Interactive Lighting Design Using Coded Light

Dragan Sekulovski[1], Lorenzo Feri[1], Tim C.W. Schenk[1], Jean-Paul M. G. Linnartz[1]

Abstract
This paper presents an interactive approach for lighting design and control of solid state lighting (SSL) systems with a large number of light sources. This method is enabled by a new technology named Coded Light, in which invisible identifiers are embedded in the light output of these SSL sources. Using these identifiers, the local illumination contributions of the different light sources can be estimated. We present different rendering techniques for single and multiple points in the space, which covers both intensity and chromaticity control. The rendering techniques are designed such to use the estimated local illumination contributions. Finally the paper presents an experimental lighting system that implements the proposed interactive lighting design method.

Keywords: Light interaction, Solid state lighting, Scene setting, Light effect control.

1. Introduction

Advances in solid state lighting (SSL) have enabled its use as a viable alternative to traditional light sources [1,2]. The multitude of color tunable light sources that comprise a modern SSL system allows for finer control and more freedom in setting and adaptation of the color, and of spatial and temporal properties of the lighting system. This results in new possibilities for atmosphere creation and lighting effects.

The large freedom in settings, typical of such lighting systems, makes traditional controls ill-suited, even for professionals. The process of manipulating the parameters of each light source individually would be overwhelming to the user. The reason for this lies in the nature of traditional controls which are based on the “cause-effect” paradigm. For example, the user flips a wall switch to control the status of a light source (the “cause”), in order to create an illumination in the room (the “effect”). This control paradigm will fall short when applied to extensive SLL systems, because of the large amount of parameters and consequently the substantial number of possible effects.

Solutions for the control of light effects have been developed. Usually these solutions rely on a two step approach. The first step consists of the description of the complete target light effect. For example, the description can be given by means of a drawing that shows the distribution of the light colors over the scene. The second step consists of the mapping of the target scene into the real space. The mapping problem can be solved once the footprints of the available light sources are known.

The mapping is traditionally solved using computer simulations [3]. Often these simulators are based on either ray tracing or the radiosity method. In general, the accuracy of these simulators is only acceptable when the space, i.e. including furniture, is accurately modeled. However, this is very cumbersome work for the light designer. Moreover, experiencing the designed light scene on a computer screen is very different from being in the actual scene.

Another approach is based on recording the various light source footprints with a camera [4]. In this approach all light sources have to be switched on and off sequentially. This time-consuming approach has the following shortcomings. First, the resulting light effect depends on the camera view point, hence, an accurate analysis on the best camera location is required. Furthermore, a new calibration is required every time something changes in the space, e.g. when the interior of the space is changed or when the location or orientation of luminaires are varied. Last, and most noteworthy, this approach is not interactive, since the creative step and the mapping step are decoupled.

Interactivity is considered an essential part of an effective creative process. Furthermore, it would be advantageous for the user to design the light effect directly *in the space* and not on paper. To this end, we present a novel solution that uses a radically different paradigm for lighting control. Our solution allows the user to control the *effect* directly on the spot where it should be and without worrying about the contribution required by every single light source. The system is based on the use of *coded light* [5-7], where each lamp embeds invisible light source identifiers in its light output. In that way, by using

---

[1] Philips Research, Eindhoven, The Netherlands, {dragan.sekulovski, lorenzo.feri, tim.schenk, j.p.linnartz}@philips.com

a photo sensor, it is possible to discern the illumination contributions from all the light sources. A lighting system based on this technology allows users to create a light effect, in real time, by just choosing the desired effect (e.g. color, intensity, distribution) for the target location. The translation from desired effect to light source parameters is performed by an algorithm that uses the chromaticity properties and the intensity settings of all sources, as measured by the photo sensor.

The paper is organized as follows. Section 2 gives a general overview of the system used in the rest of the paper. Then, in Section 3, we present the coded light technology and we explain how it is used to estimate the contributions of the individual light sources at a given location. Subsequently, Section 4 describes the algorithms that allow light effect rendering using these estimates. A system setup that we used for testing of the presented concept and application development is then presented in Section 5. Finally, conclusions are drawn in Section 6.

