ole kristensen
software artist

degnestræde 36
2670 greve

ole@kristensen.name
http://ole.kristensen.name

+4560834679

29082014

Software sketches for experiencing LED light

Lighting aesthetics has become a question, not only of designing physical objects,
but of composing the temporal dynamics of clusters of light emitters;
what controllers do and how.

How should artificial light behave aesthetically over time and towards humans? When does the lighting remain in the background, move into the foreground, start pointing to it self, and when has it become a dominant noise?

Our starting point is our experience of daylight. Daylight itself fluctuates dramatically, yet remains inconspicuous to the humans that inhabit it. We 'background' daylight fluctuations, yet remain conscious about the cues that daylight possesses. The time of day, the passing of clouds, the seasons and our changing weather conditions. Our daily backgrounding of daylight fluctuations can inform how we formulate dynamic fluctuations of interior LED light with an aesthetic awareness of 'backgrounding' and 'foregrounding'.

Here we present a series of simple compositional principles and a series of physical setups controlled by software sketches that tries to embed design criteria of dynamic interior LED lighting. We aim to unpack different lighting control strategies that respond dynamically to the fluctuations of humans and daylight over time.

The software that governs the lighting fluctuations thus becomes an instrument for experiencing when the artificial lighting foregrounds or backgrounds itself in relation to it's daylight environment. Our quest is to explore and qualify these subtleties with parametric lighting compositions. The software sketches should allow us to juxtapose extremes and work our way into a nuanced understanding of the compositional aesthetic dynamics of generative and interactive artificial LED lighting.

Intensity and color temperature

Before we look at the software let's first unpack the fluctuations of an individual light emitter by it self. In our case it has two variables, intensity and color temperature. These two variables can fluctuate completely independently, as our LED lights decouple the relationship between color temperature and intensity that was interlocked in incandescent bulbs. The variables form a two-dimensional plane of possible light outputs. Any point on this plane can describe the current state of a single light emitter. The lightness could be experienced as bright, dim, off, blinding etc. The color temperature could be warm, cold etc.

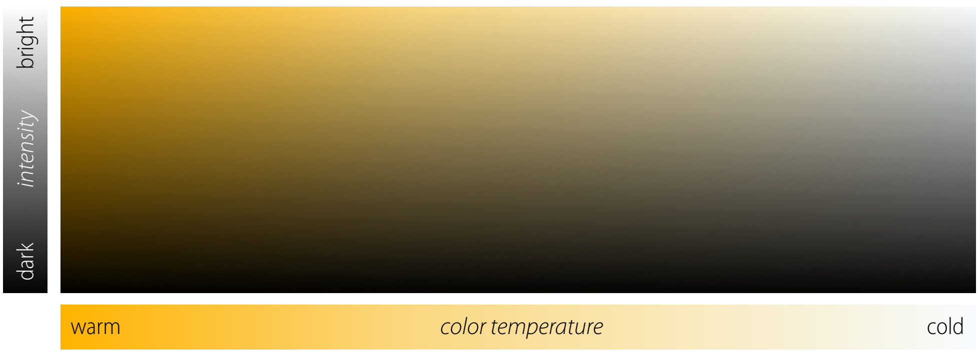

Figure 1: The two dimensions of an individual light emitter, intensity and color temperature.

Time

Fluctuations happen over time, and we are interested how fluctuations affect our experience of artificial light. The fluctuations are our temporal path through intensity and color temperature. How does the experience of the fluctuations change with their speed, and how does the relative speed of light fluctuations contribute to the foregrounding or backgrounding of the light source(s) as artificial light.

If color temperature and intensity describes the 'what' of our light emitter, the fluctuations describe the 'how'. Fluctuations can have temporal qualities such as repetition, rhythm, syncopation, flicker, etc. We would like our light compositions to potentially exhibit all of these complex qualities, without having to expose a plethora of parameters and options.

With LED lighting any change in intensity or color temperature can happen discretely or continuously, i.e. at an instant or gradually over time.

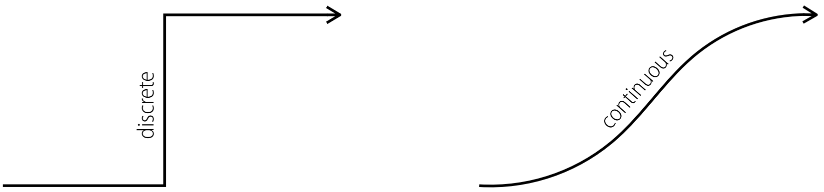

Figure 2: Discrete and continuous change seen over time

However, we are mostly interested in exploring artificial lighting fluctuations that are reminiscent of natural phenomena. As nothing moves physically in zero time, we want the fluctuations to appear continuous. We look for a simple function that will allow us to generate fluctuations that are continuous at low frequencies and appear unpredictable yet subtle. The function should have one variable 'time', and the user should be able of controlling the speed of this time.

The ideal candidate is Perlin-noise. This is a form of computer generated pseudorandom coherent noise (libnoise.sourceforge.net 2005) that has proven useful for procedural generation of seemingly natural structures. Ken Perlin invented Perlin noise in ~1983. Because his invention is used everywhere in the special-effects industry, he won an Academy Award® with the following reason:

*The development of Perlin Noise has allowed computer graphics artists to **better represent the complexity of natural phenomena** in visual effects for the motion picture industry.* (Perlin, n.d.)

A one-dimensional render of perlin noise could look like this:

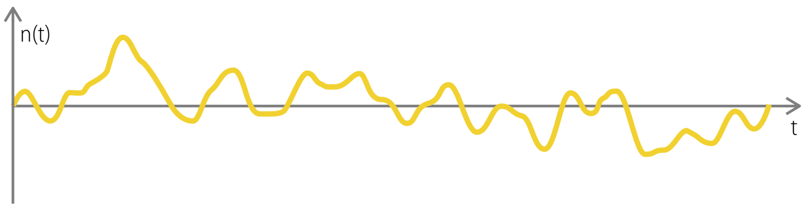

Figure 3: An example of one dimensional perlin noise seen over time

When interpreted as light fluctuations over time, the perlin noise exhibits qualities ranging from imperceptible, alive, over syncopated to noisy.

