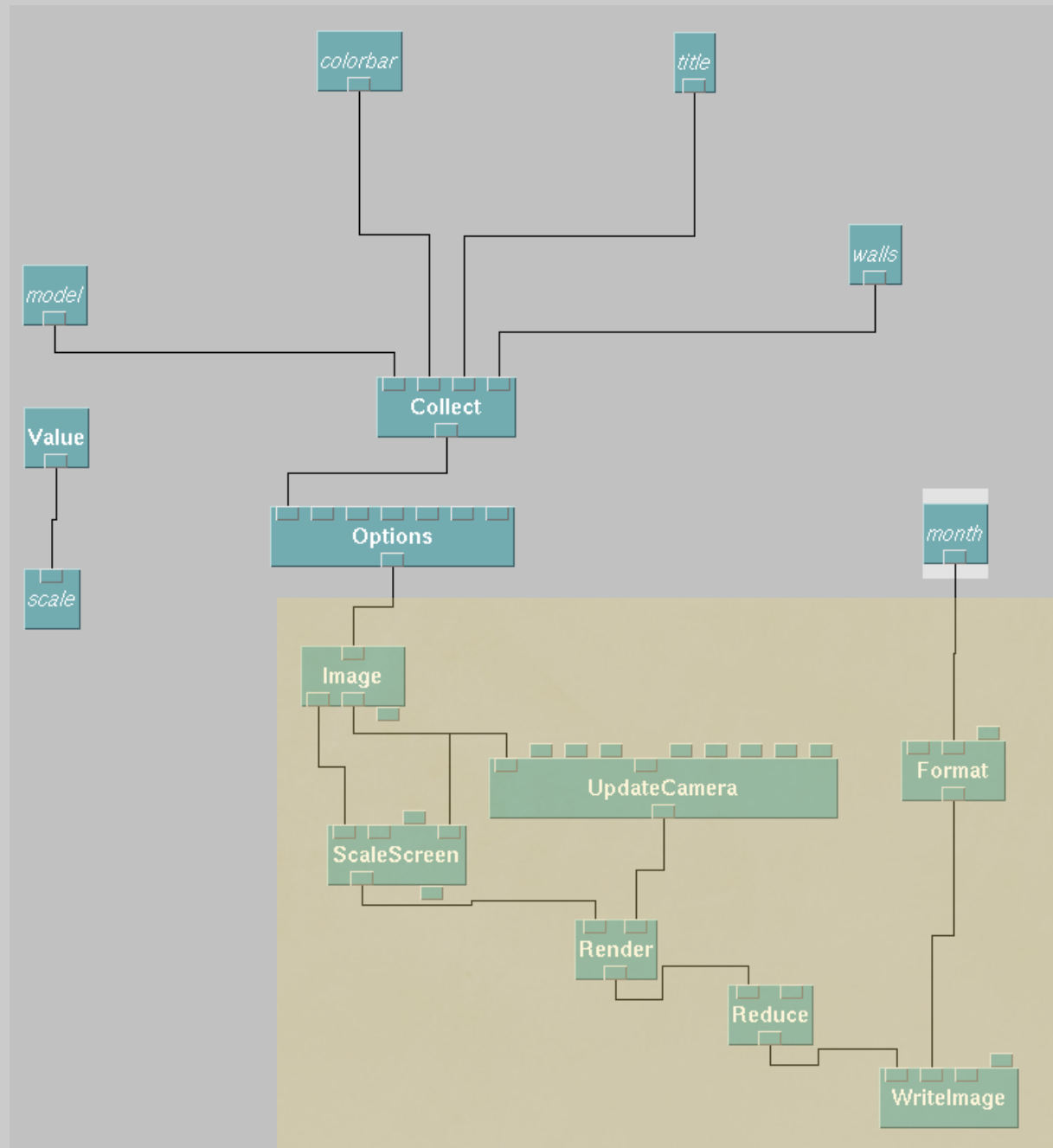
Old Baileys



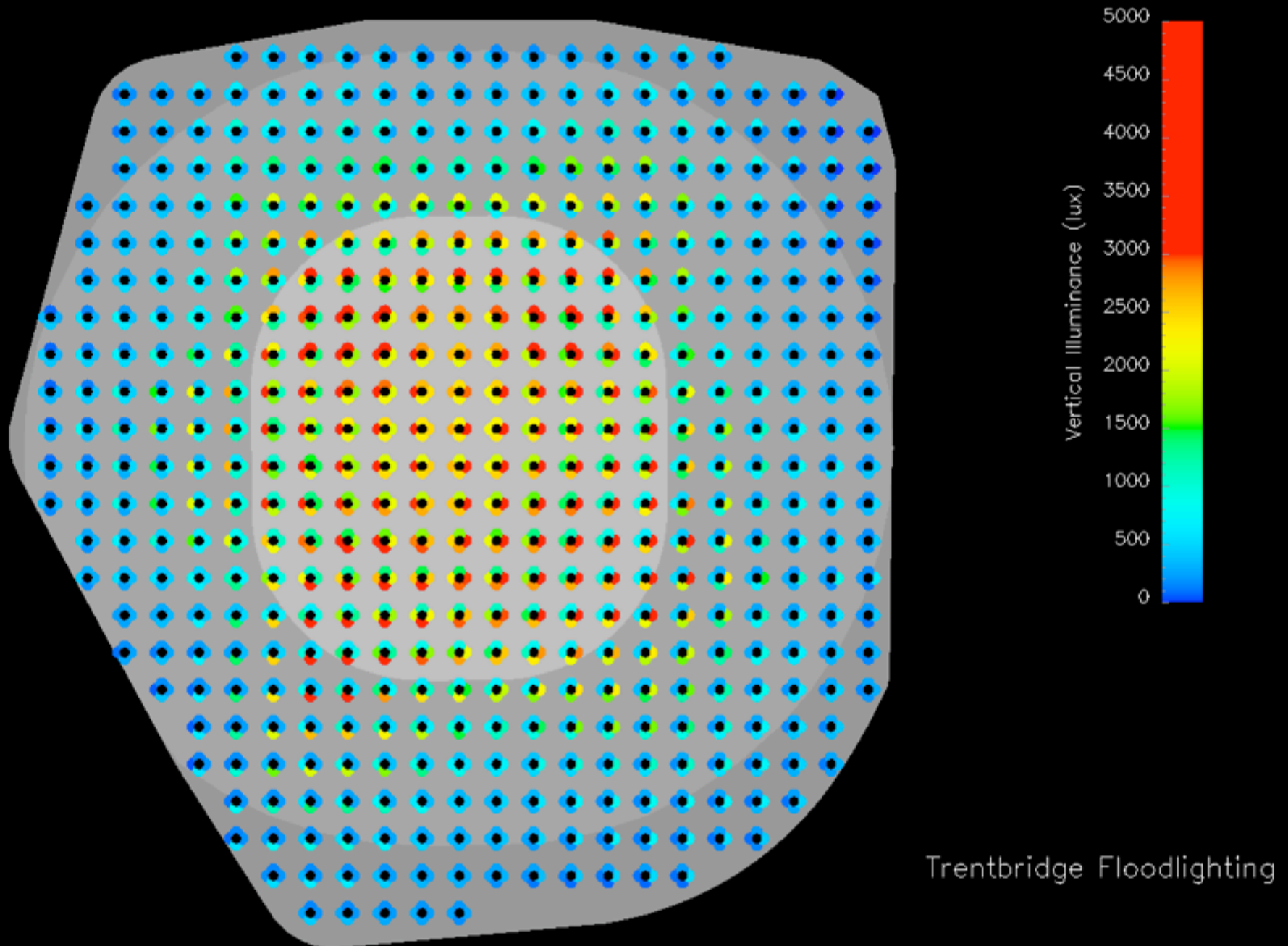
OpenDX map highlights



OpenDX map highlights



Trentbridge Floodlighting



Data Format

data file structure:

points = 470

header

76.0	25.0	1.5	151.1 196.6 458.9 260.6
82.0	25.0	1.5	167.1 209.7 365.3 278.6
88.0	25.0	1.5	183.0 223.6 279.2 267.5
94.0	25.0	1.5	197.4 237.5 259.3 294.5
100.0	25.0	1.5	209.3 253.7 175.8 247.4
70.0	31.0	1.5	151.2 197.4 536.6 180.9
76.0	31.0	1.5	170.0 211.6 662.7 324.4
82.0	31.0	1.5	189.9 226.5 646.3 415.2
88.0	31.0	1.5	210.4 242.7 448.4 365.2
...

X Y Z datum (vertical luminance in four directions)



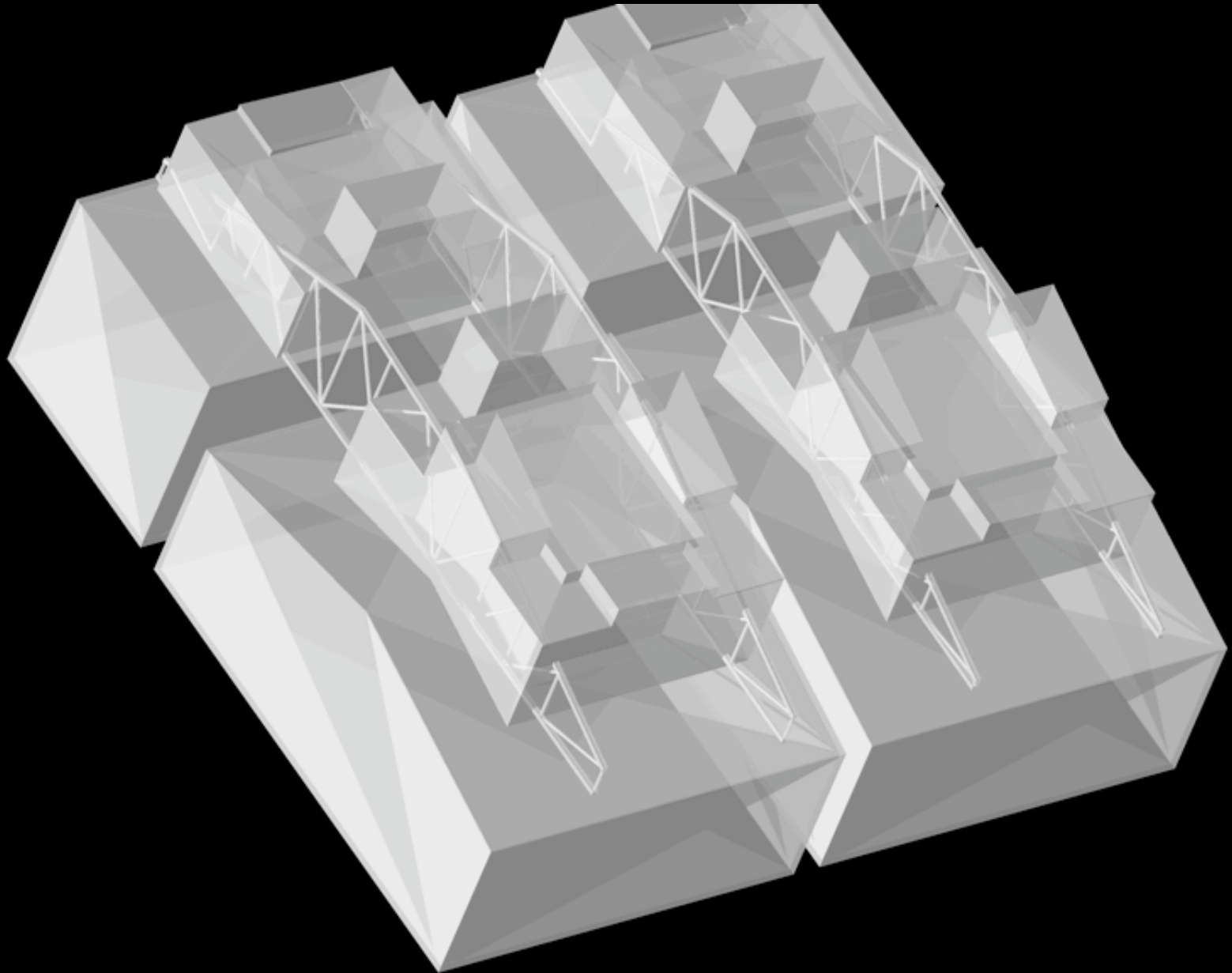
To better suit the needs, an modified data import format was used.