2. System overview

Figure 1 schematically depicts a lighting system enabling the proposed lighting design method. The system consists of a large number of SSL sources installed in the ceiling, all embedding a unique identifier in their light output. The system additionally consists of a remote control (RC) and a system controller (SC). The RC is operated by the user and it is able, using an optical sensor, to receive the coded light identifiers in the light. The user input and the received identifiers are sent to the SC, which controls the SSL light sources accordingly.

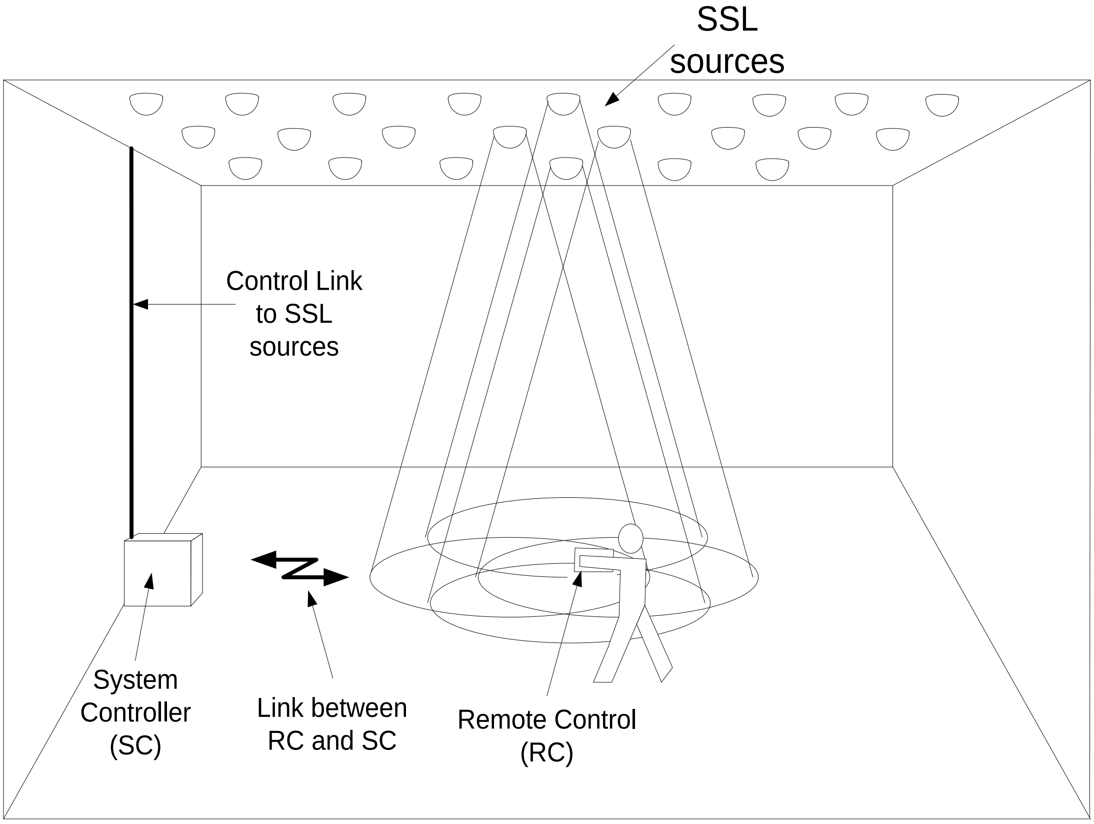

Figure 1. System overview. The system comprises SSL sources emitting coded light, a remote control for user control, and a system controller that sends control commands to the SSL sources.

3. Coded Light

The introduction explained that for the interactive control of a complex SSL lighting system it is of importance to estimate the localized illumination contribution of each light source. In the system proposed here, this estimation is achieved using a technology called *coded light*, as earlier presented in [5-7]. In this technology every light source can be identified in the system using a unique identifier embedded in the light. This identifier should be invisible to the human eye, but can be detected using the optical sensor device in the remote control.