Space

A major visual component of led lighting is the possibility to individually control light emitters. Let's now look at a collective of lights. When more lights are arranged together their fluctuations collectively assume relative spatial qualities such as dense, sparse, coarse, uniform and individual. When taking part in such an arrangement a light emitter can be interpreted as a pixel.

If the coherence takes the form of figurative representation, it is usually an effect of mapping the spatial relationships of the lights to a video input. However, for now we eschew mapping filmed video content to our lighting system in an effort to focus our experience on spatial and temporal qualities that lies beneath concretely representational uses of lights.

We then need to generate signals that gives our lights a spatial relationship. For this abstract spatial reference we can again look to perlin noise and note that it can have more than one dimension.

Let's first look at a one dimensional perlin noise this time in a three-dimensional space where time moves sideways, again we see how the value fluctuates continuously over time, and we can appreciate that there are no apparent sudden changes in how the light intensity is modulated over time:

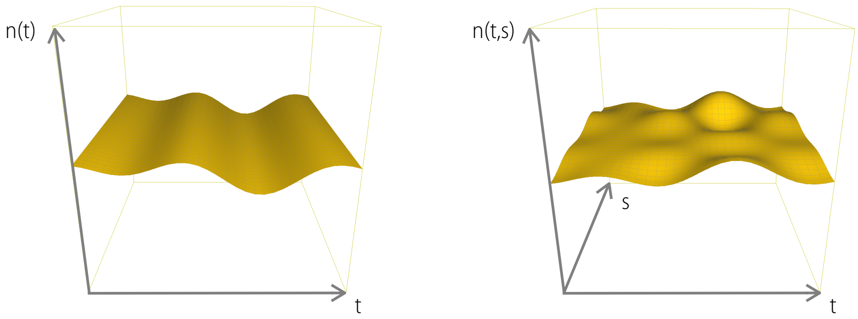

Figure 4: Comparison between one dimensional and two-dimensional perlin seen in three dimensions

By adding a dimension to the perlin noise it will describe a two-dimensional plane, that in our case can be used to modulate a spatial distribution of individual intensities of a lighting cluster over time, thus keeping a coherent and continuous relationship across fixtures.

To animate this over time we add one more dimension; time. We let the user define the speed of time and end up with the possibility of clustering lights in two dimensions over an animated plane to compose spatially coherent relationships:

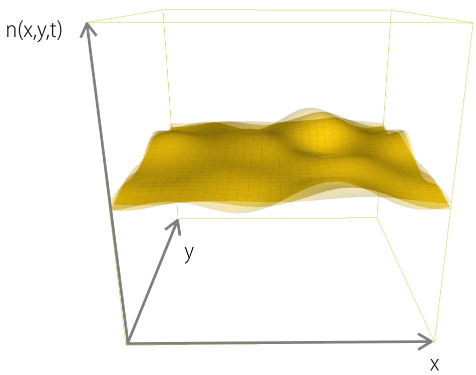

Figure 5: Animated two-dimensional perlin noise plane

With this added dimension we can construct fluctuations in space and time. This opens for an experience of spatial relatedness of multiple light fluctuations. With this we can synthesise collective relational qualities such as individuality, sameness, closeness, resemblance, mimicry, directionality and movement.

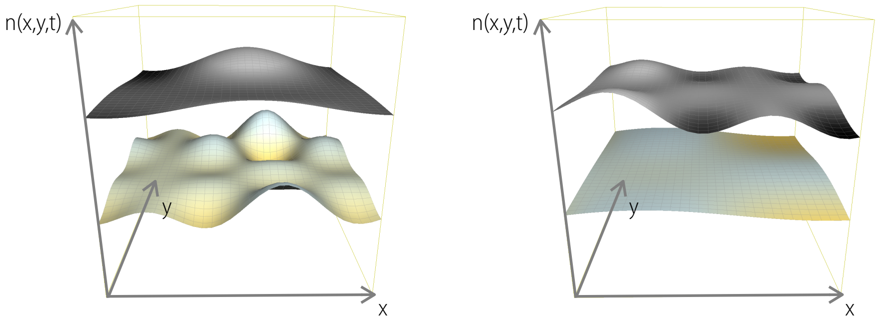

Figure 6: Examples of states of animated perlin noise in two distinct planes; intensity and temperature

We create two such fields, one for intensity and one for color temperature. Now the lights can sample their values from a point in two animated perlin noise fields, where the timing and dynamics of intensities and color temperature can have each their own temporal quality.

When a cluster of lights sample from positions in the field that correspond to their physical arrangement the weather can be scaled to form compositions of coherent fluctuations in the brightness and or color temperature of the physical lights.

To represent this in two dimensions the light temperature values are multiplied with the intensity values, and we get an image like this:

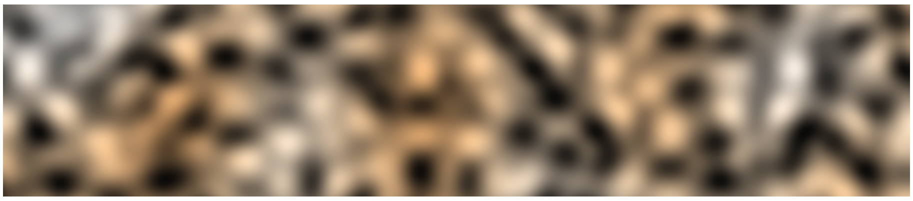

Figure 7: A composite of the two planes of perlin noise each at a given frequency, 'spread' or zoom-level.

