On top-down architectural lighting design
Constraint-based generation of light sources

Martin Moeck
Department of Architectural Engineering, University of Kansas, USA

Key words: Top-down design, constraint satisfaction optimisation, lighting design, visual performance criteria

Abstract: One key problem of architectural lighting design is to specify goals that relate to aesthetics. Since visibility is an important criterion for many visual tasks and objects, heuristics from industrial lighting and visual inspection can be used to describe the appearance of objects relevant to architectural lighting design, and to derive corresponding light sources. This has the potential to bring computation time in the range of near-interactive rates. A combination of two constraining inputs, which are the specification of desired material appearance and the selection of highlights and shadows can be successfully used in determining light sources.

1. OVERVIEW

1.1 Computer aided lighting design tools

Can the art of architectural lighting be automated? Several top-down tools exist.

Poulin and Fournier present a system where the user can control the definition and position of a light source by manipulating highlights and/or shadows. They use simple geometric constraints and a simple Phong-type illumination model to define the size of a highlight (Poulin, Fournier, 1992).

Schoeneman et al. use linear optimization (Schoeneman, Dorsey, et al,, 1993): the program user "paints" colored light onto surfaces and modifies given initial light positions and intensity distributions, while the computer determines the closest fitting combination of the given lights at user-defined locations.

Kawai et al. use unconstrained optimization techniques (Kawai, Painter, et al., 1993): the lighting designer specifies high-level goals such as "visual clarity" that relate to lighting patterns in diffuse environments. These objective functions are constrained through minimum lighting levels in specific locations, while the computer searches for the "best" possible settings for light source emissivities, surface reflectivities and spotlight directionality". In order to reduce the number of free variables, the program user must select active variables and impose constraints on them.

An approach based on randomly generated, simplified lights produces hundreds of lighting solutions for a simple scene (Marks, Andalman, et al., 1997). The solution generation and its computation time can become prohibitively expensive.

Computer based lighting design techniques based on expert rules change luminaires with respect to architectural elements (Moeck, 1999). Heuristics and expert rules are used to determine the location and rotation of realistic luminaires, given desired light patterns such as a "wash" of light. Scene geometry and material parameters can be of any complexity.

Poulin and Fournier describe another system (Poulin, Fournier, 1995). The user selects colors and applies them to any visible point in the scene. The system attempts to optimize certain functions or to fit values so the color points will remain as close as possible to their assigned colors when the full rendering is completed. The system uses simple illumination models, simple material descriptions, and simplified light sources.

Costa, Sousa, and Ferrera use optimisation to find lighting solutions from the scene geometry, the scene materials and user-specified fictitious luminaires used as design goals (Costa, Sousa, et al., 1999). The desired and undesired locations and emission directions of those luminaires must be specified using a script language.

1.2 A new approach

Most approaches suffer from the fact that lighting expert knowledge is not used. This leads to the selection of constraints that are often not relevant in practice, and to the subsequent generation of trivial solutions. In addition, most approaches do not design the desired dark areas of the scene (Waldram, 1978). Max Reinhardt said, “I am told that the art of lighting consists of putting the light where you want it and taking it away where you don’t want it” (Jones, 1941).

This paper presents heuristics-based constraint satisfaction optimization for the top-down lighting design problem: the user describes the desired appearance of selected objects or architectural features in terms of highlights and shadows. Object geometry, material reflectance functions, and luminaire intensity distributions can be of arbitrary complexity. The design problem is further constrained by specifying the desired appearance of the highlights and the shadows, i.e., strong texture, weak gloss, or flat appearance, etc. Heuristics are used to determine the optimum direction of light. Based on the desired contrast and specific material characteristics, light source locations can be computed (and excluded) such that the desired highlights and shadows are achieved.

2. CONSTRAINT BASED REASONING

Many problems, including planning, can be formulated as Constraint Satisfaction Problems (CSPs) (Shapiro, 1987) (Leler, 1988) (Meseguer, 1989) (du Verdier, Tsang, 1991) (Kumar, 1992) (Guesgen, Hertzberg, 1992) (Kautz, Selman, 1992) (Mackworth, 1992) (Yang, 1992) (Tsang, 1993) (Jampel, Freuder, et al., 1996). A finite CSP is a problem composed of a finite set of variables, each of which is associated with a finite domain, and a set of constraints that restricts the values the variables can simultaneously take. The task is to assign a value to each variable satisfying all the constraints.

