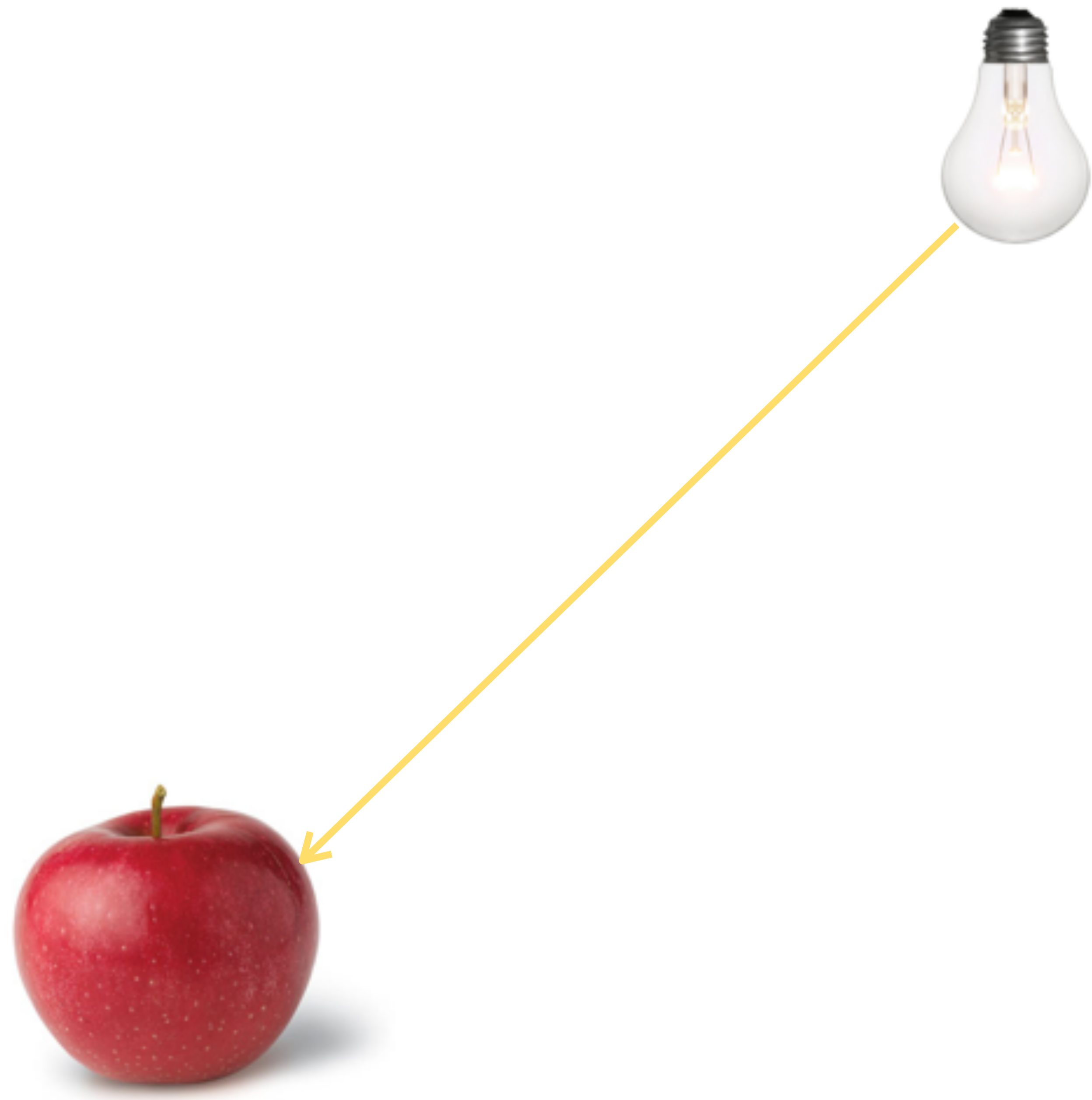
Basic Image Formation Model

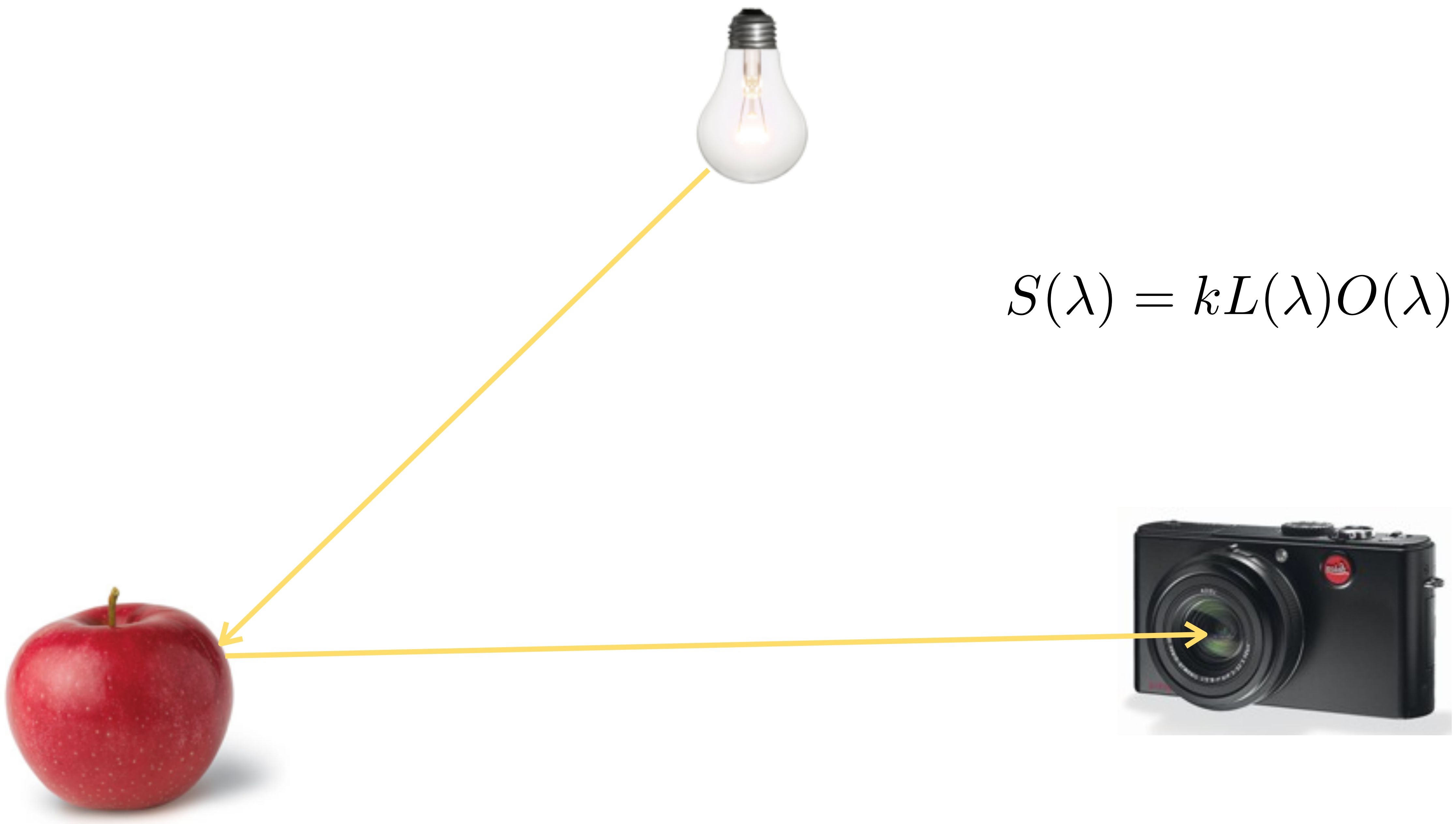
$$S(\lambda) = kL(\lambda)O(\lambda)$$