Pseudo-code

```
i_t += i_speed * 60 / fps();
t_t += t_speed * 60 / fps();

for(int x = 0; x < width; x++){
     for(int y = 0; y < height; y++){

            intensities[x, y] =
                     map( noise(x*i_spread ,y*i_spread ,z=i_t), -1.0, 1.0, i_range_begin, i_range_end )

            temperatures[x, y] =
                     map( noise(x*t_spread ,y*t_spread ,z=t_t), -1.0, 1.0, t_range_begin, t_range_end )
     }
}
```

Materials

The software is programmed in c++ using openFrameworks (openframeworks.cc, n.d.), runs on Ubuntu Linux 14.04 LTS and controls the lights over DMX via an art-net node.

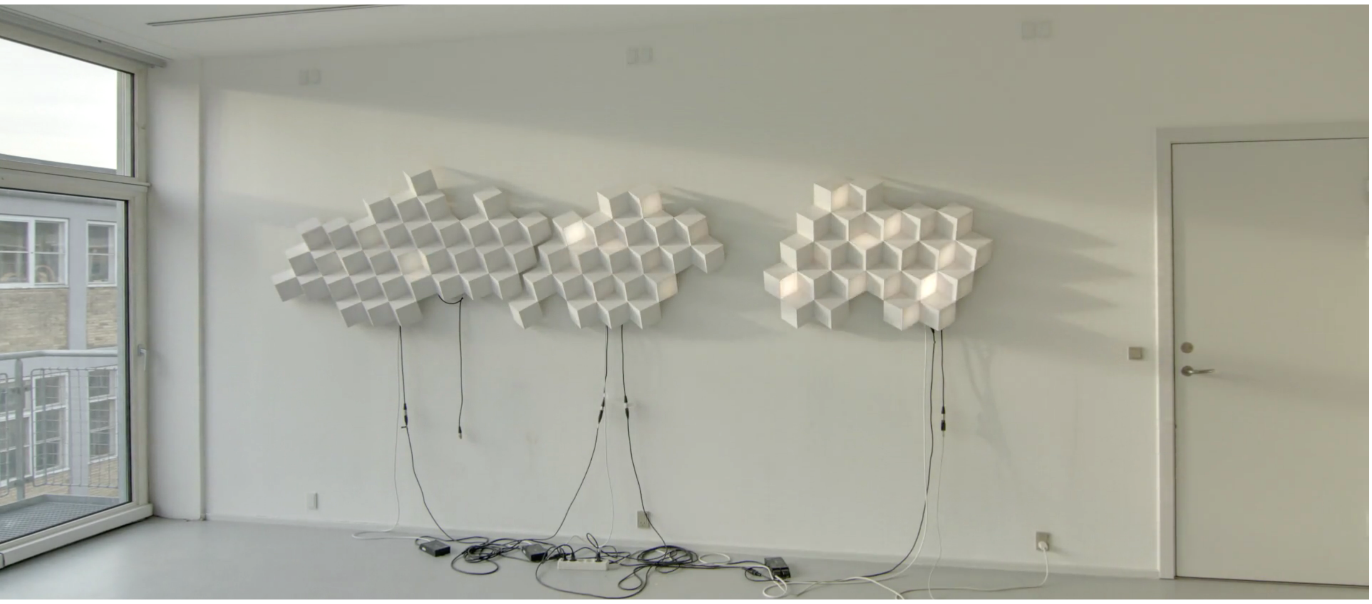

Figure 8: The tessellation setup mounted in the Light Lab of KADK.

The Tessellation

The first physical setup is constructed of quadratic surfaces arranged to reflect incoming daylight, have a clear readability of shadows and light, and enable compositions of artificial light from a subset of surfaces with integrated LEDs.

The aim was to have a structure for experiencing compositions of fluctuating artificial light integrated with natural variations of daylight.

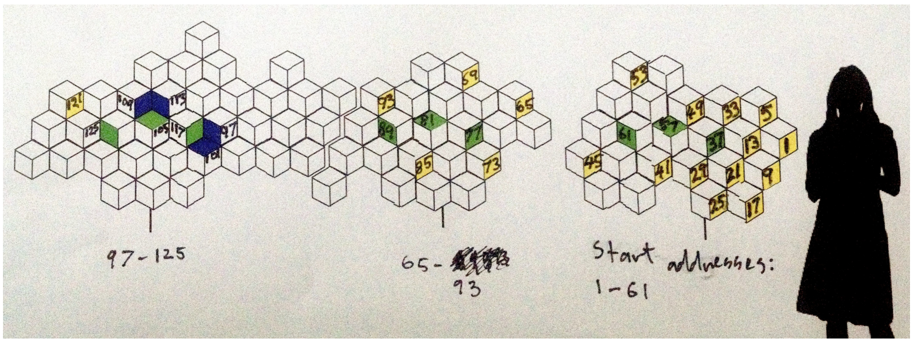

Figure 9: Map of light emitting surfaces in the tessellation with annotations of DMX start addresses

Tesselation hardware

Each light emitting surface is a piece of frosted acrylic plastic outfitted with a reflector and a piece of LED strip with high CRI warm white and cold white LEDs. The embedded led strips were driven from eldoLed drivers that are capable of operating with a bit depth of 16, giving a fine-grained control of intensities with more than 65.000 digital steps, effectively making fine and slow intensity shifts almost imperceptible. With the usual bit depth of 8, we would have experienced discrete digital quantization in slow intensity fluctuations, especially in the lower ten percent range, were a range of 256 steps becomes very visible.

Tesselation Software

With an outset in the principles of using generatively animated two dimensional perlin noise as a source for color temperature and intensity, a control software is made that enables the designers to generatively synthesize, study and describe temporal and spatial qualities of fluctuating light compositions.