3.1 Requirements on light identifiers

The requirements on the coded light are determined by the interactive lighting control applications. The most important ones are:

a) *Independent identifiers and illumination*: The main function of the light emitting diodes (LEDs) in the lighting system is providing illumination, thus the embedding of identifiers should not affect the short-term average light output of the light sources. Also, the identifier embedding techniques should preferably be compatible with the typically applied pulse-width modulation (PWM) dimming of LED light output to achieve efficient driving of the light sources.

b) *Imperceivable*: The modulation of the LED light, to embed the identifiers, should not create visible flickering, otherwise it will disturb users of the lighting system. The invisibility can be achieved by minimizing the energy in low frequency components (approximately below 100 Hz) in the light source identifiers.

c) *Number of LEDs*: The system must be able to measure the contribution from each locally relevant light source individually and simultaneously. It should be able to operate in an environment with several hundreds of LEDs.

d) *Short response time*: The modulation method should allow fast light source identification and illumination estimation. This guarantees that the user experiences an immediate reaction after pressing a control button. Hence, a sensor should be able to identify and measure all relevant light sources within several tenths of a second.

3.2 Coded Light techniques

Different techniques meeting these requirements were previously presented in [5-7]. These techniques are compatible with PWM dimming of SSL light sources, and do not impact the illumination function. The proposed coded light techniques are based on coded division multiple access (CDMA) [5,6] and frequency division multiple access (FDMA) [7].

The CDMA method is illustrated in Fig. 2 with the solid line. Basically the bits of the light source identifier are embedded by slightly varying the length of the PWM pulse, a longer pulse identifying a “1” and a shorter pulse a “0”. Figure 2 shows the resulting light output of the nth LED embedding the identifier code $\mathbf{c}_i$ = [1 0 0 1]. The dashed line indentifies the light output for the normal PWM modulated signal, i.e. without embedding an identifier. The average duty cycle of the light is d_n = 50% for both signals in this example. From this we can conclude that the average illumination level is not changed due to the embedding of the identifier. The length of one PWM pulse is T seconds, and the length of the code equals M bits. The whole code is acquired in MT seconds. If MT is smaller than or equal to 0.5 seconds, also requirement d) is met. The peak illumination level of the LED is denoted by a_n, while the actual illumination level equals $a_n d_n$ due to the PWM dimming. In the case of FDMA, every LED is assigned a unique repetition frequency $f = 1/T$ of the PWM pulses. For both the CDMA and FDMA approach the light modulation will be invisible when f is high enough.

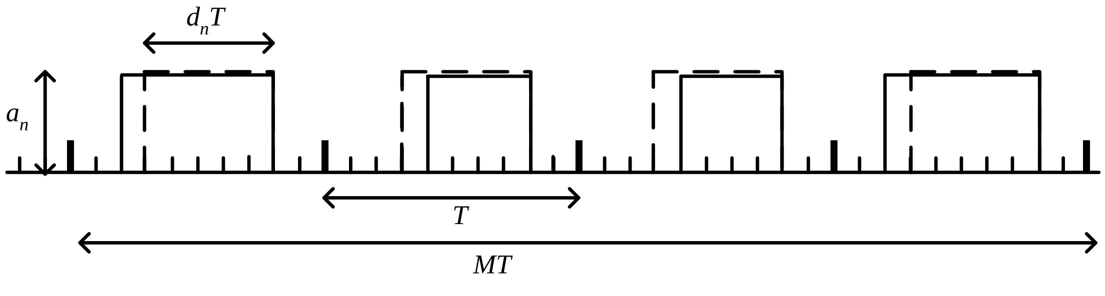

Figure 2. Embedding of the identifier code $\mathbf{c}_i$ = [1 0 0 1] in the 50% PWM dimmed light output of the nth LED light source.

The aggregated illumination contribution of all N LED sources at the ith target location can be written as

$$I_i = \sum_{n=1}^{N} a_n d_n \eta_{n,i} \,, \qquad (1)$$

where $\eta_{n,i}$ denotes the attenuation value for the light propagation from the nth light source to the ith target location.

In the remote control, one wants to estimate the maximum individual illumination contribution $b_{n,i} = a_n \eta_{n,i}$. To this end the remote control is equipped with an optical sensor, e.g. a photo detector, which converts the optical signal into an electrical signal. Then by applying digital signal processing to the received coded light signal, we find the estimate $\hat{b}_{n,i}$ for all N LEDs. These estimates are then used for light rendering.

