of theta and phi is listed, followed by the value corresponding to the first value of theta and the second value of phi, on through the end of the phis. When all phis are exhausted, the next number is the value corresponding to the second value of theta and the first value of phi. This continues until all thetas and phis are exhausted.

5.2.6 Modeling Transmitting Media

Some luminaires use semitransparent surfaces to cover the bright light sources hidden inside. This can include downlights and other lensed luminaires. In addition, some daylight control products, such as curtains and fritted glass, are most appropriately modeled as a semispecularly transmitting medium. In these cases, it is important to know how to properly specify the *trans* material type to accurately model the appearance of the luminaire or other light-transmitting architectural element.

Specifying a trans material can be intimidating. It is one of the most confusing material entities in the *Radiance* repertoire. However, it is the simplest material that will trace direct source rays through a semispecular surface in order to determine the diffuse and specular transmitted components.

You will have to measure the following surface properties in preparation for the creation of a trans material type:

- Diffuse reflectance (RGB): The color will affect both diffusely reflected light (if there is any surface roughness) and transmitted light. Call the red, green, and blue components **Cr**, **Cg,** and **Cb**. Calculate the photopic average of the RGB and call this **Rd.**
- Reflected specularity: As with glass, this is the fraction of light that is reflected off the first surface in a mirror-like way. This we'll call **Rs**; it is equal to floating-point argument 4, A4.
- Surface roughness (RMS): Facet slope as in plastic. Call this **Sr.**
- Diffuse transmissivity: Fraction of light that passes all the way through the surface diffusely. Call this **Td.**
- Transmitted specularity: Fraction of light transmitted as a beam—that is, the fraction of light *not* diffusely scattered. Call this **Ts.**

The following formulas can be used to calculate the A7 through A1 parameters for the trans material:

A7=Ts / (Td+Ts)

A6=(Td+Ts) / (Rd+Td+Ts)

A5=Sr

A4=Rs

A3=Cb / ((1-Rs)*(1-A6))

A2=Cg / ((1-Rs)*(1-A6))

A1=Cr / ((1-Rs)*(1-A6))

The behavior of trans is regulated by the -st specular threshold rendering parameter. The following formula can be used to determine the appropriate -st setting for rview, rpict, or rtrace to ensure that the transmitted (and reflected) semispecular component will be rendered. The variables `A1` through `A7` are the first through seventh floating-point arguments to the trans primitive as calculated above.

St = A6 * A7 * (1 - grey(A1, A2, A3) * A4)

Unlike most other *Radiance* material primitives, the trans material is neither intuitive nor straightforward to apply. It is a good idea to keep ample notes for later reference when you are creating a trans material. Backing out your assumptions from an unannotated trans primitive is difficult.

5.3 Analysis Techniques

Now that you have these advanced luminaire modeling and data collection skills at your disposal, we will present additional techniques and examples that will help you ground your analysis in real-world terms. The preceding sections about data collection in this chapter are required reading.

5.3.1 Visualizing Light Distribution Using Falsecolor

One of the most common and important tasks for a lighting designer is to determine the luminance levels of the space being designed. There are a number of ways to achieve this. Probably the easiest way is to generate a *false-color* version of a precalculated image. False color means that instead of displaying colors in the image based on the wavelength of light reflecting off the surfaces, we assign a color value that corresponds either to the luminance or to the illuminance at those pixel locations. If the view is an irradiance calculation (`render= -i`, or `rtrace -I`), the resulting false-color image will represent illuminance values.

The *Radiance* tool for generating false-color images is, not surprisingly, called **falsecolor**. It requires as input one precalculated image file. It can also accept two precalculated image files when the result you wish to display is a contour map of illuminance values on top of an image of radiance values. This retains the visible