Basic Image Formation Model



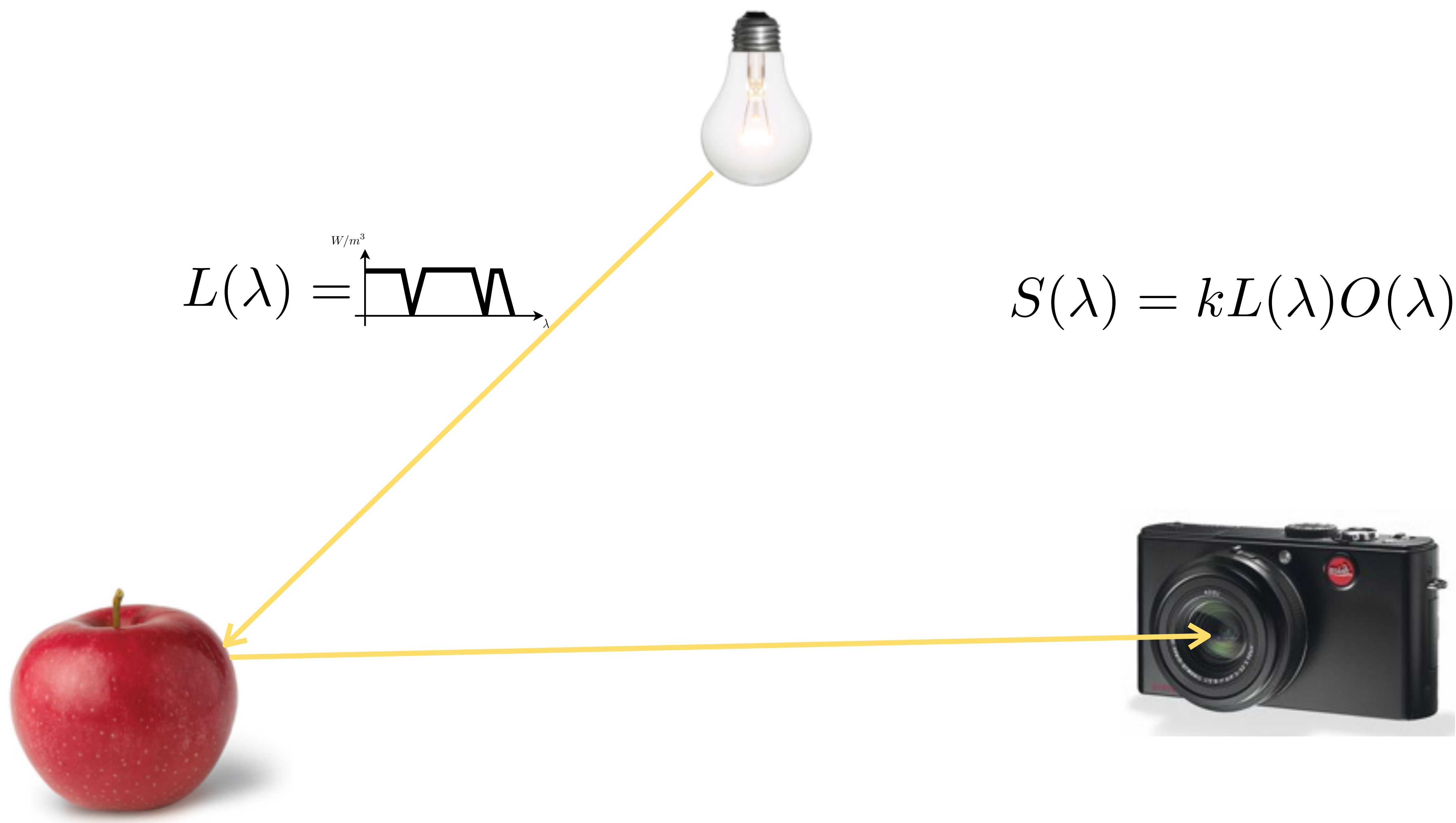
$$S(\lambda) = kL(\lambda)O(\lambda)$$

Basic Image Formation Model