4. Interactive Light Design

The estimates of the individual illumination contribution $\{\hat{b}_{n,i} | 1 \leq i < R, 1 \leq n < N\}$ can be used to control the amount of light that the system produces at a point in space to satisfy a user requirement on the light level at that point. This is achieved by controlling the duty cycles $\{d_n | 1 \leq n < N\}$ of the LEDs.

We present the user-system interactions and the algorithms used in an order of increasingly complex user requirements. First, we show how a user can control the light intensity in a single point in space; second, we show how the chromaticity of the light at the point in space can be controlled; and finally we show an algorithm that controls the LEDs to satisfy multiple user requirements.
In the rest of the section, we assume that all LEDs have the same peak illumination level, i.e. $a_n = a$. Under this assumption, the estimates of the light intensity contribution of a single LED n to a point i, $\hat{b}_{n,i}$, is inversely proportional to the distance between the LED and the point.

4.1 Single intensity requirement

The first problem that we solve using the estimates given above is the computation of the duty cycles required for the generation of the required light intensity in a point. In the case the required intensity at the target point i is the maximum light intensity at that point, the solution to this problem is trivial. However, if only a part of the total available illumination power is needed, the problem is under-constrained, as many combinations of LED duty cycle values can produce the desired light intensity.

To add a natural constraint to the distribution of the light we add a new parameter γ_i to the system that controls the distribution of the light intensity for the points around the target point. Using the estimates of the light intensity at the target point i, $\{\hat{b}_{n,i} | 1 \leq n < N\}$, the required light intensity u_i at point i, and the spatial distribution parameter γ_i, the duty cycle for LED n is computed as

$$d_{n,i} = u_i \frac{(\hat{b}_{n,i})^{\gamma_i - 1}}{\sum_{m=1}^{N} (\hat{b}_{n,i})^{\gamma_i}}. \tag{2}$$

It is easy to see, by substituting the computed duty cycles from equation (2) into equation (1), that when the required intensity at point i is smaller than the maximum achievable intensity at that point, the aggregated illumination contribution I_i, will be equal to the required one u_i.

The parameter γ_i takes values from 1 up to ∞, producing a spot of light around the target point with a variable size. When $\gamma_i = 1$, the system produces a uniform distribution, corresponding to the maximum spot size, i.e. all LEDs with a non-zero illumination contribution at the position i are assigned the same duty cycle. When $\gamma_i \rightarrow \infty$, only the LED with the highest contribution (the optically closest LED) has a non-zero duty cycle, while all the others have a zero duty cycle. This results in the smallest spot size that can be produced by the system. Figure 3 shows the relative duty cycles of the LEDs as a function of the relative individual illumination contribution for different values of γ_i. The relative individual illumination contributions are computed by dividing the individual illumination contributions by the maximal one, i.e. $\hat{b}_{n,i} / \max\{\hat{b}_{n,i} | 1 \leq n < N\}$. Similarly, the duty cycles are normalized by the maximal one to compute the relative duty cycles, i.e. $d_{n,i} / \max\{d_{n,i} | 1 \leq n < N\}$.

4.2 Color requirements

In the case of a system consisting of LEDs having different chromaticities, their mixing can produce different colors at different positions. The different parts of the system, consisting of the sets of LEDs with the same chromaticities are referred to as the primary systems. As most additive color controllable systems consist of three primary systems with red, green and blue chromaticities, we will assume a system with three primaries in the rest of the section. We denote the chromaticities of the three primaries in the CIE 1931 xyY color space by $(x,y)_\mathrm{R}$, $(x,y)_\mathrm{G}$, and $(x,y)_\mathrm{B}$, and their luminances at point i as $Y_{\mathrm{R},i}$, $Y_{\mathrm{G},i}$, $Y_{\mathrm{B},i}$. Given the CIE xyY coordinates, the CIE 1931 XYZ coordinates of the light incident at point i from the red, green and blue system can be computed and are denoted by $(X,Y,Z)_{\mathrm{R},i}$, $(X,Y,Z)_{\mathrm{G},i}$, and $(X,Y,Z)_{\mathrm{B},i}$.