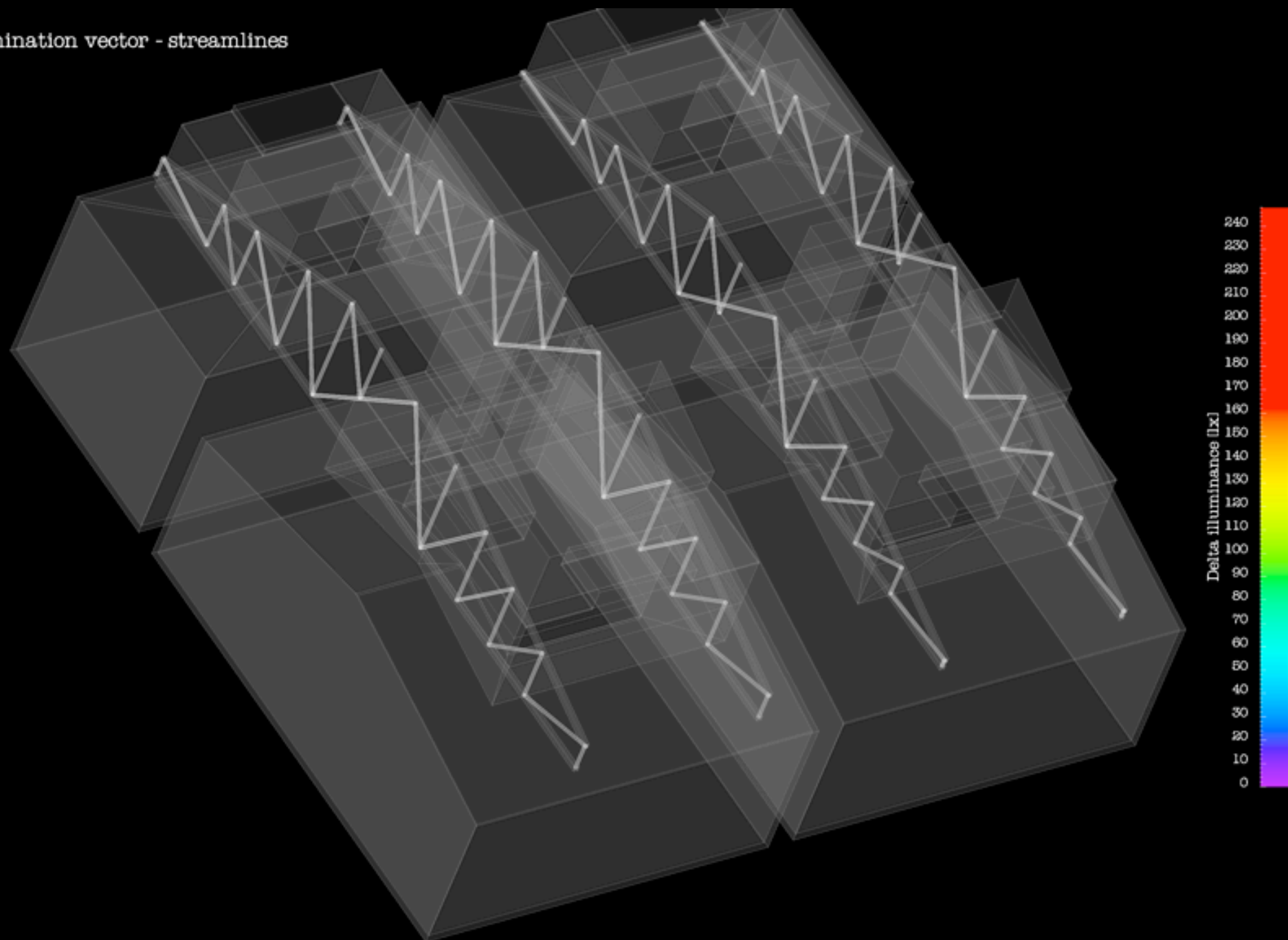
A scattered data type was used instead of an interconnected grid.