2.1 Constraint satisfaction in illumination engineering

In illumination engineering, visibility is a constraint which is used to evaluate luminance distributions that aid the perception of visual signals (Rea, 1993a) (Rea, 1993b) (Rea, 1993c) (SLG, 1992a) (Rea, 1993d). Light source properties to maximize visibility and contrast and minimize veiling reflections have been studied in industrial and office lighting (Rea, 1993e) (Reitmaier, 1979) (Rea, 1993f) (Worthey, 1989a) (Worthey, 1989b) (Worthey, 1990). Real-world applications include visual inspection for quality control of surfaces in manufacturing. Inspection techniques depend on the installation of lighting systems that will maximize the visibility of particular material properties (Rea, 1993e) (IES, 1952) (ANSI, 1972) (SLG, 1992b) (SLG, 1992c), such as scratches and cracks in matte glass, plywood, or stone plates, as well as bumps and dents in metal and plastic sheets, and scratches, granularity, or engravings on polished plates and sheets. Other visual signals include scales and meters behind glass covers, or printed circuit boards on light backgrounds (Coderre, 2000) (Herman, Radice, et al., 2000).

Different lighting systems lend themselves to reveal detail. Important light locations include overhead lighting, grazing light, lighting from the mirror angle, and lighting with the line of sight. By relating light source properties and light location or direction to specific material properties, it is possible to maximize or minimize the luminance of specific areas. This process can be seen as the CSP.

2.2 Constraints in architectural lighting design

Characteristic features and objects of an architectural setting should have visual emphasis. By relating visual emphasis to luminance contrast, a computer aided lighting design procedure would entail the selection of architectural features as target and background corresponding to highlights and shadows, and the establishment of appropriate luminance differences to enhance their visibility (Moeck, 2000).

The key idea is to specify the spatial relationship between the luminaire and the illuminated object with respect to its surface normal and the viewing direction. Maximum and minimum luminance of various materials depend on specific altitude and azimuth angles between the incident light ray, the surface normal of the illuminated object, the view ray, and the reflective properties of the object (Moeck, 2000) (Worthey, 1989a) (Worthey, 1989b) (Worthey, 1990). These heuristics can be used to optimize the constraints to be applied:

"... heuristics for selecting the right constraints and the right values for these constraints are means for further improvement" (Guesgen, Hertzberg, 1992, p. 65).

Thus, a CSP in lighting design should not only be satisfiable (luminaire locations are found), but optimal, where optimality is defined in application-specific functions according to the domain knowledge (Tsang, 1993, p. 10, p. 300-319). These problems are called Constraint Satisfaction Optimization Problems (CSOP) to distinguish them from the standard CSP. They are successful in CSPs that are too large to be solved by complete algorithms (Jampel, Freuder, et al., 1996, p. 207-216), and they are used here.

2.2.1 Constraint formulation

A CSOP is defined as a CSP together with an optimization function f which maps every solution to a numeric value: (Z, D, C, f), where Z = the finite set of light source location variables defined by their altitude “alt” and azimuth angles “az” with respect to the illuminated object {alt, az};

D = a function which maps every variable in Z to a set of luminaires:

D: Z $\rightarrow$finite set of luminaries.

D_{alt} contains possible values of altitude angles alt. The set D_{alt} is called the domain of alt. The same is true for azimuth angles az.

C = a finite (possible empty) set of constraints on an arbitrary subset of variables in Z.

If S is the set of solutions of (Z, D, C), then f:S $\rightarrow$ numerical value

The task in a CSOP is to find the solution set with the optimal (maximal or minimal) f-value with regard to the application-dependent optimization function f. Two important methods for tackling CSOPs are the branch and bound algorithms (Lawler, Wood, 1966) (Hall, 1971) (Reingold, Nievergelt, et al., 1977) (Aho, Hopcroft, el al., 1983), and genetic algorithms (Holland, 1975) (Goldberg, 1989) (Davis, 1991) (Tsang, Warwick, 1990). Branch and bound makes use of knowledge of the f function and relies on the availability of good heuristics for estimating the 'best' values. A similar approach will be used here.

2.2.2 Definition of variables

The maximum and minimum luminance of a point on an opaque object depends on the altitude and azimuth of the incident light. Angular values for various material properties (rough-smooth, matte-glossy, textured-flat, light-dark) can be found in the literature (Moeck, 2000) (Reitmaier, 1979) (Worthey, 1989a) (Worthey, 1989b) (Worthey, 1990) (IES, 1952) (ANSI, 1972) (SLG, 1992b) (SLG, 1992c) (Rea, 1993g) (Coderre, 2000) (Herman, Radice, et al., 2000). These values allow the definition of optimization functions, leading to a CSOP. In order to express the desired appearance of highlights and shadows of an object to be illuminated as constraints, the interface shown in figure 1 is used.