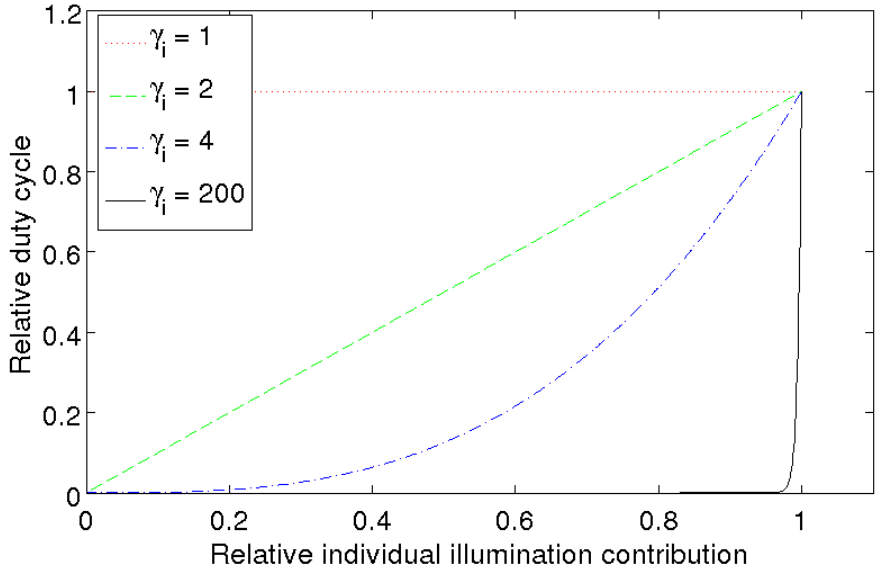

Figure 3. Relative duty cycles for LEDs as a function of the relative individual illumination contribution for different values of γ_i.

Due to the additive nature of light and the trichromacy of human color vision, the problem can be subdivided in two parts. The first one is finding the combination of the intensity of the primaries that will produce the desired chromaticities and the second one is computing the duty cycles that produce the desired intensity per primary system.

Given the required chromaticity $(x,y)_i$ and the required luminance Y_i for a point i, or $(X,Y,Z)_i$ in CIE XYZ coordinates, the required luminances for the primary systems $u_{\mathrm{R},i}$, $u_{\mathrm{G},i}$, and $u_{\mathrm{B},i}$ are computed using a standard [8] transformation

$$\begin{bmatrix} u_{\mathrm{R},i} \\ u_{\mathrm{G},i} \\ u_{\mathrm{B},i} \end{bmatrix} = \begin{bmatrix} X_{\mathrm{R},i} & X_{\mathrm{G},i} & X_{\mathrm{B},i} \\ Y_{\mathrm{R},i} & Y_{\mathrm{G},i} & Y_{\mathrm{B},i} \\ Z_{\mathrm{R},i} & Z_{\mathrm{G},i} & Z_{\mathrm{B},i} \end{bmatrix}^{-1} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}. \tag{3}$$

In the second step, using the required luminances for the primary systems $u_{\mathrm{R},i}$, $u_{\mathrm{G},i}$, and $u_{\mathrm{B},i}$ and equation (1), the duty cycles for the LEDs in each primary system are computed separately using equation (2).

In the case of a number of primaries larger than three, only the first part of the algorithm has to be changed. In this case there is no unique combination of primary luminances that produce the required chromaticity at the required luminance. Hence, additional constraints or optimization criteria have to be used. One example optimization criterion is the minimization of the power used to produce the required settings. Another is the maximization of the color rendering quality provided by the illumination system.

Figure 4 shows an example user interface that can be used to select the desired chromaticity, luminance and spatial distribution at the target point in space, determined by the position of the remote control. The desired chromaticity is selected on the chromaticity diagram. The luminance and spatial distribution are selected using up-down dials.

The system can be in active mode, where the target position follows the position of the remote, or in a passive mode, where the last active position of the remote is taken as the target position. Using the button “Freeze” the user can change between the active and passive mode. The user interface can for instance be run on a PDA that has the remote control sensor integrated.