Tenerife Art museum

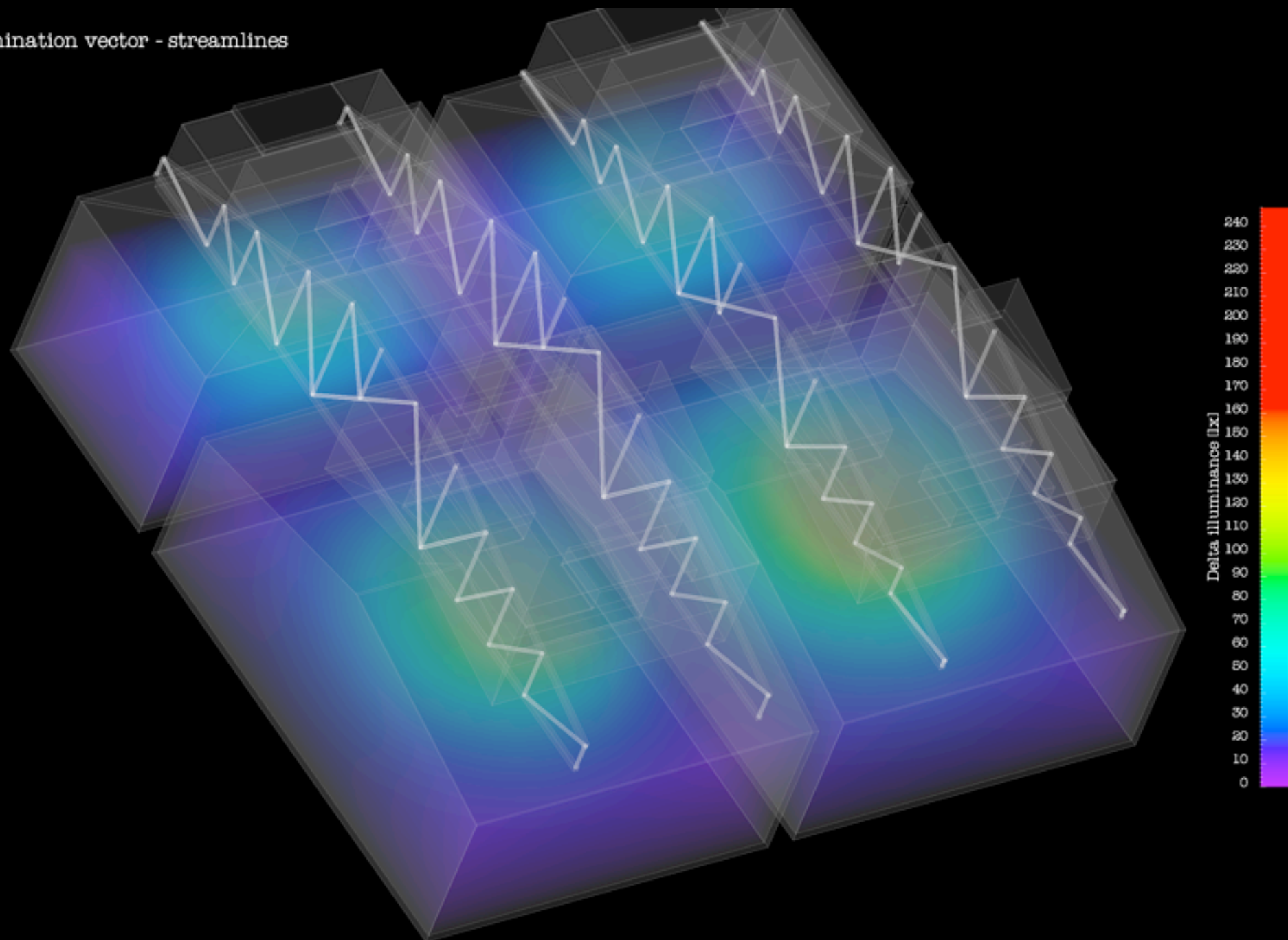
Experimenting with vector visualisation



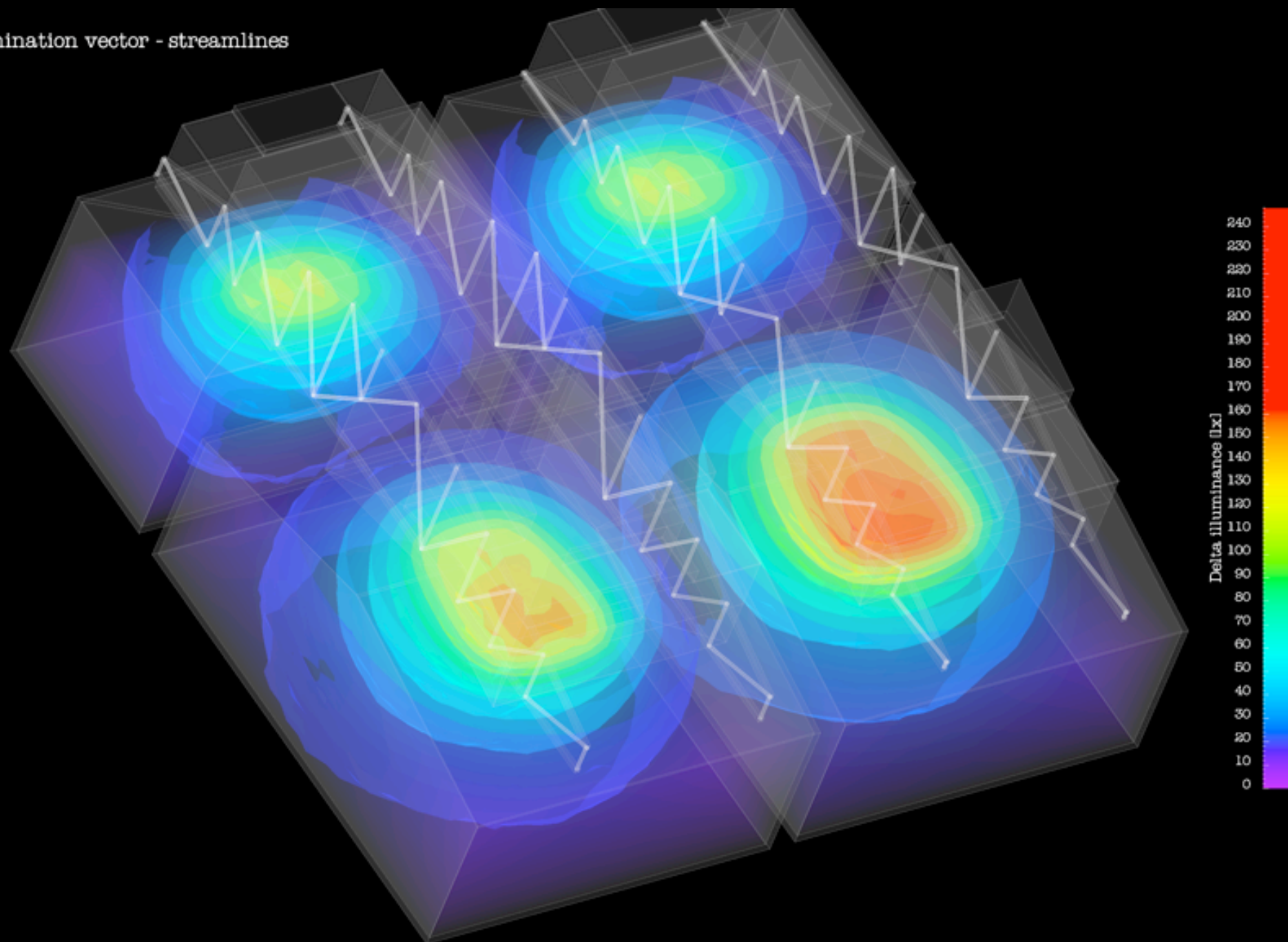
Illumination vector - streamlines



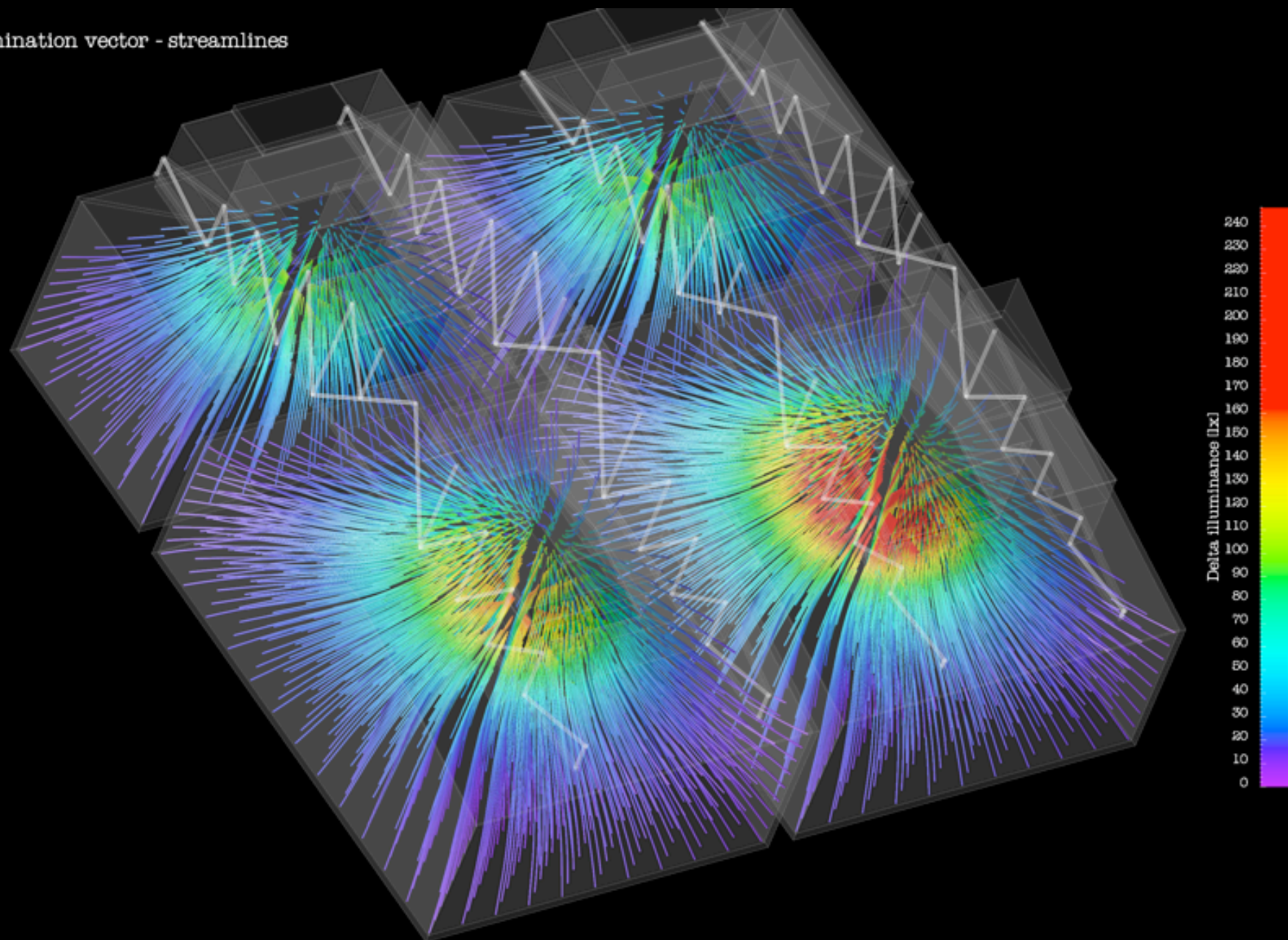
Illumination vector - streamlines