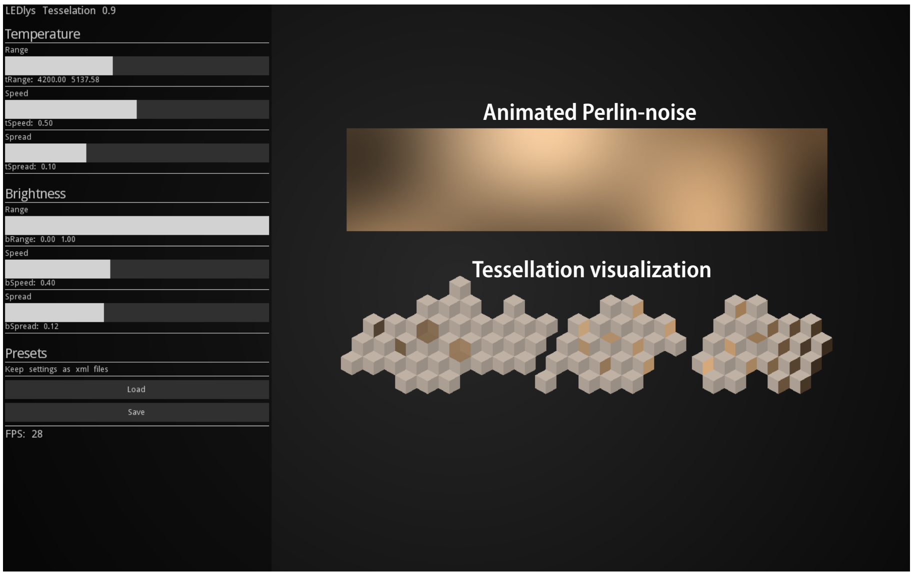

Figure 10: Screenshot of LEDlys Tessellation software with annotations

The software allows for control of the two perlin noise-fields for temperature and brightness. Lets look at the screenshot above (figure 10). The animated perlin-noise in the top of the screen is a composite of two perlin clouds, one for color temperature and one for brightness. This perlin noise is sampled and mapped onto the individual light emitting surfaces that are shown in the Tessellation visualization. The mapping retains the spatial relationship from the image to the arrangement of lights.

To the left are the controls. There are two similar-looking sections for Temperature and Brightness and a bottom section for loading and saving presets.

The two sections for Temperature and Brightness each have three parameters that control how their perlin noise behaves.

The first parameter is manipulated with a range slider, that sets two values; the minimum and maximum for the fluctuation. This means the white portion of the slider can be dragged sideways in both ends, effectively contracting or expanding the possible range of intensities of temperatures. Note the minimum range from the Brightness section seen in figure 8. It is raised to 45%, meaning that no lights will go below this intensity level. The range slider is linear.

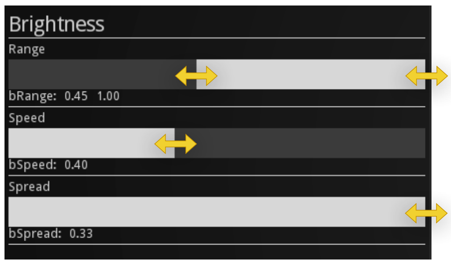

Figure 11: Three sliders governing the perlin noise.

The middle slider sets the speed of the fluctuation. This slider is cubic, prioritizing high resolution in the lower (slow) end of the scale. This is illustrated in figure 12. A cubic slider allows for finely tuning extreme slowness, an important prerequisite for composing fluctuations that are changing at an almost imperceptible pace. On the other hand the slider will still allow for extremely fast fluctuations in the higher end of the scale, allowing for comparative experiential research of the extremes. When at zero, the animation is stopped.

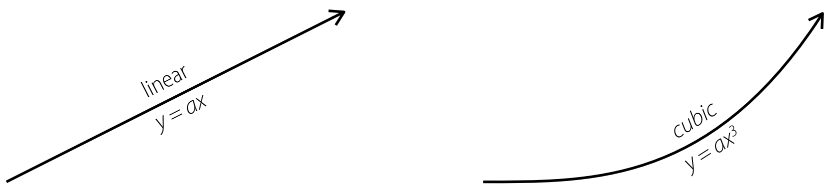

Figure 12: The cubic slope prioritizes very high precision in the lover end of a slider input.

The last slider sets the so called Spread of the cloud. This is effectively a 'zoom' slider allowing scaling of the perlin noise. This can be understood as 'how far the lights are from each other' or 'a scale between uniformity and individuality'. When at zero the spread parameter will generate a perlin noise that is 1x1 pixel, rendering a uniform value across the noise field, in effect letting the lights behave in unison. When dialed all the way up, the spread parameter will generate a perlin noise that is very fine-grained, in effect letting the lights behave totally individual without any apparent coherence.

Figure 11 shows the three sliders from the brightness controls, the temperature sliders are identical, except that the range slider covers values in kelvin degrees within a range derived from the specifications of the cold and warm LEDs.

Below are two illustrative examples of different settings of the software, figure 10a and 10b. They are helpful in understanding how the parameters affect the generated perlin noise. First note that the parameters in the Temperature section are set identically in both screenshots; fluctuations happen over the full range, and have their speed and spread sliders set to a medium position. As a consequence the temperature distribution in both screenshots are of a similar nature.

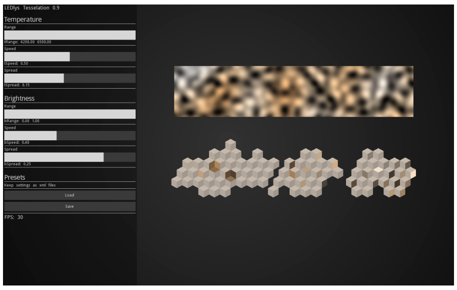

Figure 13a: One possible configuration

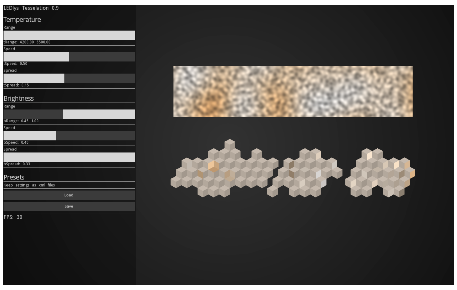

Figure 13b: After changing two brightness sliders.