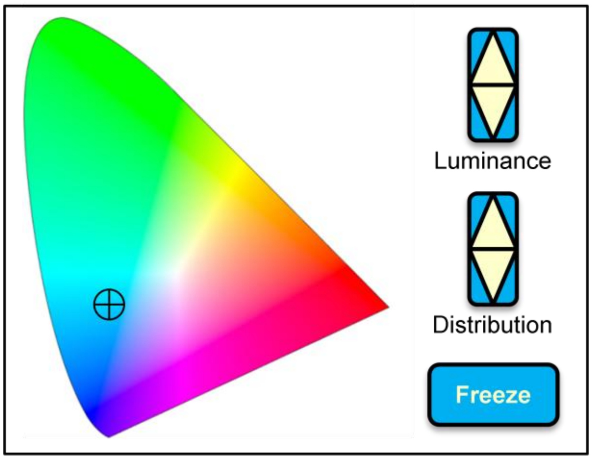

Figure 4. An example remote control user interface to select the chromaticity, luminance and spatial distribution at a target point.

4.3 Multiple requirements

In a practical system, one wants to specify the chromaticity, luminance and spatial distribution at several target locations. Not all of these requirements can be satisfied simultaneously, so the system has to weigh the different requirements to come to a rendering of the scene. The proposed solution is based on the least squares method and provides a tradeoff between a local and global accuracy.

Given *R* requirements for a primary system, the duty cycles that satisfy the individual requirements $d_{n,i}$, and a local priority parameter ρ_i, the duty cycle for LED n is calculated as the weighted average of the duty cycles for the individual requirements

$$d_n = \sum_{i=1}^{R} d_{n,i} \frac{(\hat{b}_{n,i})^{\rho_i}}{\sum_{m=1}^{N} (\hat{b}_{m,i})^{\rho_i}} . \quad (4)$$

The local importance parameter ρ_i takes values from 0 up to ∞. It controls the spatial weighting of the individual duty cycles $d_{n,i}$ as function of the distance between LED n and position i. The values of the weights for different distances and different importance parameter values are similar to the ones shown in Fig. 3. When $\rho_i \to \infty$, the value of the duty cycle of the nth LED, d_n, approaches the value of $d_{n,r}$, where r is the target point closest to the LED. So, for large values of ρ_i, requirements have mainly a local influence. When $\rho_i = 0$, $d_{n,i}$ will have the same influence independent of the distance between LED n and position i, i.e. the local and the global influence of the requirement is the same.

Figure 5 shows an example user interface for the selection of multiple requirements. Additional to the controls of the single requirement selection user interface, given in Fig. 4, the multiple requirements user interface includes a list of saved requirements and a control over the local importance parameter. Additionally, two buttons that control the addition and the removal of requirements from the list are added.

The list initially has one entry, denoted by the label “1”. The user initiates the interactive design by pressing the “Freeze” button, which initiates the active mode. After selecting the color and distribution parameters, the user can save the requirement using the “Save” button. This adds a new entry in the requirement list, and selects the new requirement as the active one.

Previously saved requirements can be changed by clicking on their label in the requirements list, which activates the selected requirement. When activated, the previously saved requirements are in passive mode and have the position at which they were saved as a target position. Pressing the “Freeze” button enters active mode, where the target position for the selected requirement follows the position of the remote. The user can remove requirements by selecting one from the list and pressing the “Delete” button.

The example user interface shows one possible interaction with the system, which allows the control of chromaticity, luminance and spatial distribution for a number of points in space. This allows the user the same freedom of design as having a number of light spots with a variable color and beam angle. The advantage of the presented system is that the design is done without any mechanical movement or reconfiguration of the system. Hence, the requirements from a user point of view can be seen as a set of *virtual lights,* giving the definition of the desired *effect*. The system consequently translates these requirements into a set of controls for the light sources present in the environment. The user is never exposed to the complexity of the system, the number of light sources and their capabilities.