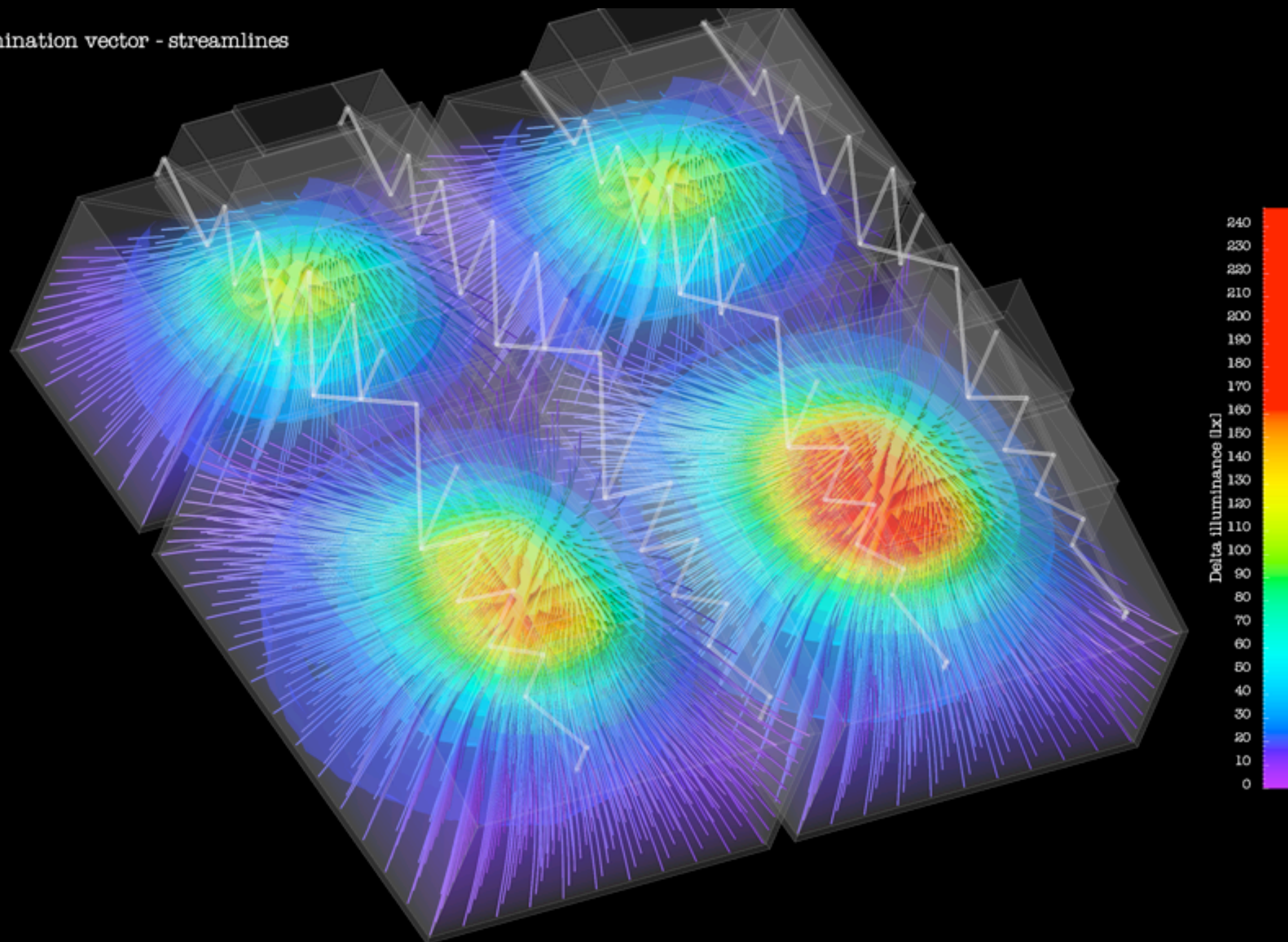
Illumination vector - streamlines



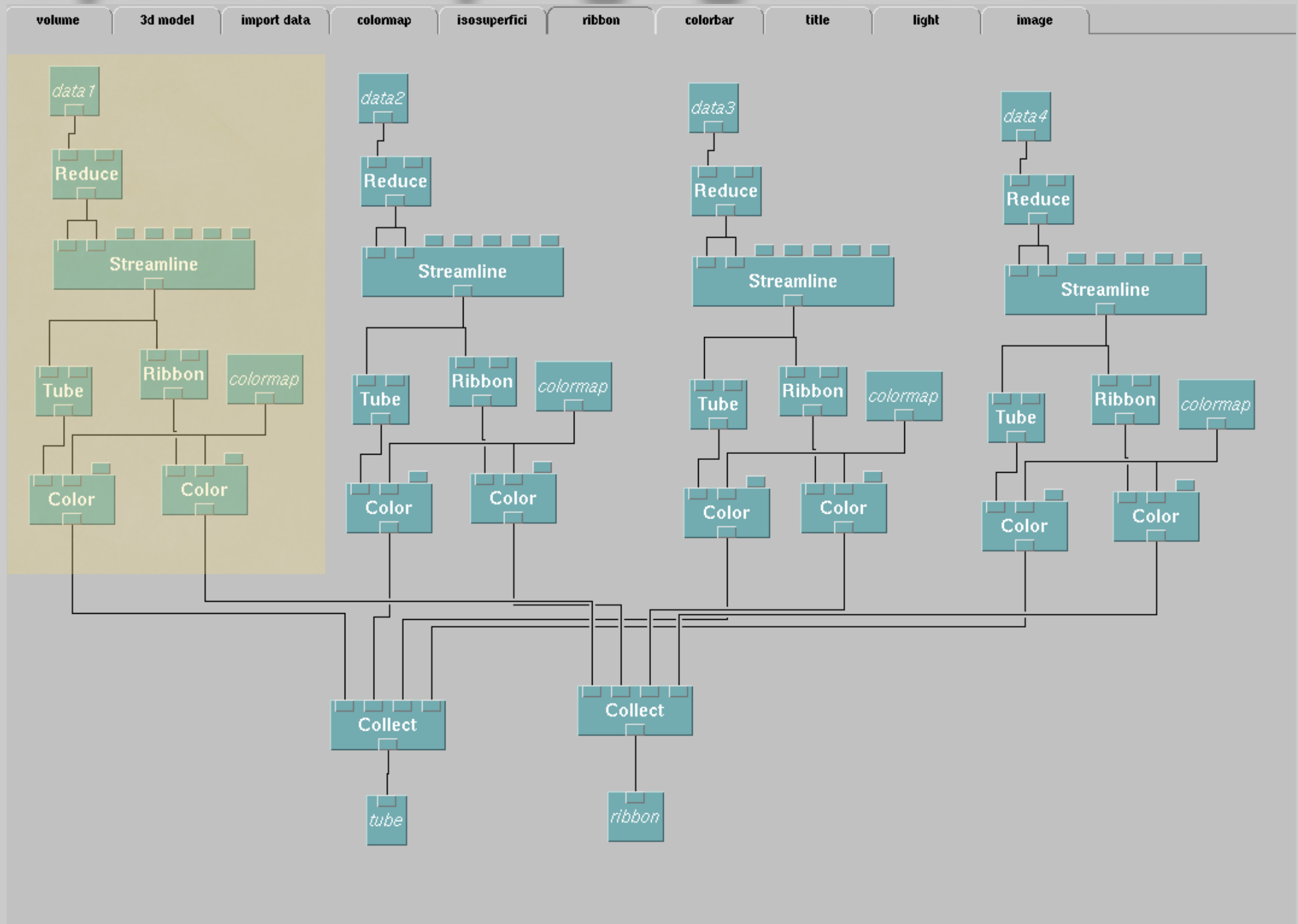
Illumination vector - streamlines



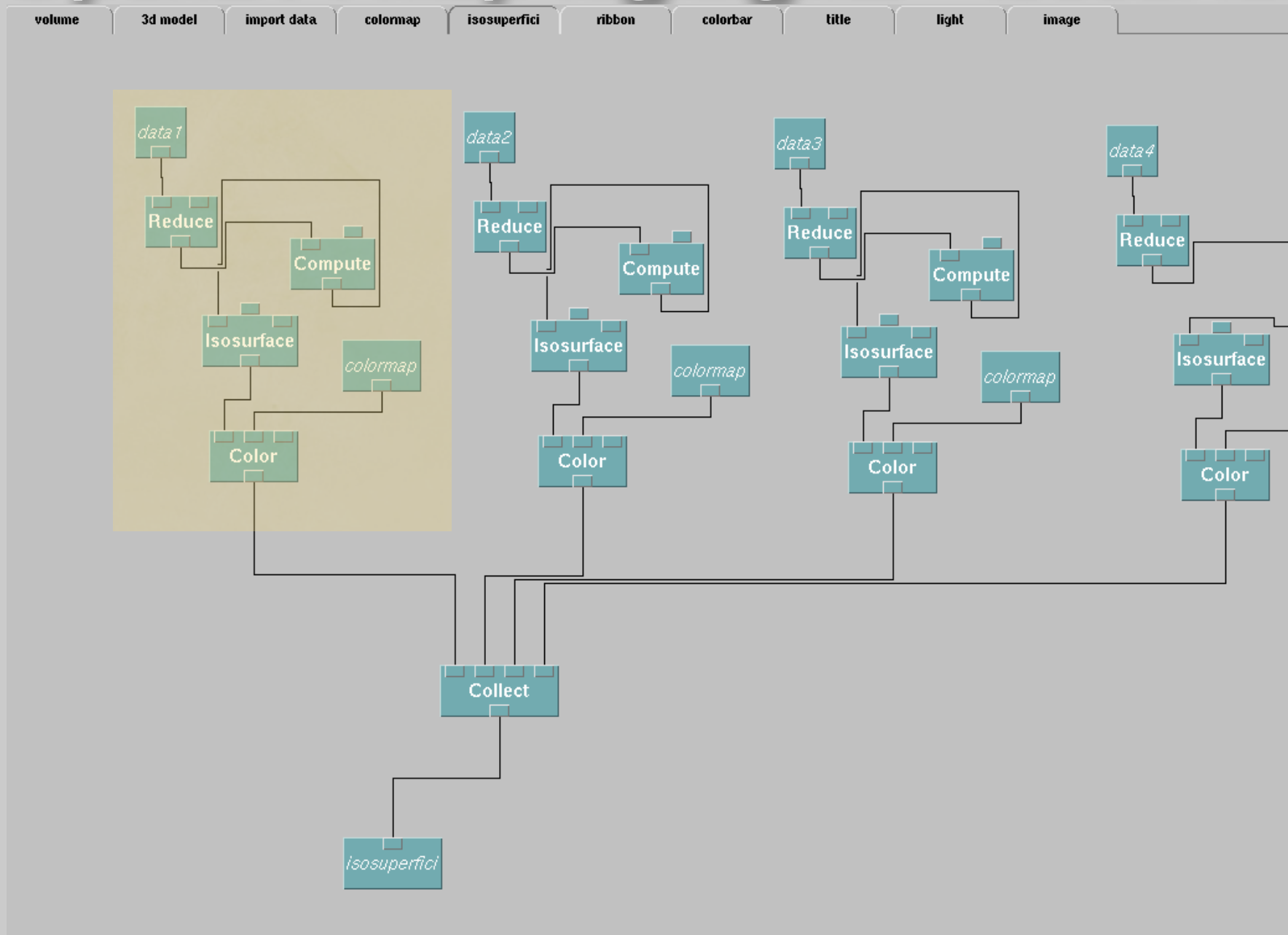
Illumination vector - streamlines



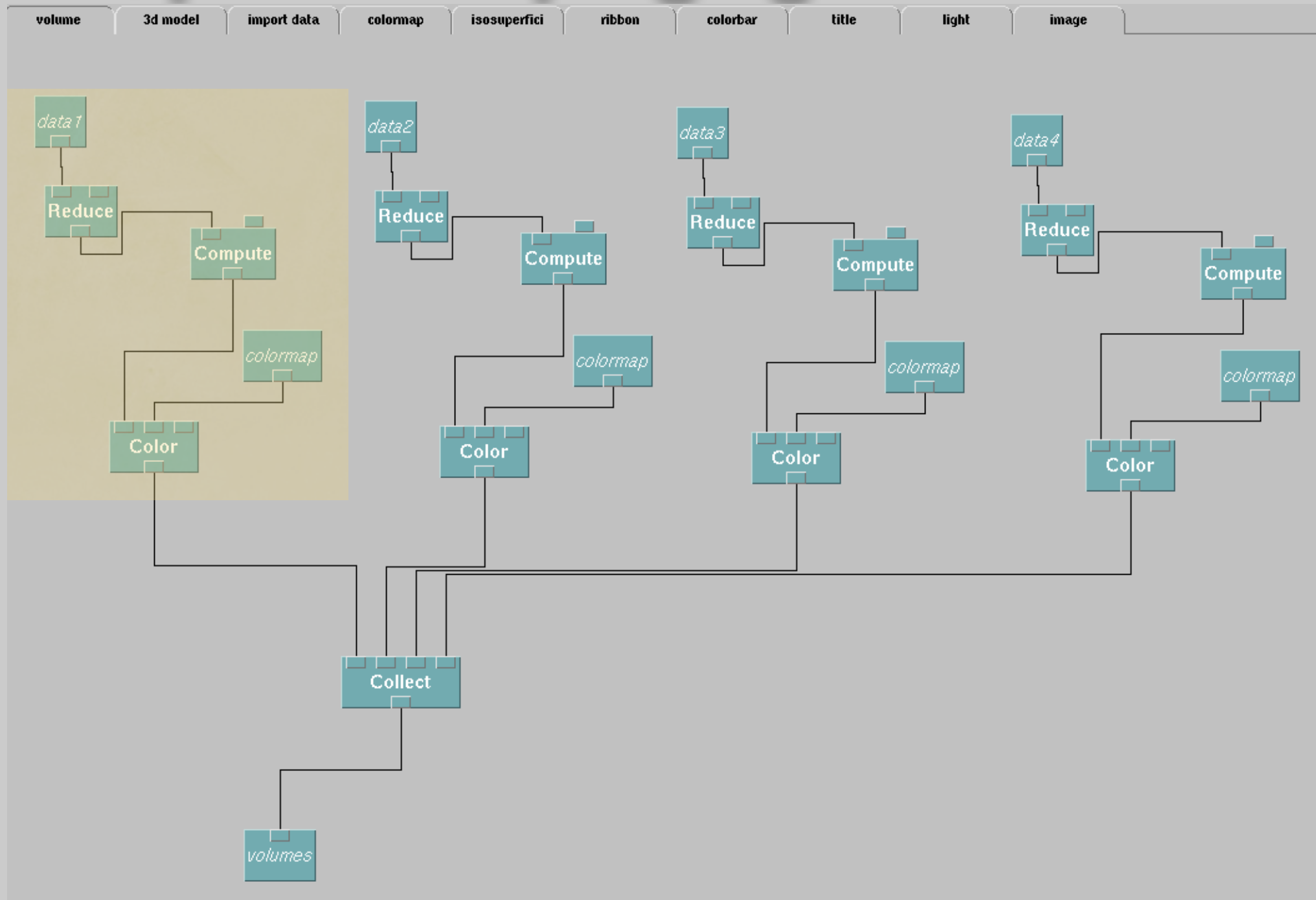