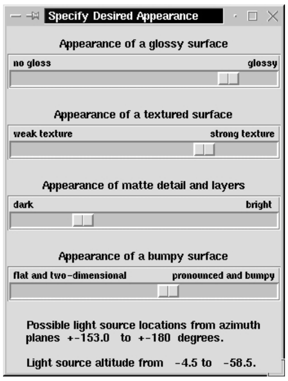

Figure 1. Appearance parameters used as constraints

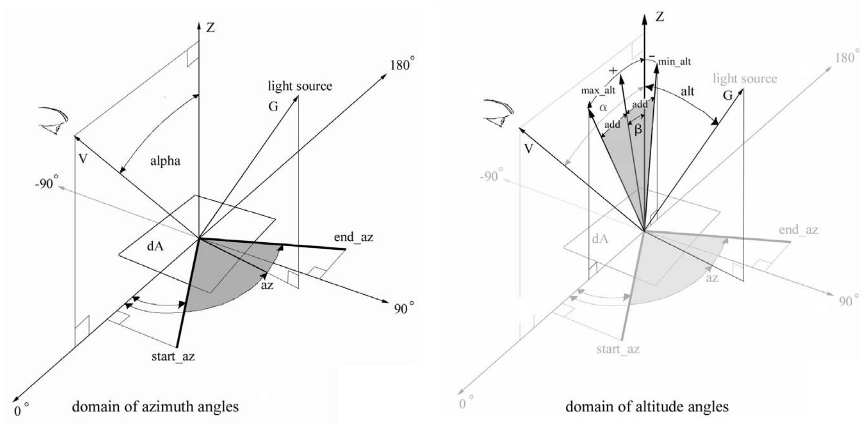

Figure 2. The direction G from the highlight to the light source is to be determined. The possible range of azimuth angles of G starts at start_az and ends at end_az, while the range of possible altitude angles for G starts at min_alt and ends at max_alt.

Figure 2 shows a point on an object to be illuminated, given a view point. Its luminance is to be maximized or minimized. Its surface normal is Z, and the eye ray is V. The direction vector to the light source is G. G is defined by its altitude and azimuth angles az and alt with respect to to V and Z. alt and az need to be found for a set of points on an architectural feature or object with specific material properties.

3. IMPLEMENTATION

3.1 Optimisation functions

The material properties corresponding to the constraints of figure 1 depend on the light source location, and should be enhanced or suppressed. For example, the scale variable gloss operates from "no gloss" (gloss = 0) to "glossy" (gloss = 100). A desired glossy appearance means to place luminaires at the mirror angle, no gloss means to place the light along the eye ray direction. A desired strong texture places lights at the grazing angle. Texture suppression is achieved by placing lights along the surface normal of the object.

The optimization functions shown in tables 1 - 3 determine the light source direction G for highlights and shadows. These constrain the values of Z{alt, az} using heuristics. Input is from the four scale values operating from 0 (left value) to 100 (right value) from figure 1.

Table 1. "Appearance of a glossy surface"
Scalevalue = gloss; alpha = 45 degrees
No gloss: gloss = 0; glossy: gloss = 100

Formula for angle	Angle values for gloss = 0	Angle values for gloss = 25	Angle values for gloss = 50	Angle values for gloss = 75	Angle values for gloss = 100
If gloss <= 50 then start_az = 0 else start_az = 180 – 2*((abs(gloss-100)/100)*90)	0	0	0	135	180
If gloss <= 50 then end_az = start_az + (gloss/50)*90 else end_az = 180	0	45	90	180	180
add = 50 – abs(gloss –5 0)*(90/50)	0	45	90	45	0
beta = (-(gloss-50))*alpha/50	alpha	22.5	0	-22.5	-alpha
max_alt = beta + add	alpha	67.5	90	22.5	-alpha
min_alt = beta - add	alpha	-22.5	-90	-67.5	-alpha

Table 2. "Appearance of a textured surface"
Scale variable: texture; alpha = 45 degrees
Weak texture: texture = 0; strong texture: texture = 100

Formula for angle					
start_az = 0	0	0	0	0	0
end_az = 180	180	180	180	180	180
add = (50 – abs(texture – 50))*90/50	0	45	90	45	0
If texture < 50 beta = 0 else beta1 = (texture-50)*90/50 beta2 = -beta1	beta = 0	beta = 0	beta1 = 0 beta2 = 0	beta1 = 45 beta2 = -45	beta1 = 90 beta2 = -90
If texture < 50 max_alt1 = beta + add min_alt1 = beta – add else max_alt1 = beta1+ add min_alt1 = beta1 – add max_alt2 = beta2+ add min_alt2 = beta2 - add	max_alt1 = 0 min_alt1 = 0	max_alt1 = 45 min_alt1 = -45	max_alt1 = 90 min_alt1 = -90 max_alt2 = 90 min_alt2 = -90	max_alt1 = 90 min_alt1 = 0 max_alt2 = 0 min_alt2 = -90	max_alt1 = 90 min_alt1 = 90 max_alt2 = -90 min_alt2 = -90