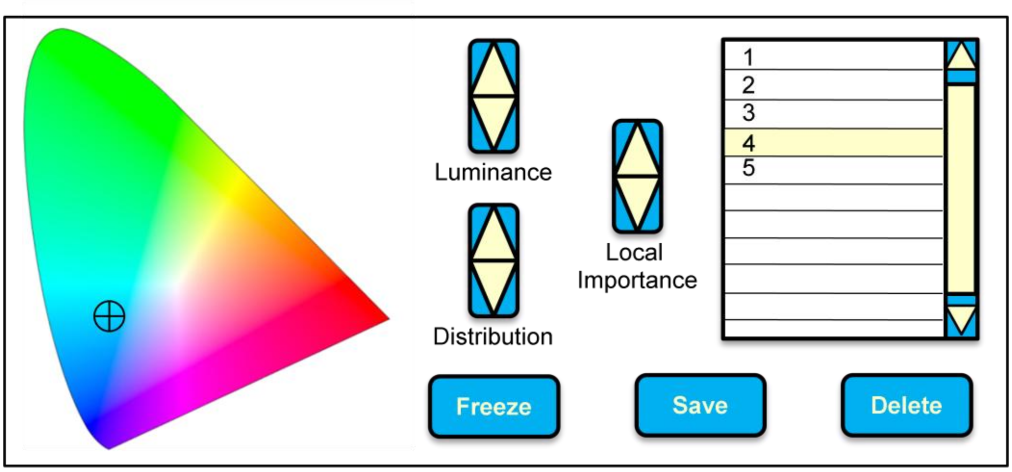

Figure 5. An example remote control user interface to select multiple requirements.

5. Experimental setup

An experimental setup was realized to develop and test interactive light effect control applications based on coded light. A block diagram of this setup is presented in Fig. 6. The system consists of eight LED-based light sources installed in the ceiling, a RC and a SC. Each light source has three primaries, i.e., red, green and blue, which are independently controllable and assigned a unique identification code. Hence, 24 unique identifiers are embedded in the light, for which the CDMA technique was applied. The RC is implemented in a standalone unit equipped with a sensor that is able to estimate the illumination contributions corresponding to the different LEDs. The SC is implemented in a laptop. The interfacing between the RC and the SC, as well as between the SC and the SSL sources, are implemented as wired serial links. For simplicity of implementation, the user interface is not implemented in the remote control but in the laptop.

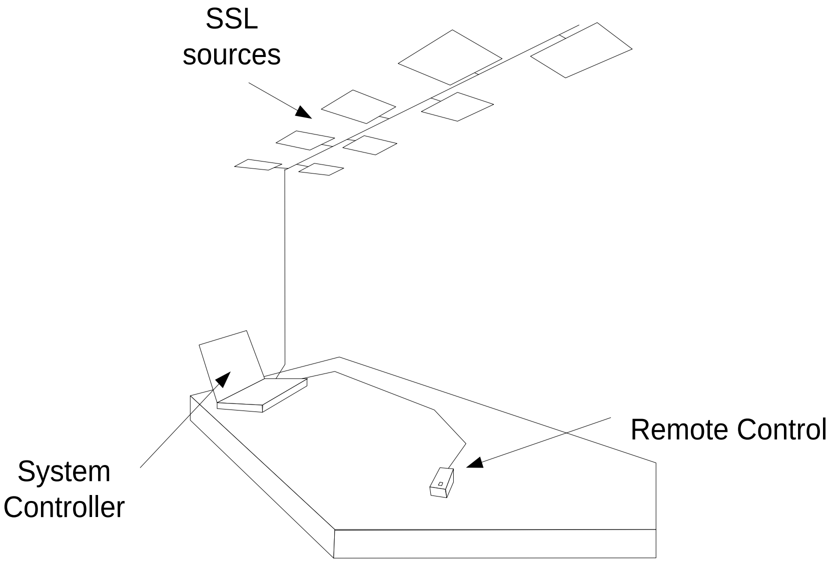

Figure 6. Block diagram of the implemented test setup.

The algorithms for interactive light effect control described in Section 4 are implemented in the laptop in a combined LabView-Matlab software environment. The performed tests proved the robustness of these algorithms in effectively rendering up to eight light effects in eight distinct locations. Figure 7 captures an example of one of the tests. In the left diagram of Fig. 7, the light effect editing is shown. During this phase, the remote control is placed on the target location and the desired color is selected via the user interface. In the right diagram of Fig. 7, the result of the editing and the following rendering is shown.