OpenDX map highlights - streamlines



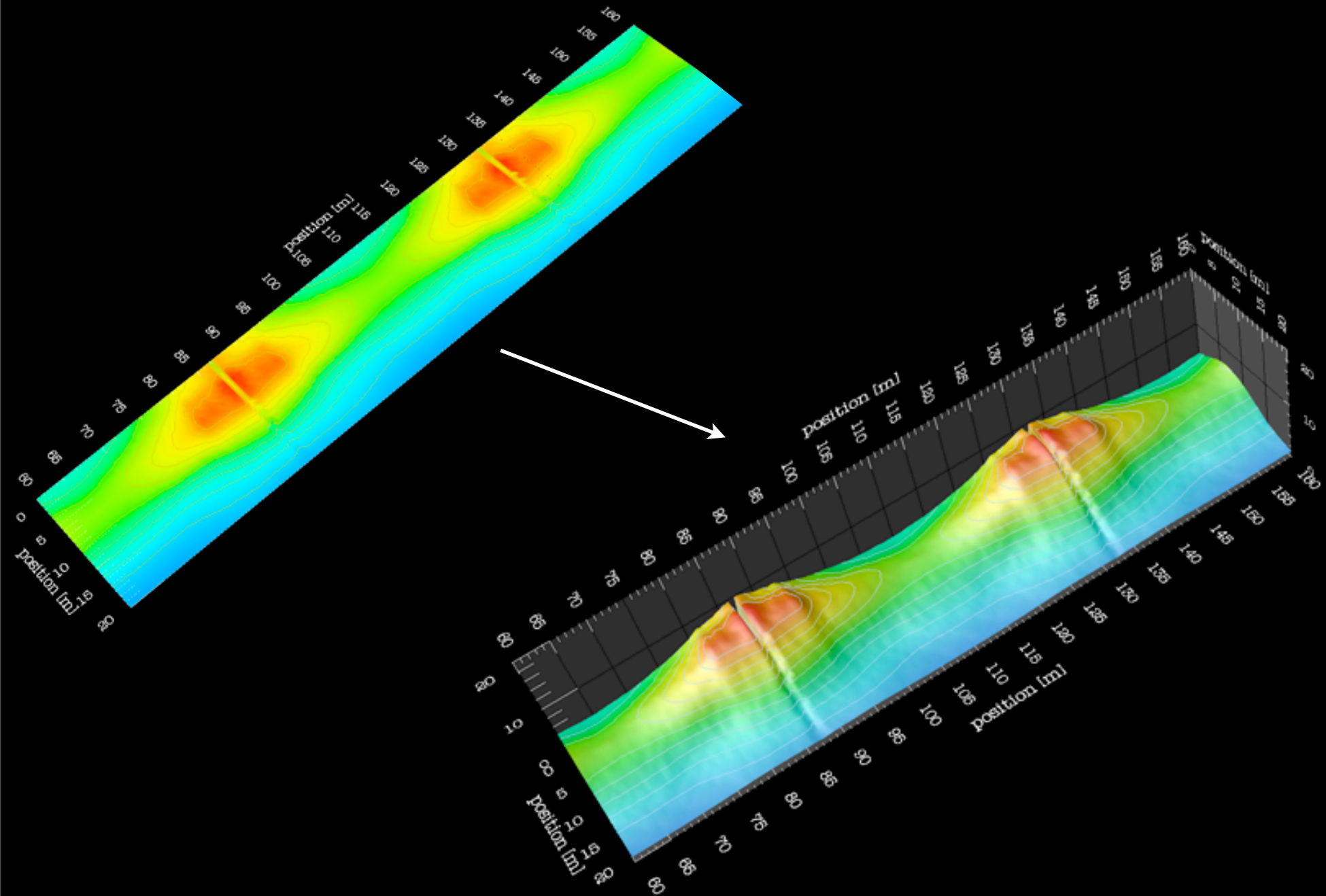
OpenDX map highlights - isosurfaces



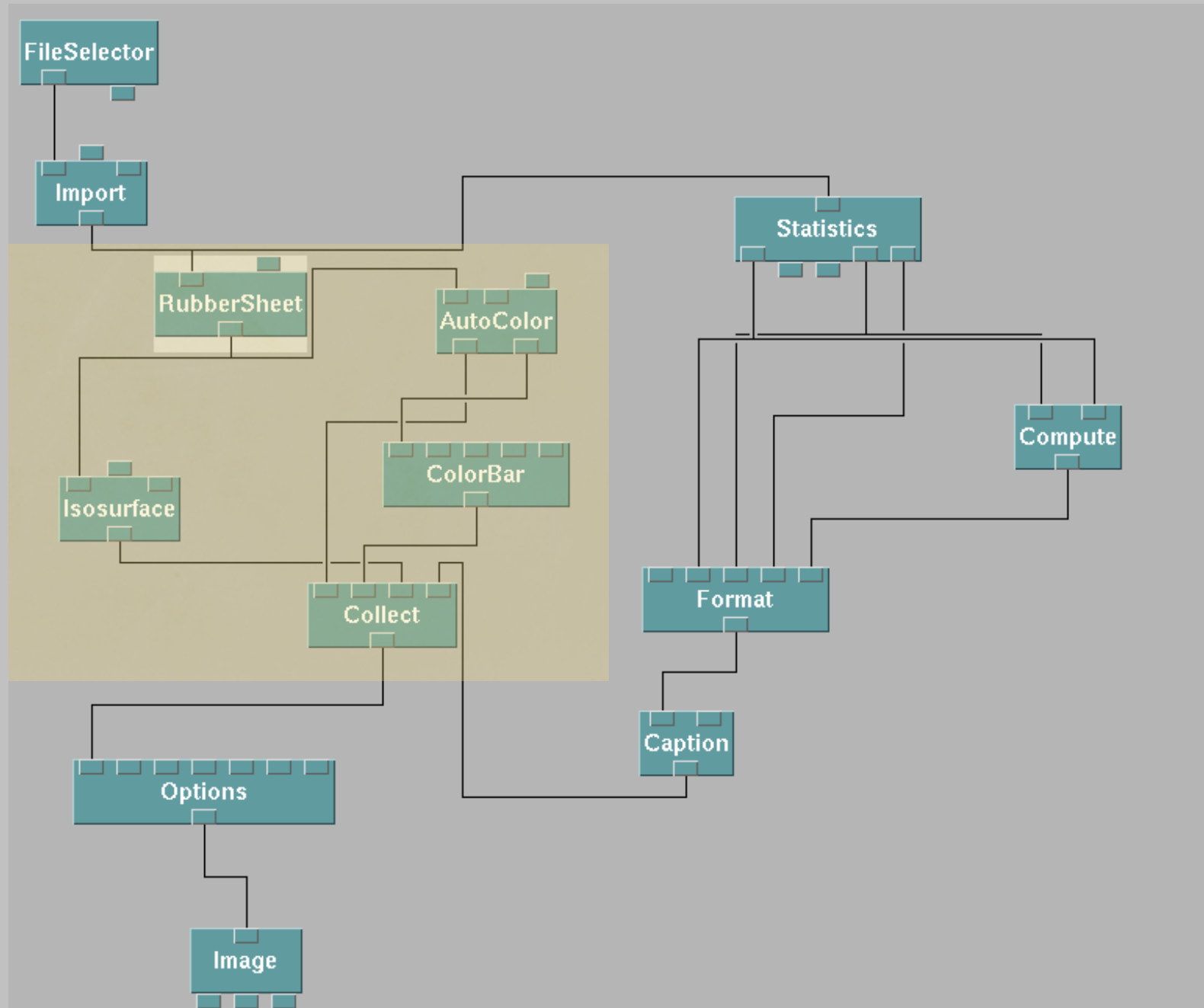
OpenDX map highlights - volume



Grid visualisation enhancement



OpenDX map highlights - 'rubber sheet'