Table3. "Appearance of a bumpy surface"
Scale variable: eyeray; alpha = 45 degrees
Flat appearance: eyeray = 0; bumpy appearance: eyeray = 100

Formula for angle	Angle value for eyeray = 0	Angle value for eyeray = 25	Angle value for eyeray = 50	Angle value for eyeray = 75	Angle value for eyeray = 100
start_az = 0	0	0	0	0	0
If eyeray < 50 end_az = 3.2*eyeray + 20 else end_az = 180	20	100	180	180	180
add=(50–abs(eyeray–50)) * 90/50	0	45	90	45	0
If eyeray < 50 beta=-1*(eyeray-50)* alpha/50 else beta1 = (eyeray-50)*90/50 beta2 = -beta1	beta = 45	beta = 22.5	beta1 = 0 beta2 = 0	beta1 = 45 beta2 = -45	beta1 = 90 beta2 = -90
If eyeray < 50 max_alt1 = beta + add min_alt1 = beta – add else max_alt1 = beta1+ add min_alt1 = beta1 – add max_alt2 = beta2+ add min_alt2 = beta2 - add	max_alt1 = 45 min_alt1 = 45	max_alt1 = 67.5 min_alt1 = -22.5	max_alt1 = 90 min_alt1 = -90 max_alt2 = 90 min_alt2 = -90	max_alt1 = 90 min_alt1 = 0 max_alt2 = 0 min_alt2 = -90	max_alt1 = 90 min_alt1 = 90 max_alt2 = -90 min_alt2 = -90

The scale “Appearance of matte detail and layers” is very similar to the scale “Appearance of a textured surface” and its functions are omitted here.

The final solution is the angular range with the largest minimum and the smallest maximum of the four domains. The following constraints serve as an example. Desired is a highlight with an appearance described in figure 1. The following scale values are desired:

"Appearance of a glossy surface": gloss = 85
"Appearance of a textured surface": texture = 75
"Appearance of matte detail and layers": illum = 25
"Appearance of a bumpy surface": eyeray = 60

The angle alpha between the light of sight and the surface normal of the highlight is 45 degrees. Based on tables 1-3, the angular ranges for alt and az are as follows:

altitude angles for gloss: from -58.5 to -4.5 degres
azimuth range for gloss: 153.0 to 180 degrees
altitude angles for texture: from 0.0 to 90.0 and from -90.0 to 0.0 degrees
azimuth range for texture from 0 to 180 degrees
altitude angles for illum: from 0.0 to 90.0 and from -90.0 to 0.0 degrees
azimuth range for illum from 0 to 180 degrees
altitude angles for eyeray: from -54.0 to 90.0 and from -90.0 to 54.0 degrees
azimuth range for eyeray from 0 to 180 degrees

Therefore, the final altitude range is from -4.5 to -58.5 degrees, while the azimuth range is from -153 to -180 and from 153 to 180 degrees.

Figure 3 shows the corresponding light source arrangements with respect to a small square patch selected as the "highlight". The arrows correspond to vectors Z, x, and y from figures 2 and 3.

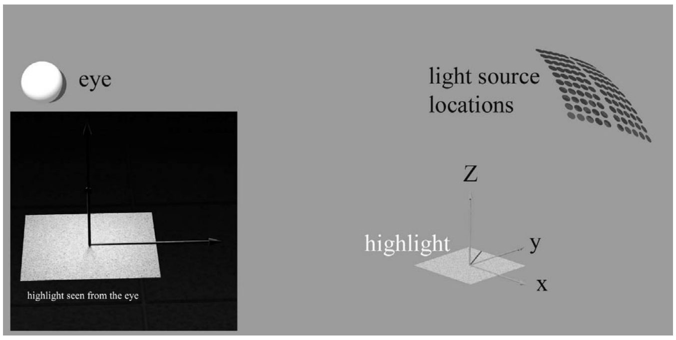

Figure 3. Highlight coordinate system and luminaire locations based on the constraints of figure 1

The implementation of the illumination of a 3D object based on constraints of figure 1 follows. Desired are the highlights and shadows on an object shown in figure 4.