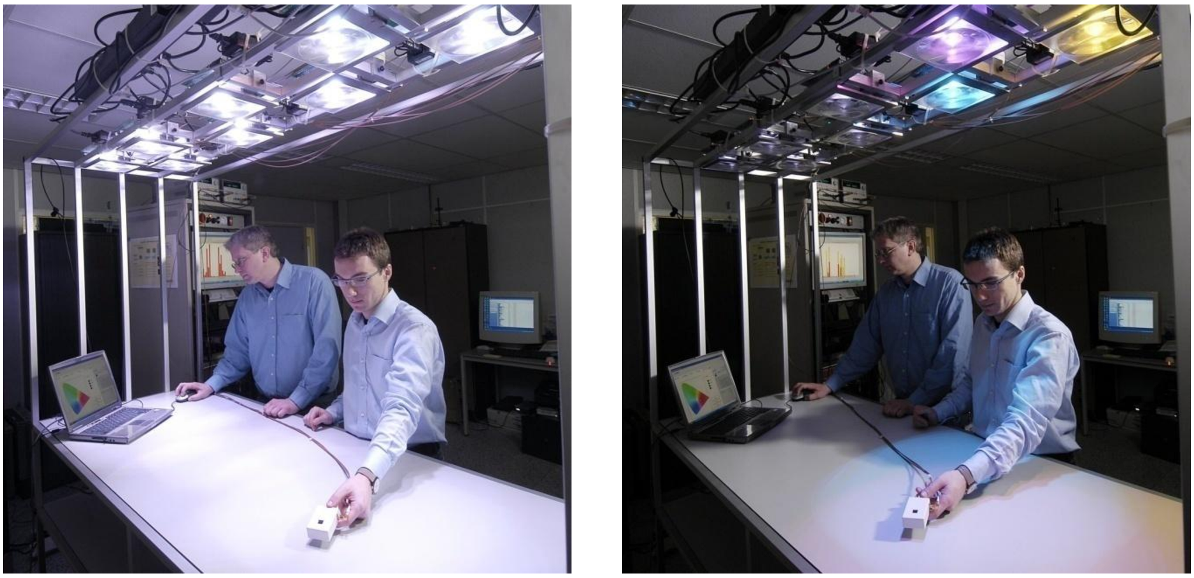

Figure 7. Intuitive light effect control application as tested in the experimental setup. In the left diagram, the target location and color are selected. The right diagram shows the resulting light effect.

6. Conclusions

We presented an interactive lighting design and control approach based on *coded light* for large LED-based lighting systems. The coded light technology enables the online estimation of individual illumination contributions, using invisible light source identifiers. Algorithms were presented to allow light rendering for single and multiple target positions in such system. These algorithms enable the control of chromaticity, luminance and spatial distribution. These were implemented in a test system presented here.

References

1. E. F. Schubert and J. K. Kim, “Solid-state light sources getting smart," *Science*, vol. 308, pp. 1274-1278, May 2005.
2. S. Muthu, F. J. P. Schuurmans, and M. D. Pashley, “Red, green, and blue LEDs for white light illumination," *IEEE J. Select. Topics Quantum Electron.*, vol. 8, no. 2, pp. 333-338, Mar./Apr. 2002.
3. DIALux software, DIAL GMBH, http://www.dial.de/CMS/English/Articles/DIALux/Features/Features.html
4. A. Sarkar, M. Fairchild, C. Salvaggio, “Integrated daylight harvesting and occupancy detection using digital imaging” in proc. *Sensors, Cameras, and Systems for Industrial/Scientific Applications IX*, Vol. 6816, 68160F, Jan. 2008.
5. J.-P. M.G. Linnartz, L. Feri, H. Yang, S. B. Colak, and T. C. W. Schenk, “Communications and Sensing of Illumination Contributions in a Power LED Lighting System,” in proc. *IEEE ICC*, May 2008, , pp. 5396-5400.
6. J.-P. M.G. Linnartz, L. Feri, H. Yang, S. B. Colak, and T. C. W. Schenk, “Code division-based sensing of illumination contributions in solid-state lighting systems," accepted for publication in *IEEE Trans. on Sign. Proc.*, 2009.
7. H. Yang, J.W.M. Bergmans, and T.C.W. Schenk, “A Filter Bank Approach for LED Illumination Sensing based on Frequency Division Multiplexing,” in proc. of *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 2009, pp. 3189-3192.
8. G. Wyszecki, W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*, Wiley-Interscience; 2 ed., 2000.