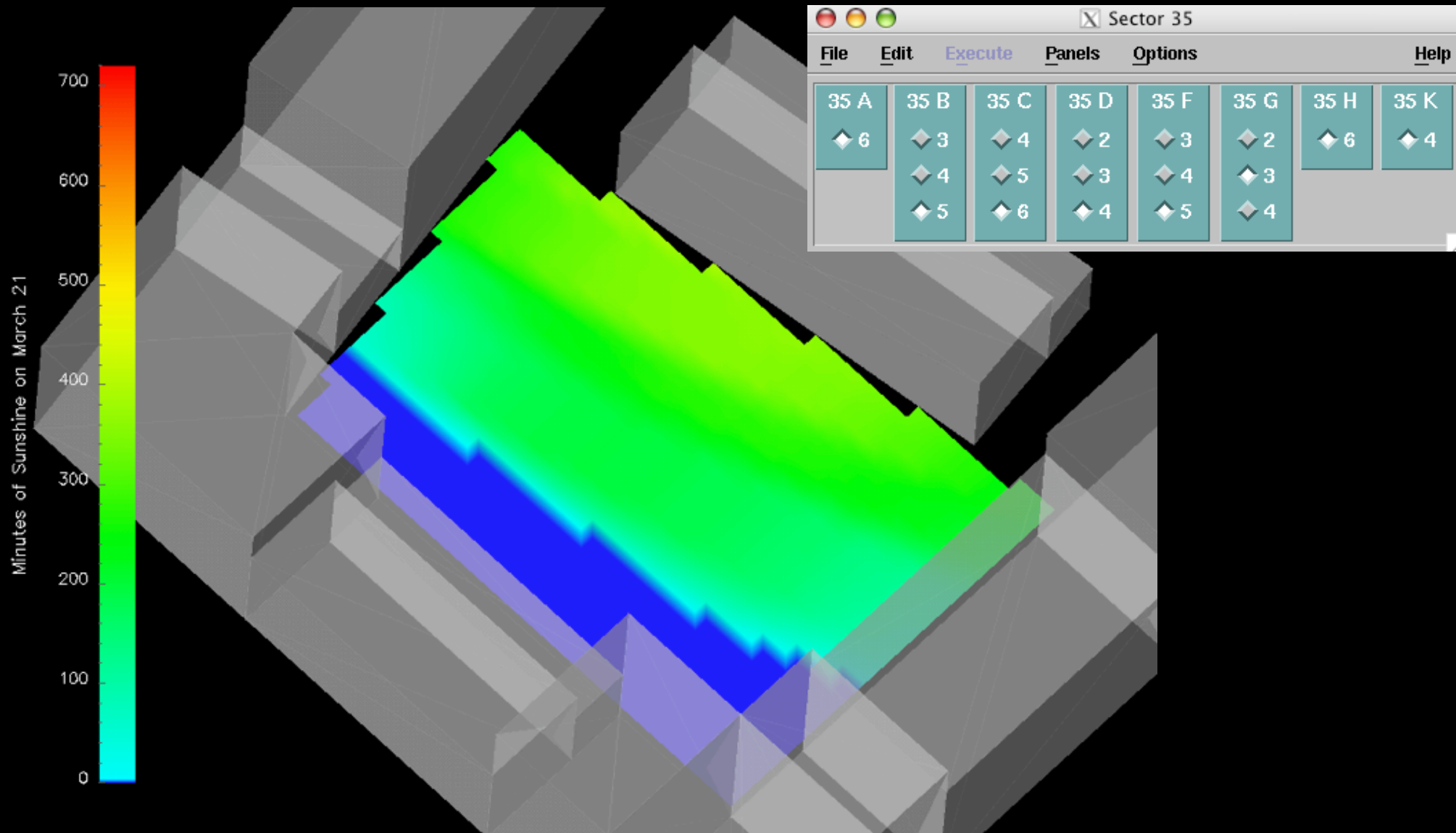
Daylight and data visualization

Variabilità dell'illuminamento nell'arco della giornata

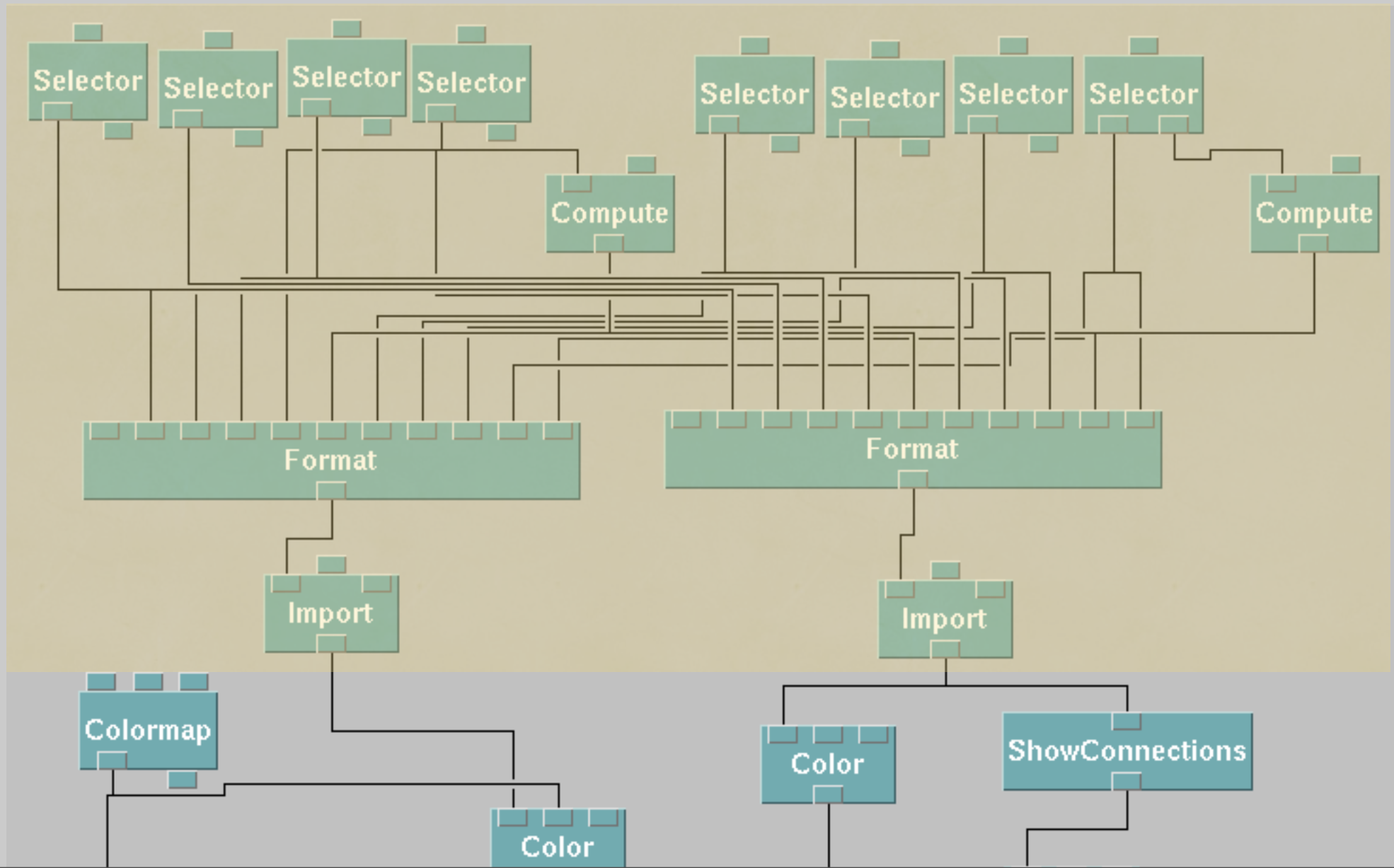
Vettore gradiente di illuminamento

Stapolin Village

Experimenting with interactivity



OpenDX map highlights - Selector



End Part I

Generating Options Using String Analysis

using strings to define and control parametric analysis

String / Tuple

- An arrangement of not necessarily distinct symbols
- Order matters
- Not a permutation, combination or set.

String / Tuple Example

- Given the alphabet $\{1,2,w\}$
- There are nine strings of length 2

11	12	1w
21	22	2w
w1	w2	ww

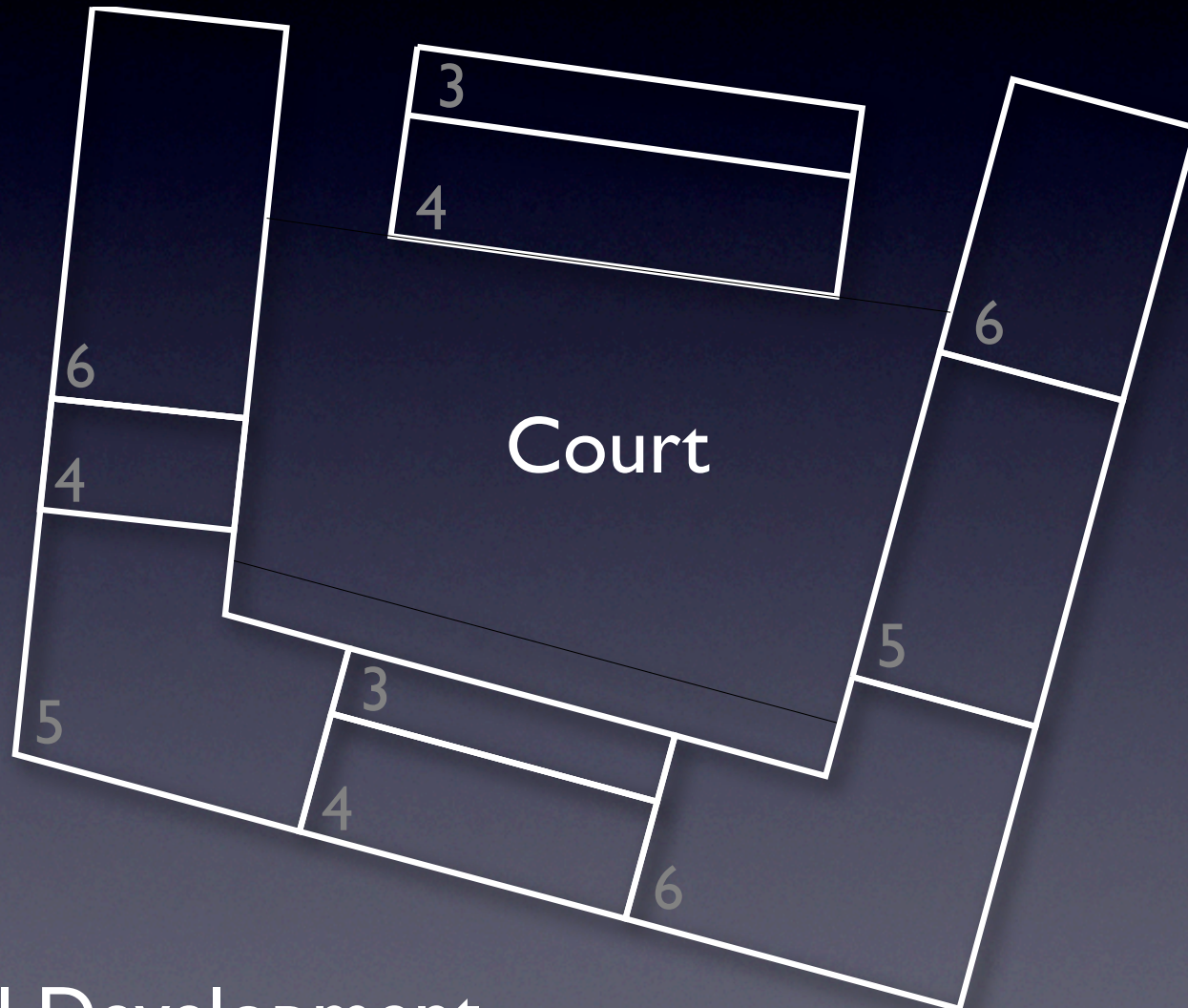
String → Parameters

- Each string element defines a model parameter.
 - building size
 - orientation
 - facade element

Using Strings

- Strings defining all possible cases are generated.
- Number of strings are reduced by imposing constraints.
- Remaining strings are written to a control file.

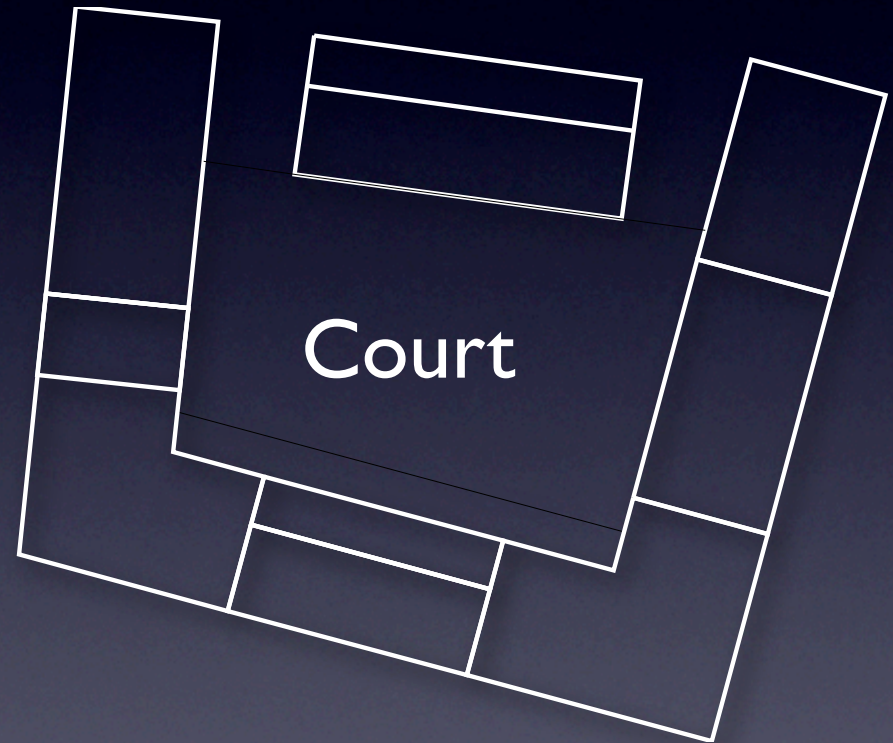
Case Study - Massing



Site Plan
Residential Development

Courtyard Sunshine Requirement

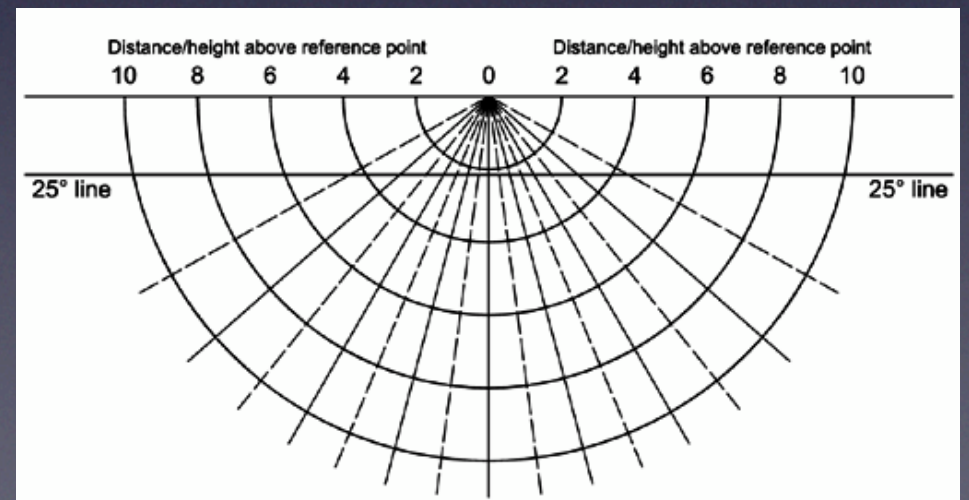
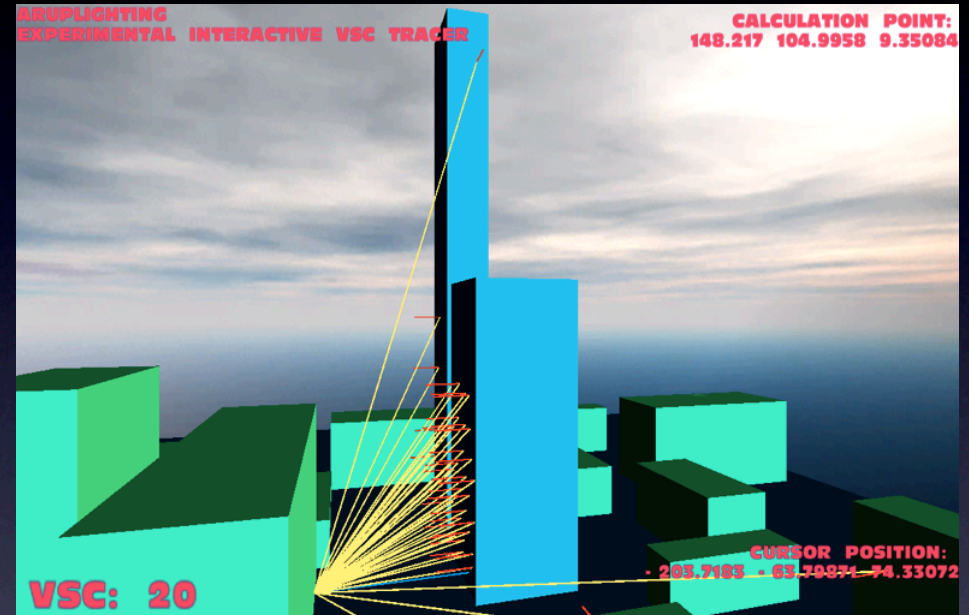
- 75% of court must receive direct sun on Mar 21.
- In the initial scheme only 70% of court receives direct sunshine.