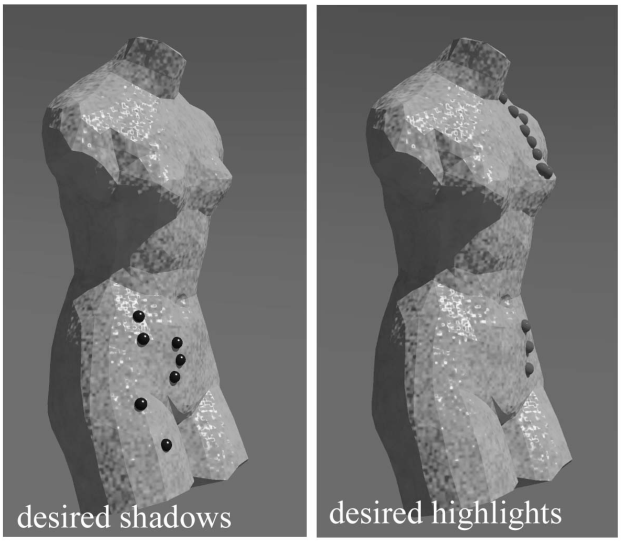

Figure 4. Desired highlights and shadows on a glossy, textured object

3.2 Algorithms

The resulting spotlights are arrived at as follows in algorithm 1:

3.2.1 Algorithm 1:

```
input: start_az, end_az, min_alt, max_alt

for each highlight
    for each view point
        determine vectors V, Z, x, and y
        az = start_az
        alt = min_alt
        while az <= end_az
            while alt <= max_alt
                calculate vector G with altitude angle alt and azimuth angle az
                create a light source with surface normal -G
                starting at the highlight, fire G into the scene
                determine the intersection with the nearest scene element: S
                place the light source with surface normal –G at S
                alt = alt + increment
            end while
            az = az + increment
```

```
    end while
  end for
end for
```

See figure5 for the object in an interior setting.

Figure 5. Object with illuminated highlights. The desired dark areas are partially illuminated.

The angular spacing between the luminaires (variable "increment" of algorithm 1) is 5 degrees. The 48 generated light sources are commercially available low voltage incandescent spotlight with a parabolic reflector and a very tight cutoff angle. Nevertheless, the desired dark areas are illuminated. This is due to the non-parallel emission characteristics of the spots. Therefore, it is necessary to introduce an algorithm to locate the spots interfering with the desired dark areas, as shown in algorithm 2. These spots need to be eliminated, reducing the domain D{az, alt}.

3.2.2 Algorithm 2

```
for each dark area
  for each view point
    determine vectors V, Z, x, and y
    az = start_az
    alt = min_alt
    while az <= end_az
```

```
        while alt <= max_alt
            calculate vector G with altitude angle alt and azimuth angle az
            starting at the dark area, fire G into the scene
            determine the intersection with the nearest scene element: A
            alt = alt + increment
        end while
        az = az + increment
    end while
end for
```

See figure 6 for luminaire locations and absorbers with locations “A”. It is obvious that some spots interfere with the desired dark areas.

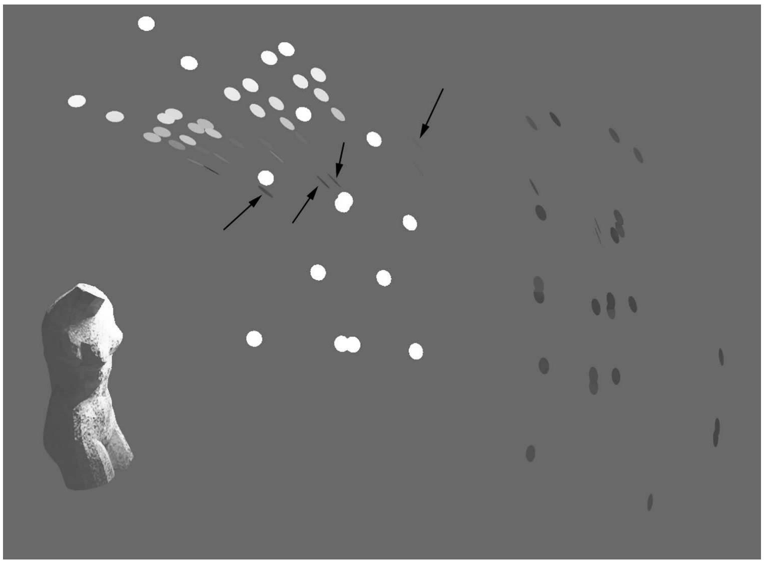

Figure 6. Some of the 48 luminaires with 5 degree angular spacing interfere with absorbers marked with arrows and must be removed.

The domain of possible light locations from algorithm 1 is reduced by eliminating all spots which are closer than a certain distance to any absorber location in A. For this particular case, it is approximately 2 m. The resulting illuminated object with the desired shadows and its reduced domain of 12 spotlights is shown in figure 7.