However, the settings for brightness vary between the two screenshots, and first of all the spread is noticeably different, with less spread in figure 10a than 10b. However, the range of 10b is limited to 45% - 100%, thus generating intensities with less contrast as no led's are allowed to turn completely off.

Remember that this is a animated noise, generative and continuous. When formatted as light output in the tessellation it is possible to synthesize abstract lighting fluctuations that are reminiscent of the fluctuations natural daylight as reflected from water surfaces, filtered from the movement of leaves in trees or modulated by the passing of clouds. Other more extreme or 'unnatural' compositions are also possible, and allow for comparatively studying and qualifying the temporal aesthetics of dynamic artificial light interplaying with the always dynamic daylight.

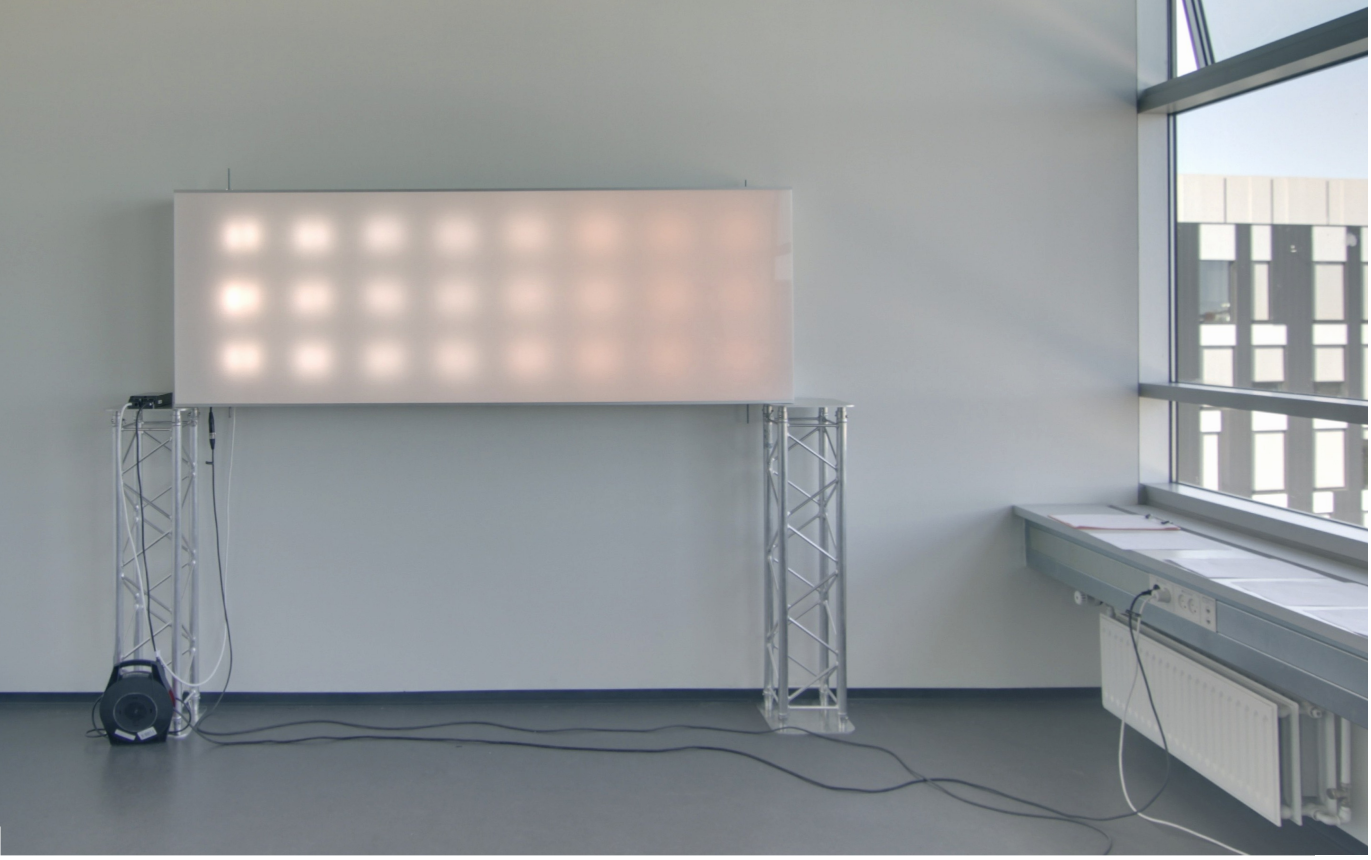

Figure 14: The Wall Box setup in an office space at the IT University of Copenhagen

The Wall Box

As a juxtaposition to the tessellation and it's clear rendering of daylight shadows, we construct a traditional flat wall mounted light box.

The aim was to have a structure for experiencing traditional diffused arrangements of light 'pixels' in low resolution.

Additionally the structure allows for an experience of sharp vs. diffuse light rendering, inspired by the works of Jim Campbell (Campbell 2001).

Wall Box hardware

The front is a uniformly opaque acrylic surface that will act as a diffuser. Inside the wall box is mounted the same type of cold and warm LEDs and driver circuitry as in the tessellation. The LEDs are mounted in a planar grid configuration of 'pixels' in a 8 x 3 on a backplane.

There is an adjustable distance between the LED backplane and the acrylic diffuser. This allows for comparing sharp and diffuse renderings of fluctuating light emitters. The backplane mounts allow for adjusting the distance between LEDs the diffuser uniformly or at a slope. Mounted at a slope the light diffusion varies horizontally along a linear gradient.

Below are three top section drawings with examples of back panel positions and descriptions of their consequences for the light diffusion.