Development Site Plan

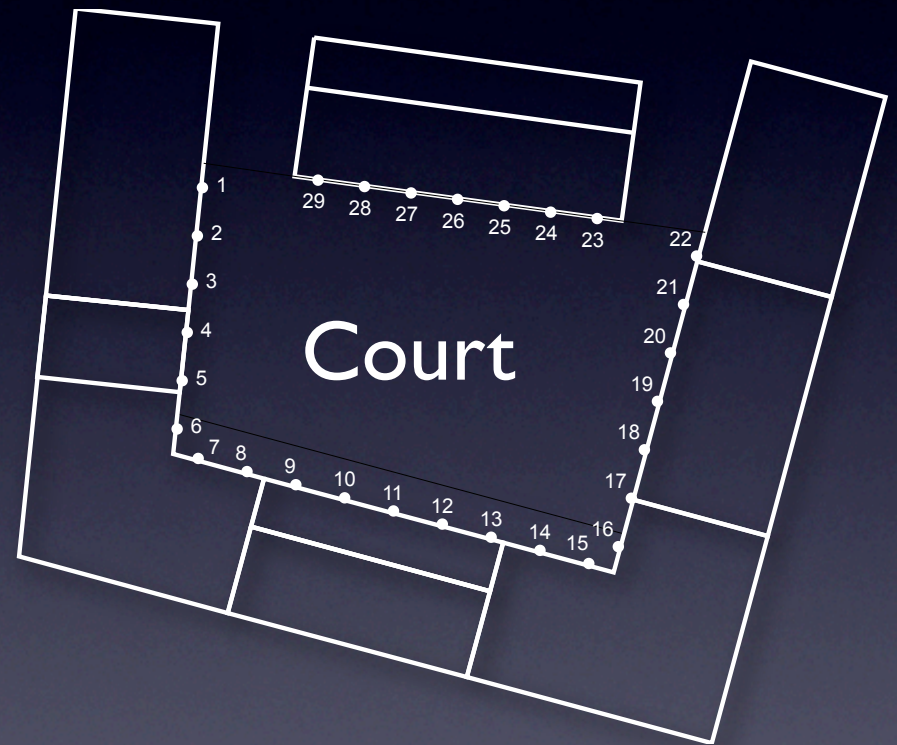
Vertical Skylight Component

- Minimum skylight requirement for windows.



Vertical Skylight Component

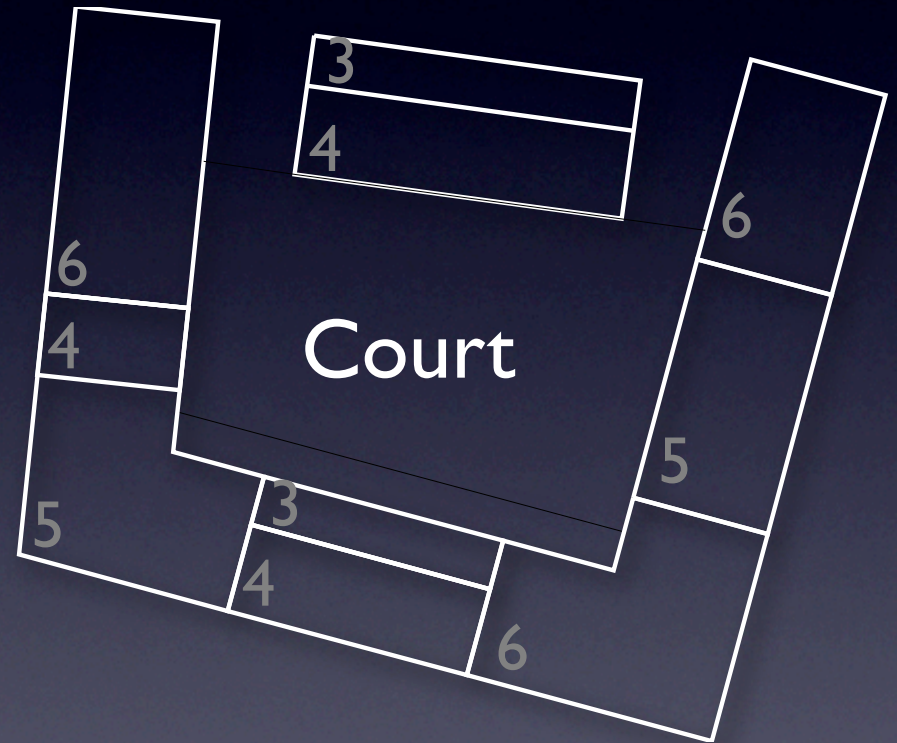
- In the initial scheme only 18 of 25 windows passed.



Development Site Plan

Finding a Solution

We were asked to develop a massing model that meets planning requirements.



Development Site Plan

Defining a String

Block heights in Stories:
{ A, B, C, D, E, F, G, H, J, K }

A = {0 ... 6}

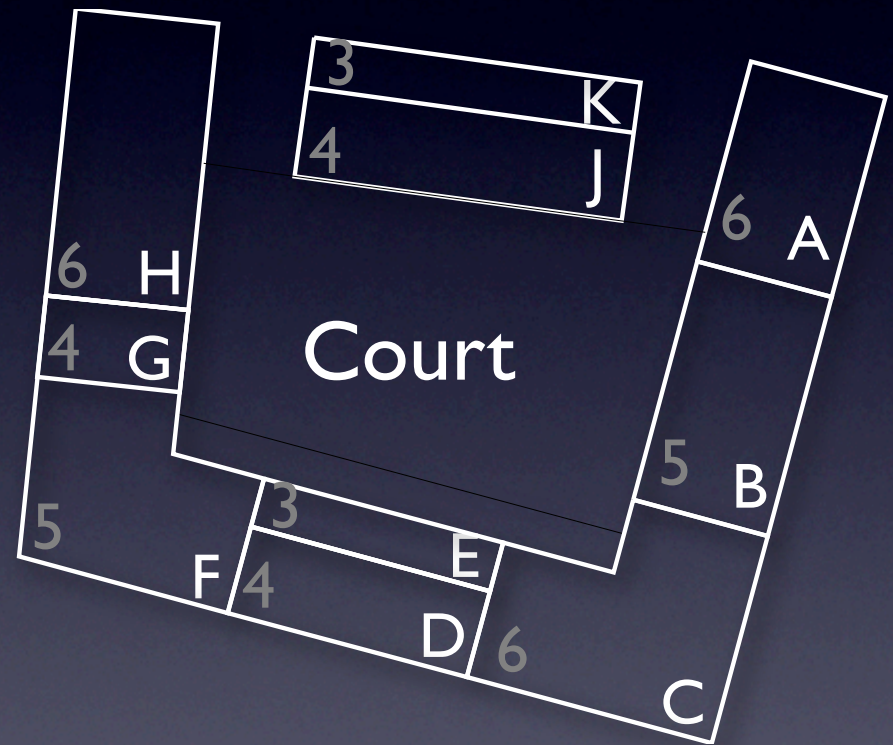
B = {0 ... 5}

C = {0 ... 6}

D = {0 ... 4}

etc.

24,696,000 tuples!



Development Site Plan

Generating Strings

genstring

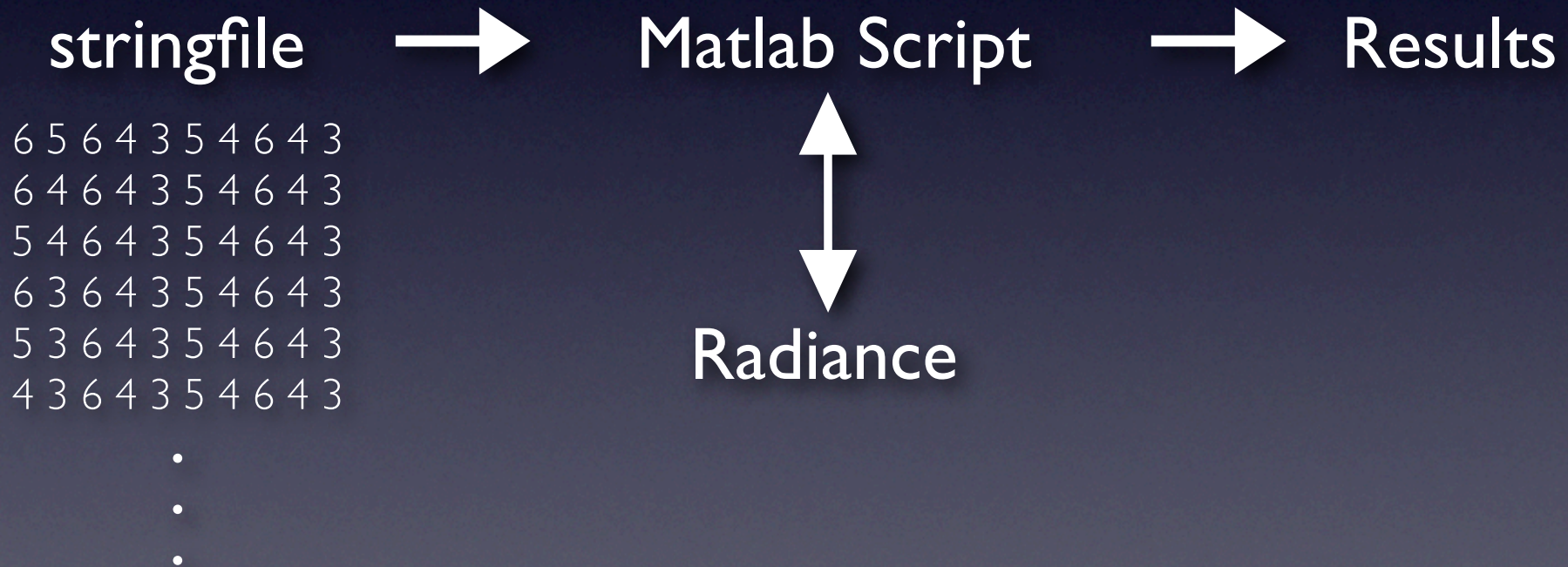
-n num	<i>number of elements</i>
-l l1 l2 l3 ... ln	<i>lower value for each element</i>
-u u1 u2 u3 ... un	<i>upper value for each element</i>
-a a1 a2 a3 ... an	<i>area (weight) for each element</i>
-t val	<i>area limit (constraint)</i>

genstring ... | **awk** '{more constraints}' > stringfile

Reducing Test Cases

24,696,000	Total possibilities
1,543,500	Coupling E&D and K&J
24,000	No less than 2 floors per block
4,506	No less than 80% floor area
695	Corner blocks taller than adjacent.