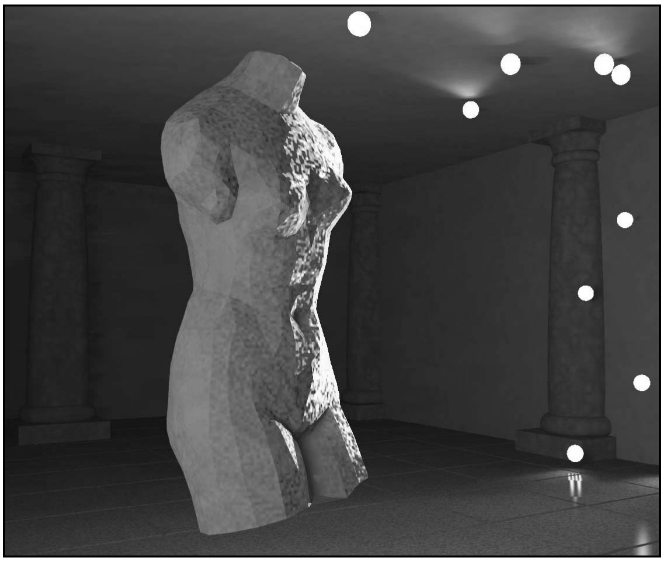

Figure 7. Illuminated object with 36 interfering spots removed. The remaining 12 lights fulfil three constraints: the highlights reflect light in the eye (high gloss), texture is emphasized through a grazing angle of light incidence, and the desired shadow areas are not illuminated.

Although the highlights and dark areas rather close, the desired luminance pattern is achieved, while the remaining population is still large enough to select individual spots. This would not have been possible using educated expert guesses about light source locations, because the 3D geometry of the illuminated object is too complex.

4. PROGRAM IMPLEMENTATION

This program uses the Radiance lighting simulation system for rendering (Ward, 1994). The code for the tool discussed is implemented in Tcl/Tk under Unix (Osterhout, 1994), and consists of the following stand-alone programs:

– a material selection tool. It allows the definition and assignment of a wide range of materials to the object to be illuminated.
– a viewer tool. It allows the definition and manipulation of viewpoints, view directions, view types, angular view sizes, zoom functions, view rotations, and so forth.
– the constraint definition tool shown in figure 1, which imposes constraints on the appearance of highlights and shadows

- a luminaire editing tool to change light source characteristics (wattage, size, cutoff angle, luminaire selection)
- an application for the selection of highlights and shadows on the object to be illuminated. This program manages the interaction of the other programs.

The stand-alone programs communicate via the send command. With send, any Tk-based application can invoke arbitrary Tcl scripts in any other Tk application on the display, allowing the development of reusable, small software packets.

5. DISCUSSION

In principle, highlights can be illuminated from 64080 possible light sources, assuming 1 degree angular spacing between the luminaires. This domain is too large to do exhaustive searches. However, different material characteristics (bumpy or flat, glossy or matte, textured or smooth) will yield drastically different appearances of highlights. These appearances relate to the amount of visible detail in the illuminated surface and can be used as design intentions to constrain the desired look of the highlighted material. Heuristics from industrial lighting and visual inspection techniques are used to derive the corresponding light incidence angles. These expert rules limit the angles to useful values. For example, a desired glossy appearance requires to place a light source at altitude angle -alpha and azimuth angle 180 degrees (see figure 2), while a surface without a glossy look requires an azimuth angle of 0 degrees and an altitude angle of alpha.

The formulae from tables 1 - 3 serve as a starting point for maximizing the luminance of highlights, and for minimizing the luminance of shadows. In the future, more comprehensive formulae should be developed to derive the light incidence angles. These should consider more sophisticated material parameters and more constraints or appearance descriptors, and could even consider mixed material properties (i.e. black ink on glossy paper). On the other hand, appearance descriptors need to be simple and intuitive to be understood by the lay program user.

Lighting design based on material properties requires rendering systems with material definitions based on bi-directional reflectance functions. The tool should be extended to include a building material library of known reflectance functions. In this way, the light incidence angles at which maximum luminance occurs could be calculated, in addition to using heuristics, and results could be fine tuned.

It has been claimed that lighting designers are also shadow designers, because they keep light away from desired dark areas. This is problematic due to spill light. Light beams are not parallel and will illuminate highlights and desired shadows. This problem has been solved by introducing an algorithm which locates light sources which conflict with the shadow areas. The initial luminaire population is screened and detrimental lights are eliminated, leaving a useful selection as a starting point for the final manual lighting design implementation. In the future, additional constraints could be easily imposed by selecting luminares based on their location (wall, ceiling, floor), orientation, distance, etc. In addition, the effect of a range of different view points and viewing directions needs to be tested.

The computation time is around 8 minutes for the case shown (400 Mhz processor with 128 MB RAM). This time is mainly a function of the desired angular spacing of the luminaires (see angle increment in algorithms 1 and 2). It also depends on how tight the constraints are set and how large the resulting solution space is. The expert rules expressed as formulae in tables 1 - 3 speed up generation significantly.