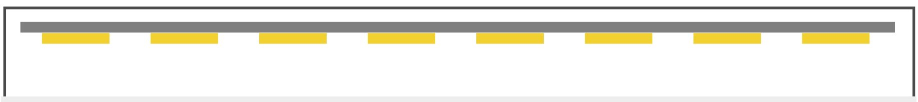

Figure 15a: Equidistantly retracted furthest from the opaque front plate, the 'pixels' are uniformly diffuse.

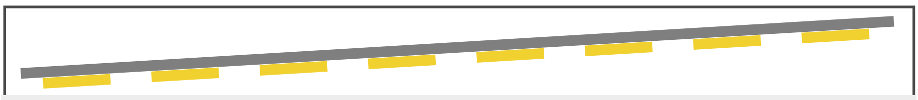

Figure 15b: Mounted at a slope, the 'pixels' are rendered with a diffusion that increases towards the right.

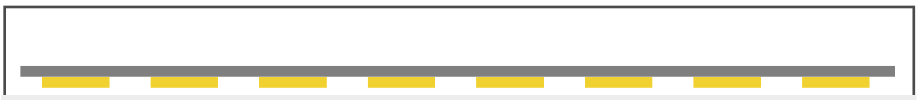

Figure 15c: Mounted close to the front, the LED 'pixels' are rendered sharper.

Wall Box software

As the main differentiation of the wall box lies in the physical design, i.e. the regular pixel grid arrangement and the adjustable diffusion, the wall box software is just a minor adjustment of the software that controlled the tessellation setup.

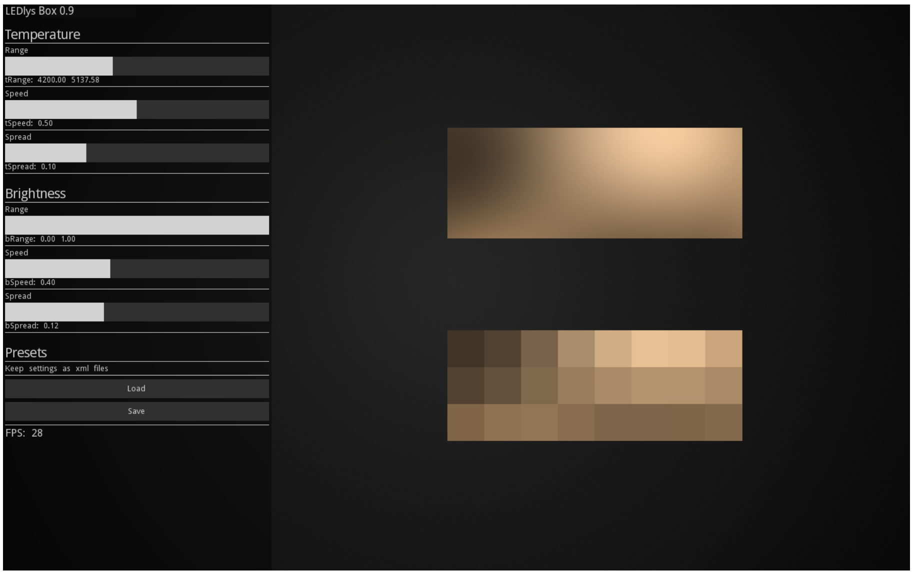

Figure 16: Screenshot of LEDlys Wall Box software

The only change is the dimensions of the perlin-noise field and the visualization of the light output. Parameters and their interface, remain the same. The perlin noise algorithm remain untouched.

By keeping the generative software dynamics constant, we can better compare the qualities of the two physical arrangements of light emitters.

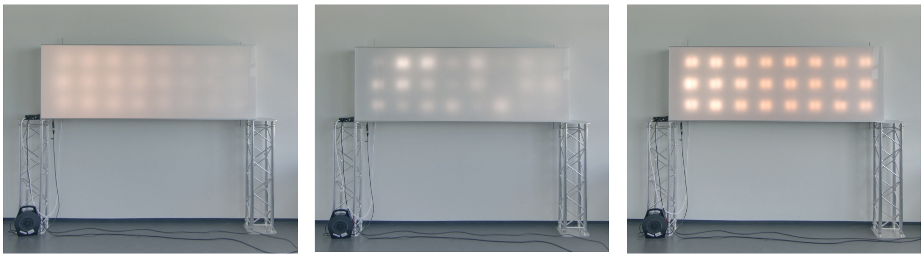

Figure 17: Three different settings of the wall box.

Figure 18: The Light Follows setup at Intermedia Lab of the IT University of Copenhagen.

Light Follows

The third setup is an investigation of interactive light compositions that are informed by tracking of a human participant. The two former setups were generative in nature and aimed to qualify temporal and spatial aspects of a dynamic lighting aesthetics. The Light Follows setup adds a third aspect, that of the involved and interacting human.

This setup consists of a room with luminescent walls and grid of ceiling mounted down lights. The lighting control system follows the position and orientation of the participant.

The aim was to have a structure for experiencing the emergent qualities of interactive compositions of lightness and darkness.

Light Follows hardware

The room is constructed in a frame of standard aluminum box truss. The floor is covered with medium grey carpet. The walls consist of translucent white molton fabric, strung on rectangular wooden frames.

Behind the walls are LupoLED Dynamic White studio lights and the ceiling contains a 3 x 4 grid of professional stage quality LED fixtures facing downwards. The LupoLED lights can also be moved from behind the walls and mounted in the corners of the ceiling, rotated to face the center of the room and tilted downwards at 45°. All lights are DMX controllable and can synthesize warm and cold white light with a high CRI.

There is no hardware for tracking human position and orientation in the space. Tracking is done reliably and elegantly by a live human operator seated outside the room. This is a simple way of enabling complex improvisations with the interactive aspect of the compositions, i.e. speed-of-thought experiments with different rulesets, latencies and behaviors without reprogramming the software or overloading the interface with options. It is then possible to experience interactions that require e.g. tracking a space between people in a crowded room, anticipative tracking of where the human will go, tracking of where the human is looking etc. by instructing the human operator to move his mouse differently.