Workflow



Results

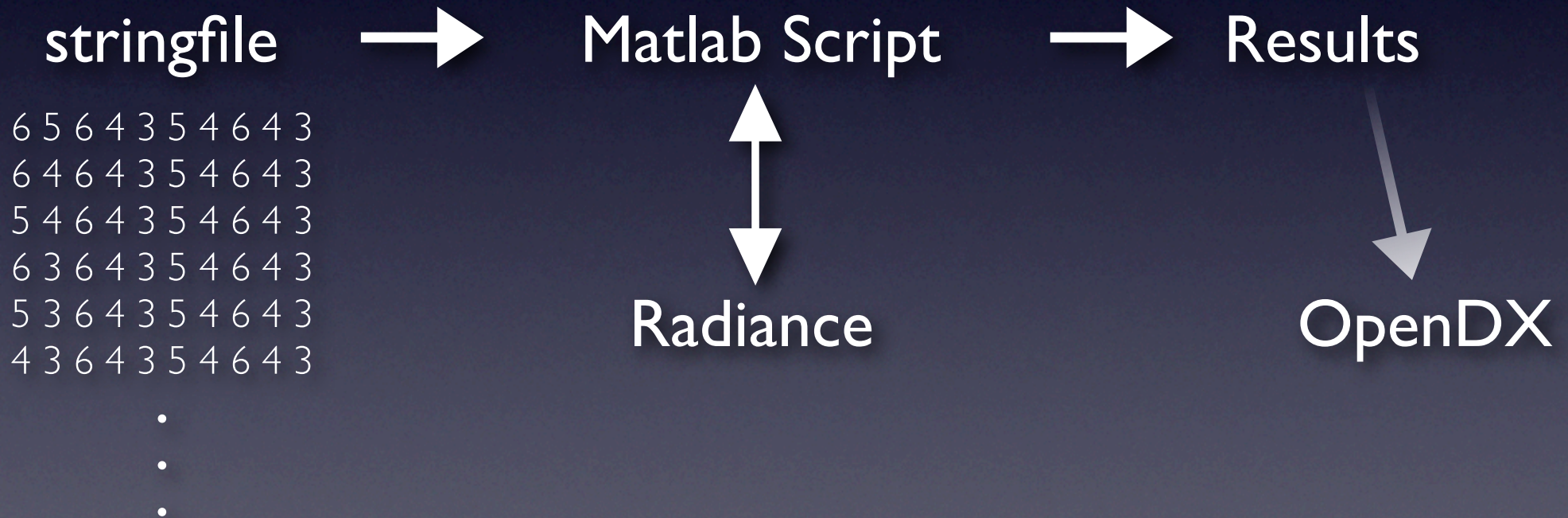
All Results

Blocks										Floor Area	Points passing BRE skylight	Percent no sunshine
A	B	C	D	E	F	G	H	J	K			
4	3	6	4	3	5	3	4	4	3	10086	24	29.8%
5	4	6	3	2	5	2	5	4	3	10084	21	16.9%
5	4	6	3	2	5	3	5	3	2	10083	22	16.9%
5	4	5	3	2	4	3	6	4	3	10081	22	16.3%
4	3	5	4	3	5	2	6	3	2	10076	23	28.7%
6	4	6	3	2	5	3	4	4	3	10074	23	16.9%
6	2	5	3	2	5	2	6	4	3	10072	23	16.3%
6	2	5	3	2	5	3	6	3	2	10071	23	16.3%
5	3	5	4	3	5	2	5	4	3	10067	23	28.7%
5	3	5	4	3	5	3	5	3	2	10066	23	28.7%
6	4	5	3	2	5	2	6	3	2	10064	23	16.3%
5	3	6	3	2	4	2	6	4	3	10060	22	16.9%
5	3	6	3	2	4	3	6	3	2	10059	25	16.9%
5	2	4	3	2	5	4	6	4	3	10058	24	15.5%
6	3	5	4	3	5	3	4	4	3	10057	25	28.7%
6	3	6	3	2	4	3	5	4	3	10050	25	16.9%
4	3	6	3	2	5	3	5	4	3	10049	23	16.9%
4	3	6	3	2	5	4	5	3	2	10048	23	16.9%
5	2	6	4	3	5	2	5	3	2	10045	23	27.4%
6	5	6	3	2	4	2	5	4	3	10043	23	16.9%
6	5	6	3	2	4	3	5	3	2	10042	23	16.9%
3	3	5	4	3	5	3	6	4	3	10041	23	33.3%

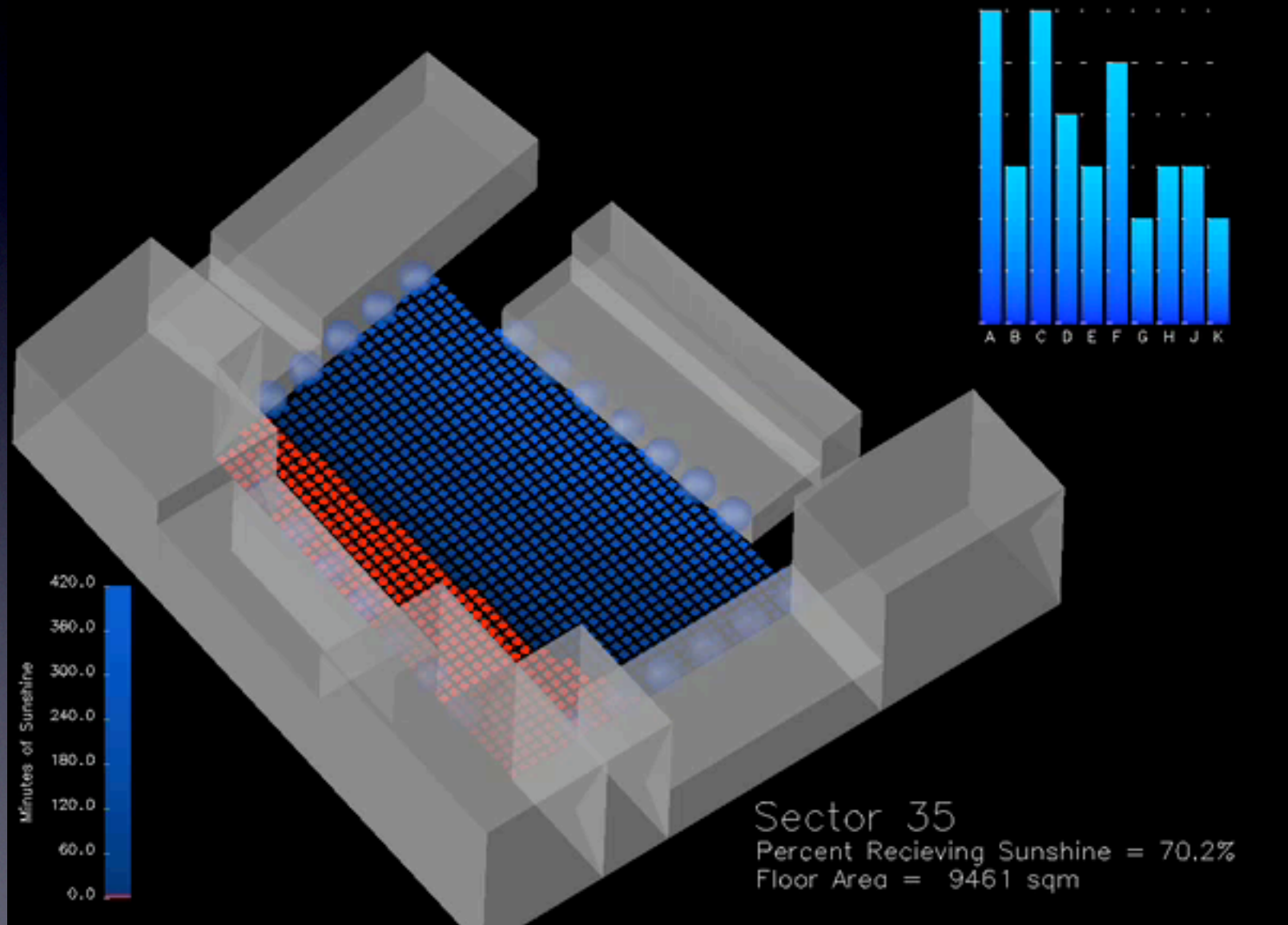
Filtered Results

Blocks										Floor Area	Points passing BRE skylight	Percent no sunshine
A	B	C	D	E	F	G	H	J	K			
6	3	6	3	2	4	3	6	3	2	10228	25	16.9%
6	2	6	3	2	4	3	6	3	2	10101	25	16.9%
5	3	6	3	2	4	3	6	3	2	10059	25	16.9%
6	3	6	3	2	4	3	5	4	3	10050	25	16.9%
6	4	5	3	2	4	3	6	3	2	9988	25	16.3%
6	3	6	3	2	4	2	6	3	2	9967	25	16.9%
6	3	6	3	2	5	3	4	4	3	9947	25	16.9%
5	2	6	3	2	4	3	6	3	2	9932	25	16.9%
6	2	6	3	2	4	3	5	4	3	9923	25	16.9%
6	4	6	3	2	4	3	5	3	2	9915	25	16.9%
4	3	6	3	2	4	3	6	3	2	9890	25	16.9%
5	3	6	3	2	4	3	5	4	3	9881	25	16.9%
6	3	5	3	2	4	3	6	3	2	9861	25	16.3%
6	2	6	3	2	4	2	6	3	2	9840	25	16.9%
6	2	6	3	2	5	3	4	4	3	9820	25	16.9%
5	4	5	3	2	4	3	6	3	2	9819	25	16.3%
5	3	6	3	2	4	2	6	3	2	9798	25	16.9%
6	3	6	3	2	4	2	5	4	3	9789	25	16.9%
6	3	6	3	2	4	3	5	3	2	9788	25	16.9%
5	3	6	3	2	5	3	4	4	3	9778	25	16.9%
4	2	6	3	2	4	3	6	3	2	9763	25	16.9%
5	2	6	3	2	4	3	5	4	3	9754	25	16.9%

Workflow



Results



Advantages of String Analysis

- Produces several viable options based on any number of criteria
- Number of test cases can be reduced systematically
- Control file can be easily split among multiple processors/machines
- Find optimal scheme with certainty

Disadvantages of String Analysis

- Radiance calculations need to be brief.
- The number of possible strings can sometimes be too great.
- Other algorithms can find the optimal scheme with fewer iterations.

Thanks!