6. OUTLOOK AND FUTURE APPLICATION TO COMPLEX ARCHITECTURAL SETTINGS

This paper introduced the theory of CSOPs to top-down lighting design, using an example with one viewpoint, and one illuminated object with eleven highlights and seven shadow points. Future studies will determine the suitability of the algorithms to more complex architectural settings. This will include furnished rooms with various objects on display, many different viewpoints, and multiple constraints for various highlights and shadows scattered widely across the room. Additional constraints will exclude direct glare at the eye positions of observers exploring the space from various vantage points. This will make the study quite complex. It will determine the effects of multiple constraints, how their settings affect the size of the solution space, and how multiple viewpoints and additional glare constraints affect previous solutions. This is beyond the scope of this paper.

7. CONCLUSION

The advantages of constraint-based optimization are the setup of meaningful constraints, and the reduction of a very large solution space to a small domain. This makes computer aided lighting design a tractable problem. One key problem of CAAD is to relate constraints to esthetics, and the setup of lighting design goals which allow to specify the desired design product. Since visibility is an important criterion for many visual tasks and objects, heuristics from industrial lighting and visual inspection can be used to describe the appearance of objects relevant to architectural lighting design, and to derive corresponding light sources. This has the potential to bring computation time in the range of near-interactive rates. It has been shown that a combination of two constraining inputs, which are the specification of desired material appearance and the selection of highlights and shadows, or dark areas, can be a successful lighting design goal for determining light sources.

8. REFERENCES

Aho, A.V., J.E. Hopcroft, and J.D. Ullman, 1983, *Data structures and algorithms*, Addison-Wesley.

ANSI PH2.32-1972, 1972, *American national standard viewing conditions for the appraisal of color quality and color uniformity in the graphic arts*, American National Standards Institute, New York.

Coderre, J., 2000, "Machine Vision Illuminates the Future of Electronics", *Photonics Spectra*, December 2000, p. 90-92.

Costa, A.C., A.A. Sousa, and F. N. Ferreira, 1999, "Lighting Design: A Goal Based Approach Using Optimisation", *Rendering Techniques '99*, *Proceedings Eurographics Workshop*, Springer Verlag, Wien, p. 317-328.

Davis, L. (ed.), 1991, *Handbook of genetic algorithms*, Van Nostrand Reinhold, New York.

du Verdier, F, and J.-P. Tsang, 1991, "A spatial reasoning method by constraint propagation", *Proceedings 11th International Workshop of Expert Systems and Their Applications*, p. 297-314.

Goldberg, D.E., 1989, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley.

Guesgen, H.W., and J. Hertzberg, 1992, *A Perspective of Constraint-Based Reasoning*, Lecture Notes in Computer Science, Springer-Verlag, Berlin.

Hall, P.A.V. 1971, "Branch and bound and beyond", *Proceedings International Joint Conference on AI*, pp. 641-650.

Herman, J.W, J. Radice, J. Disciullo, and P. Chipman, 2000, "Flexible lighting modules", *Advanced Packaging*, May 2000, p. 27-32.

Holland, J.H., 1975, "Adaptation in natural and artificial systems", University of Michigan Press.

IES Committee on Lighting Study Projects in Industry, Subcommittee on Supplementary Lighting, 1952, "Recommended practice for supplementary lighting", *Journal of the Illuminating Engineering Society* 47(11), p. 623-635.

Jampel, M., E. Freuder, and M. Maher, 1996, *Over-Constrained Systems*, Lecture Notes in Computer Science, Springer-Verlag, Berlin.

Jones, R. E., 1941, *The Dramatic Imagination*, Duell Sloan & Pearce, New York, p. 111.

Kautz, H., and B. Selman, 1992, "Planning as satisfiability", *Proceedings 10th European Conference on Artificial Intelligence*, 1992, p. 360-363.

Kawai, J.K., J.S. Painter, and M.F. Cohen, 1993, "Radioptimization - Goal Based Rendering", *Computer Graphics Proceedings, SIGGRAPH 1993*, p. 147-154.

Kumar, V., 1992, "Algorithms for constraint-satisfaction problems: a survey", *AI Magazine*, Vol. 13, No. 1, Spring 1992, p. 32-44.

Lawler, E.W., and D.E. Wood, 1966, "Branch-and-bound methods: a survey", *Operations Research*, Vol. 14, 1966, p. 699-719.

Leler, W., 1988, *Constraint programming languages, their specification and generation*, Addison-Wesley, 1988.

Mackworth, A.K., 1992, "The logic of constraint satisfaction", *Artificial Intelligence*, Vol. 58, Nos. 1-3, p. 3-20.

Marks, J., B. Andalman, P.A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, and T. Kang, 1997, "Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation", *Computer Graphics Proceedings, SIGGRAPH 1997*, pages 389-400.