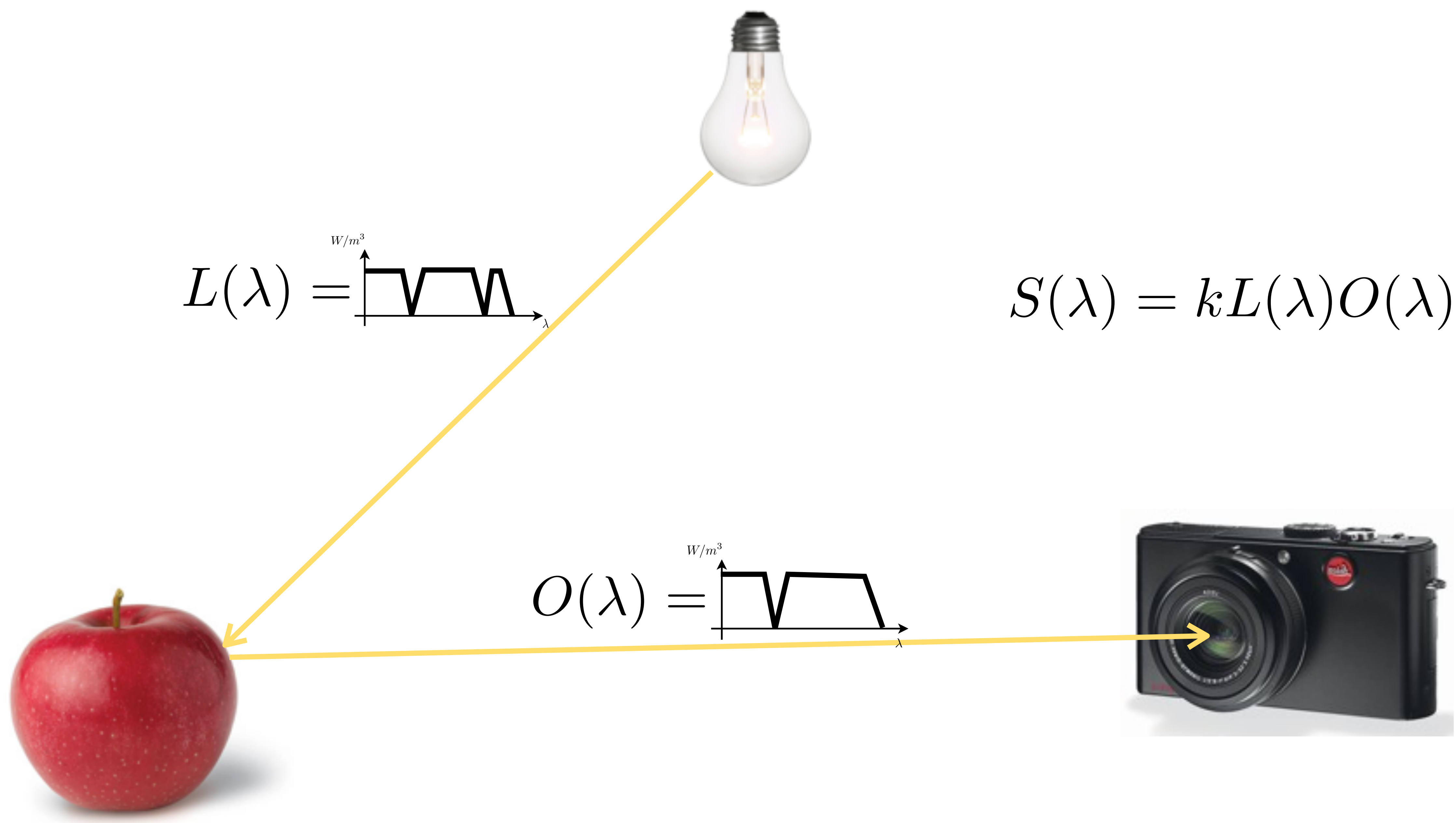


$$S(\lambda) = kL(\lambda)O(\lambda)$$

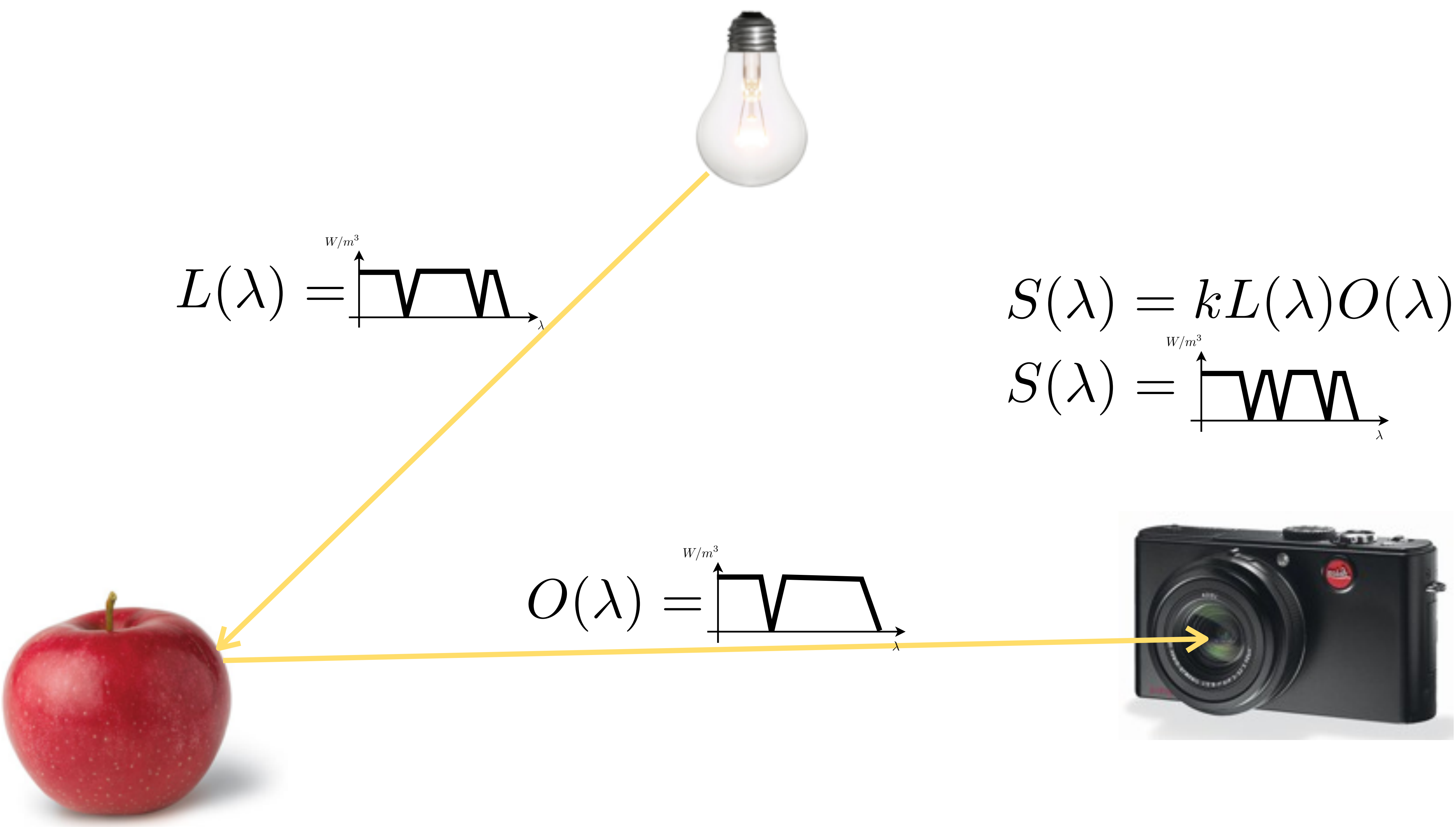
Basic Image Formation Model