Figure 19: The operator at the laptop with an extensive documentation setup including HDR captures.

Light Follows software

This software continues from the two other setups and retain the possibility of generating two superimposed perlin noise fields to generate intensity and color temperature fluctuations. As seen in the screenshot below the six first sliders, in the control sections Temperature and Brightness are kept. In the interest of clarity the generated perlin noise is not shown on screen, as the screen real estate is needed for a visualization of the physical arrangement of the lighting fixtures and their effect on the floor. However, the lights can get their color temperature and brightness from the same perlin noise algorithm that synthesized the light fluctuations in the two other setups. As before uniform lighting situations can be achieved by narrowing ranges down and/or zeroing spread. Static situations can be achieved by zeroing speed.

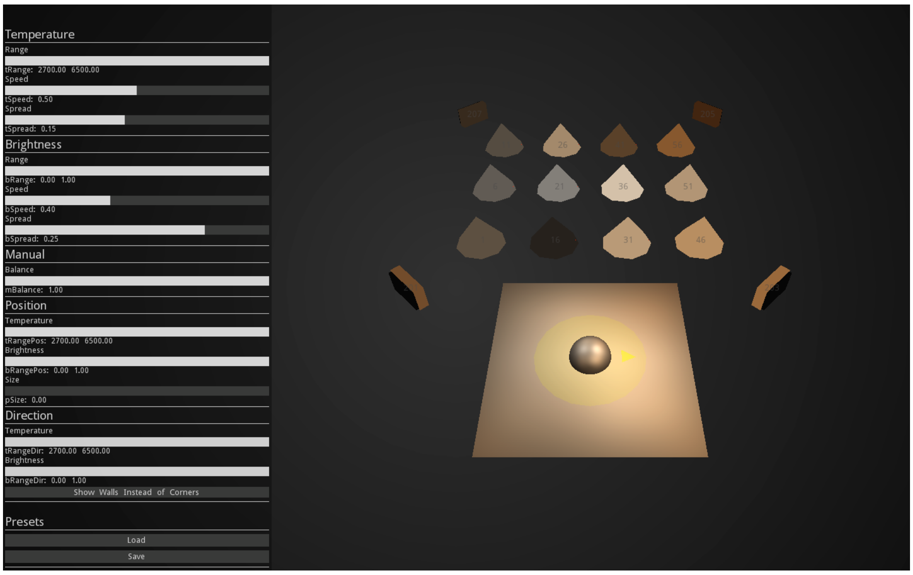

Figure 20: Light Follows software, here setup to act like the two other setups (using only perlin noise).

The room is visualized at an angle from above selected to avoid overlapping elements. The light symbols and square floor act as interactive parts of the control interface and can be manipulated directly by the computer mouse.

The round ball on the square floor visualization can be dragged to inform the system of a position. The ball will normally be used to follow the position of the human participant. Protruding from the ball is a yellow arrow. The arrow can be rotated around the ball on the floor plane to indicate a direction. The arrow will normally be used to follow the orientation of the human participant.

In the position section of the controls there are three parameters that affect how the position of the ball interacts with the light composition. The first slider is a range slider that lets the position affect the range of temperatures for lights above the ball. The second slider is a range slider that lets the position affect the range of brightness for nearby lights. The third slider adjusts size of the area that is affected by the position. Lights are affected in reverse proportion to their distance to the ball on the floor. When the size parameter is at zero the position has no influence, when it is at max, the position ranges control all lights.

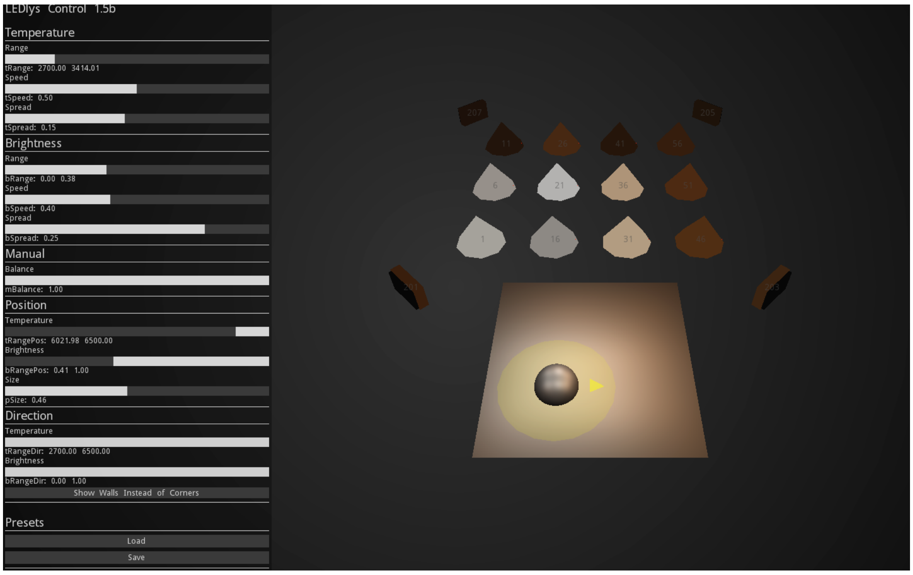

Figure 21: The position affects the ranges of the perlin noise to make nearby lights colder and brighter.

In the screenshot above the room is setup to have a warm dim light that fluctuates at a moderate speed. The position shifts the temperature of nearby lights towards a narrow range of cold colors, and the brightness into a higher range.

All lights have three possible modes. They can be generative, sensitive to direction or manually set. The generative mode is the default and covers the behaviors explained so far.