Meseguer, P., 1989, "Constraint satisfaction problems: an overview", *AI Communications*, Vol. 2, No. 1, March 1989, p. 3-17.

Moeck, M. 2000, "Lighting design based on luminance contrast", *Lighting Research and Technology* 32(2), 2000, p. 55-64.

Moeck, M., 1999, "On Computer Aided Lighting Design: Lighting Techniques", *Journal of the Illuminating Engineering Society of North America* 28(2), p.33-41.

Osterhout, J., 1994, *Tcl and the Tk Toolkit*, Addison-Wesley, Reading, Massachusetts.

Poulin, P., and A. Fournier, 1992, "Lights from highlights and shadows", *Computer Graphics*, volume 25, p. 31-38.

Poulin, P., and A. Fournier, 1995, "Painting surface characteristics", Rendering Techniques '95, *Proceedings Eurographics Workshop*, Springer-Verlag, Wien, p. 160-169.

Rea, M. (ed.), 1993a, "Visibility for Specific Visual Tasks", *IESNA Lighting Handbook*, 8th edition, IESNA, New York, p. 664.

Rea, M. (ed.), 1993b, "Visual ability", *IESNA Lighting Handbook*, 8th edition, IESNA, New York, p. 86-89.

Rea, M. (ed.), 1993c, "Visual Range of Luminous Signals", *IESNA Lighting Handbook*, 8th edition, IESNA, New York, p. 93-91.

Rea, M. (ed.), 1993d, "Luminance Contrast", *IESNA Lighting Handbook*, 8th edition, IESNA, New York, p. 83.

Rea, M. (ed.), 1993e, "Industrial Lighting", *IESNA Lighting Handbook*, 8th edition, IESNA, New York, p. 629-684.

Rea, M. (ed.), 1993f, "Office Lighting", *IESNA Lighting Handbook*, 8th edition, IESNA, New York, p. 517-540.

Rea, M. (ed.), 1993g, "Classification of Visual Tasks and Lighting Techniques", *IESNA Lighting Handbook*, 8th edition, IESNA, New York, p. 636-638.

Reingold, E.M., J. Nievergelt, J., and N. Deo, 1977, *Combinatorial algorithms: theory and practice*, Prentice Hall.

Reitmaier, R., 1979, "Some effects of veiling reflections in papers", *Lighting Research and Technology* 11(4), p. 204.

Schoeneman, C., J. Dorsey, B. Smits, J. Arvo, and D. Greenberg, 1993, "Painting with Light", *Computer Graphics Proceedings, SIGGRAPH 1993*, p. 143-146.

Shapiro, S.C., 1987, *Encyclopedia of Artificial Intelligence*, John Wiley & Sons, Chichester, England.

SLG Schweizerische Lichttechnische Gesellschaft (ed.), 1992a, "I-2 1.2.2 Hellempfindlichkeit", *Handbuch für Beleuchtung*, Ecomed Verlag, Landsberg, Germany, p. 2-3.

SLG Schweizerische Lichttechnische Gesellschaft (ed.), 1992b, "Tab. II-2.1.4: Sonderaufgaben in der Industrie", *Handbuch für Beleuchtung*, Ecomed Verlag, Landsberg, Germany, p. 4-14.

SLG Schweizerische Lichttechnische Gesellschaft (ed.), 1992c, "Tab. II-2.1/2: Örtliche Sehaufgabe am Arbeitsplatz - Sehaufgaben und praktische Lösungen", *Handbuch für Beleuchtung*, Ecomed Verlag, Landsberg, Germany, p. 4-14.

Tsang, E., 1993, *Foundations of Constraint Satisfaction*, Academic Press, London.

Tsang, E.P.K., and T. Warwick, 1990, "Applying genetic algorithms to constraint satisfaction problems", *Proceedings European Conference on AI*, p. 649-654.

Waldram, J.M., 1978, "Designed Appearance Lighting", *Developments in Lighting-1*, Chapter 5, Lynes, J.A., (ed.), Applied Science Publishers, London.

Ward, G.J., 1994, "The Radiance Lighting Simulation and Rendering System", *Computer Graphics Proceedings, SIGGRAPH 1994*, p. 459-472.

Worthey, J. A., 1989a, "Geometry and amplitude of veiling reflections", *Journal of the Illuminating Engineering Society* 18(1), p. 49-62.

Worthey, J. A., 1989b, "Effect of veiling reflections on the vision of colored objects", *Journal of the Illuminating Engineering Society* 18(2), p. 10-15.

Worthey, J. A., 1990, "Lighting quality and light source size", *Journal of the Illuminating Engineering Society* 19(2), p. 142-148.

Yang, Q., 1992, *Artificial Intelligence*, Vol. 58, Nos. 1-3, p. 361-392.