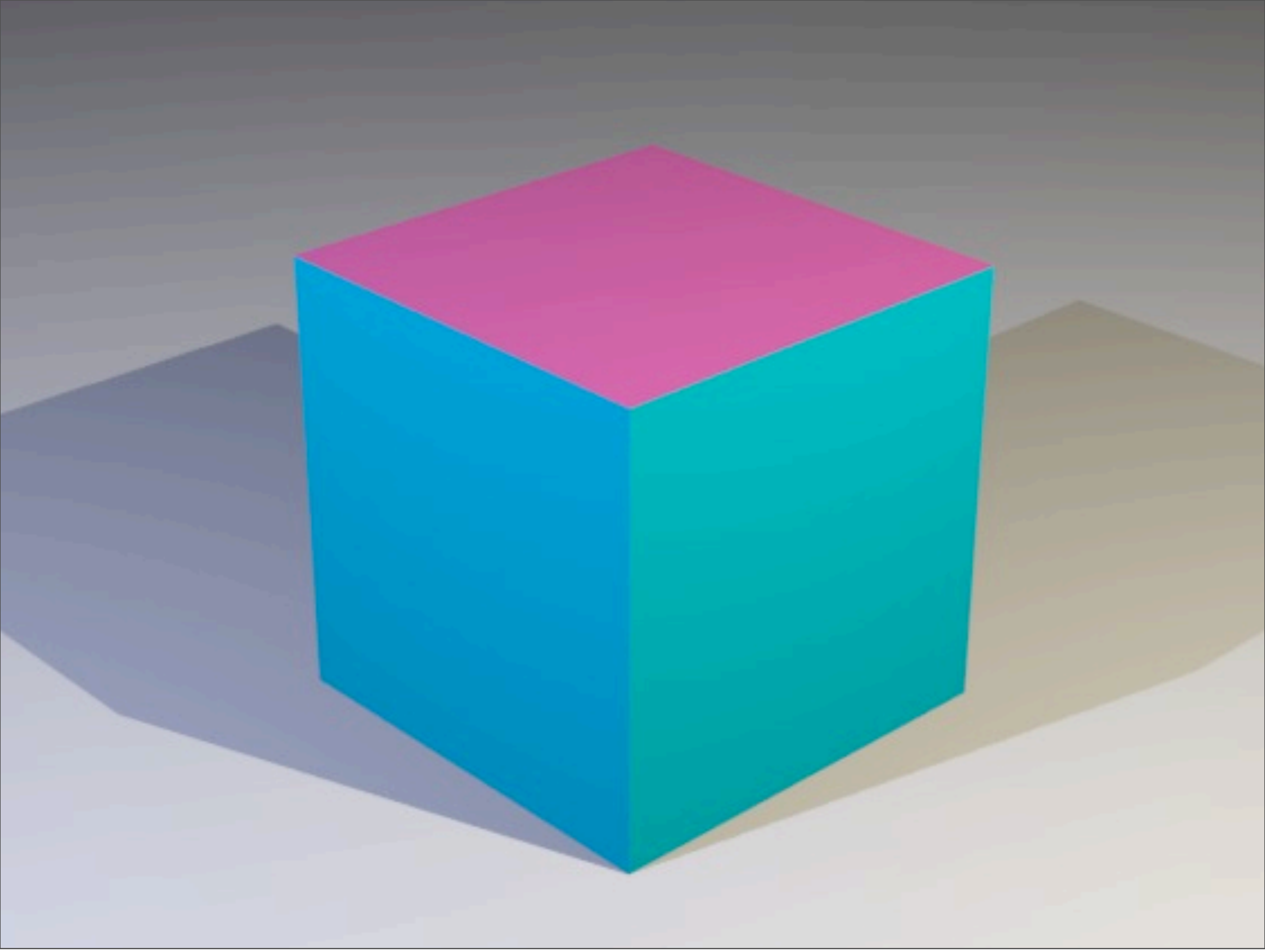


Basic Image Formation Model



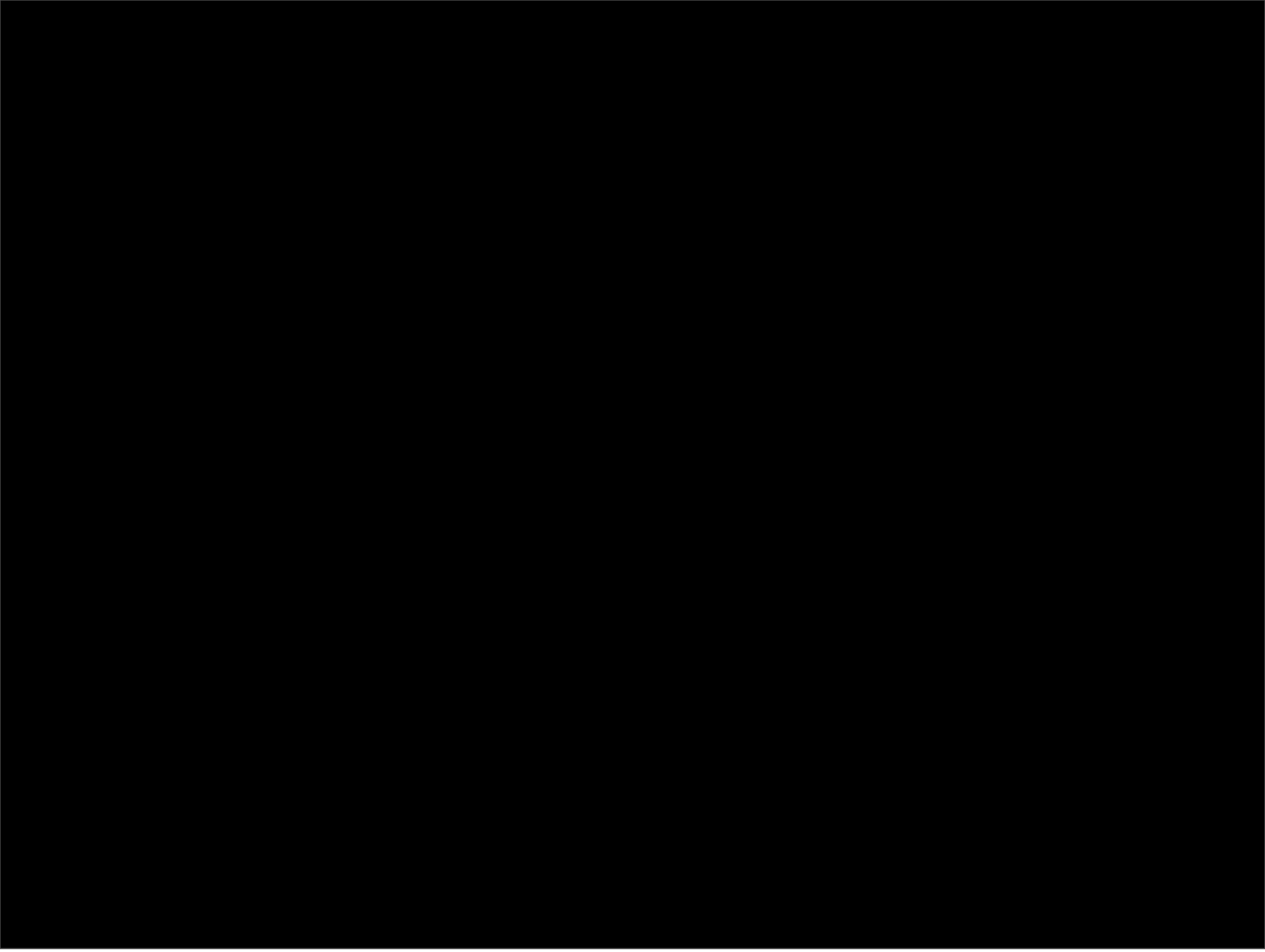
Basic Image Formation Model