To make a light sensitive to direction, it can be right-clicked. The light is then drawn with a blue outline. When lights are sensitive to direction their ranges fade towards the values given by the two range sliders in the direction section of the controls. The direction ranges affect light that face at the position in the opposite direction of its orientation most. The desired effect is that a light can react when you look at it. In the example screenshot below the four direction sensitive lights in the corners will increase their brightness and temperature as they face the direction of the yellow arrow on the floor.

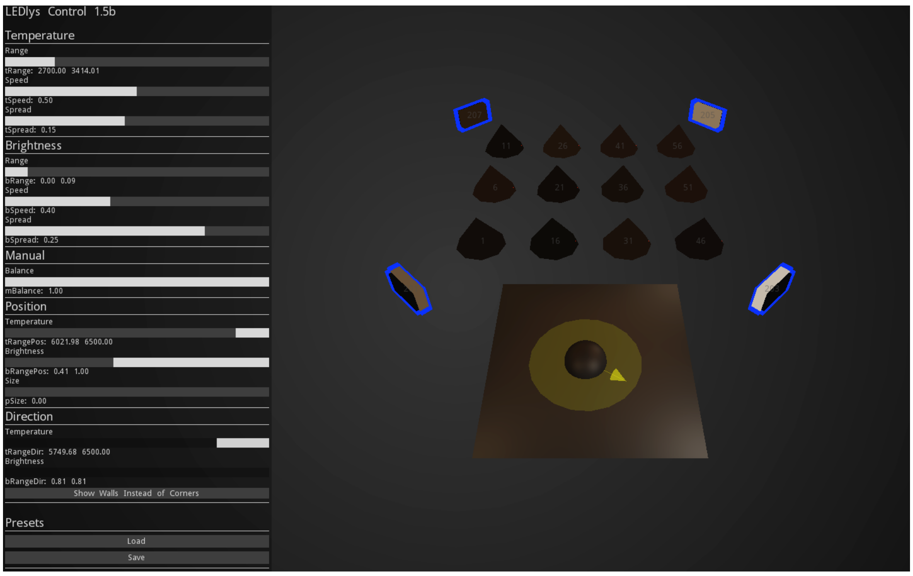

Figure 22: The four lights with blue outlines have been made sensitive to direction.

Lights can also be set to a manual mode. This is done by left-clicking and dragging on a light symbol. Manual lights are drawn with a red outline. When the mouse is over a manual light a small square appears with a two-dimensional gradient resembling the one in the beginning of this article. The manual value is set by dragging the mouse to the desired color temperature and brightness in the small square. A light remains in manual mode until it is clicked again.

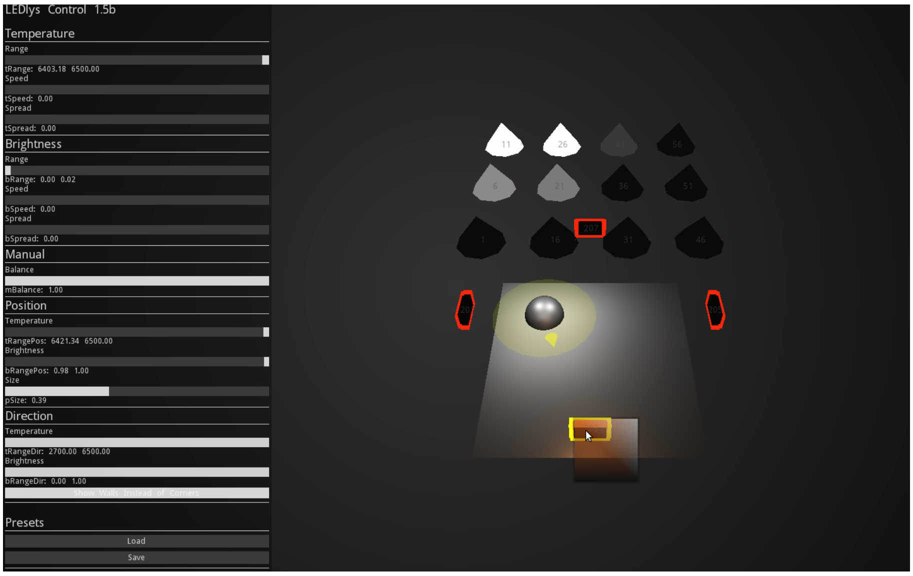

Figure 23: Four lights are in manual mode, the lowermost light is currently being manipulated as indicated by the yellow outline and shows a popup for manually setting the temperature and intensity.

There is also a slider called balance in the manual section of the controls. This slider allows to face the manual lights between their manual setting and the values they would otherwise get from the system. In the screenshot above the manual balance slider is at full, and the user uses manual mode on the four wall light fixtures to turn off the luminescent walls and only have the down lights react to the position input.

Conclusion

This paper presented three physical dynamic lighting setups with accompanying control software. The setups facilitate situated practice-based experiential research. The aim is to qualify an emergent generative and interactive aesthetics for coherent compositions of dynamic lighting with granular control over dynamic white LED lights.

The software prioritizes simple composition of fluctuating light scenes, that can be saved as presets. The growing collection of presets can act as references and be compared by the engaged participants. The researchers can present a collection of preset compositions for an invited set of peers from diverse fields. Based on in situ interviews the emergent qualities can be discussed, named and start to inform qualitative and subtle designs of dynamic light behaviors.

References

Bevins, Jason, *libnoise: Glossary, Perlin Noise, 2005*. Available from: <http://libnoise.sourceforge.net/glossary/index.html#perlinnoise>. [29 August 2014]

openFrameworks, computer software, n.d. Available from: <http://openframeworks.cc>. [1 September 2014]

Perlin, Ken, n.d. *Ken's Academy Award.* Available from: <http://mrl.nyu.edu/~perlin/doc/oscar.html>. [29 August 2014]

Campbell, Jim, 2001, *Fifth Avenue Cutaway 1*. Available from: <http://www.jimcampbell.tv/portfolio/low_resolution_works/fifth_avenue/fifth_avenue_cutaway_one/> [2 September 2014]