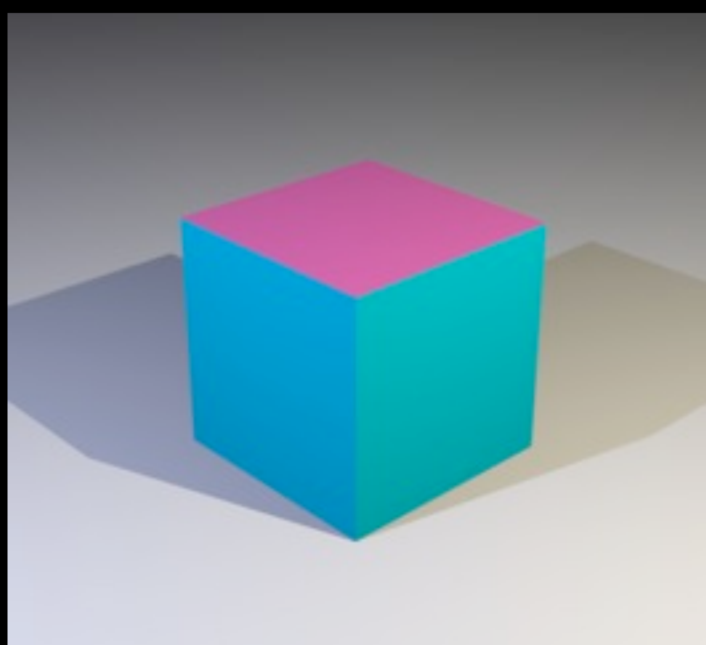


Thursday, September 29, 11

Observe this image. We have here two colored lights, a blue light on the left side and a yellow light on right side, as can be seen in the shadow colors. There are three material colors, blue on the cube sides, magenta on the cube top, and white on the ground. Light color interacts with the material colors to produce a much wider range of colors than the 3 material colors and 2 light colors.

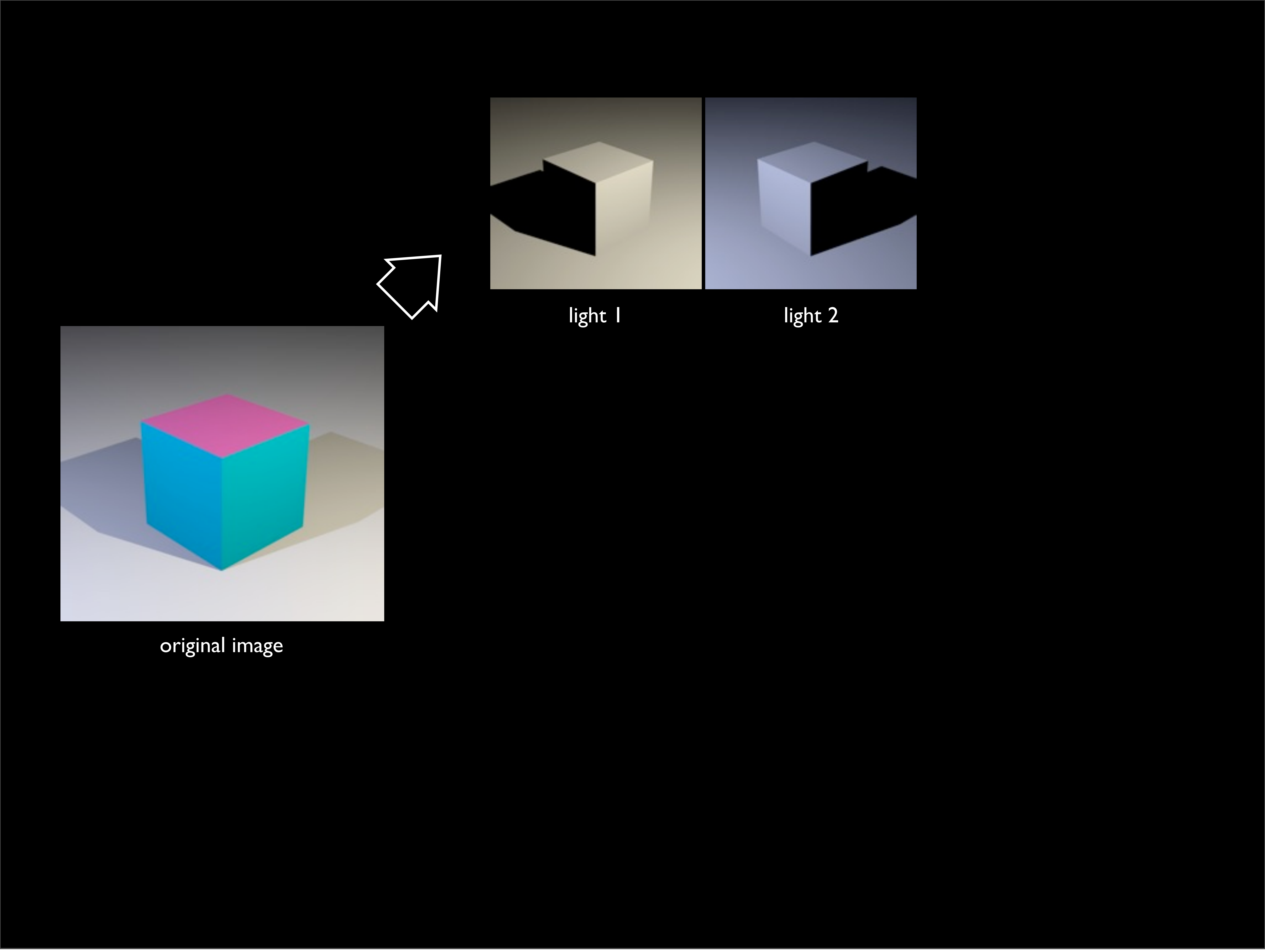


Taking this image and going back to the basic image formation model we can do two things
1) Determine the light color/intensity for each pixel
2) Determine the material color for each pixel
1.1) If we succeed in obtaining the first a neat application is to modify color/intensity of the lights, perhaps making them all white.
2.1) In the other case an interesting application is to modify the material colors without modifying the lights.
3.1) White balance, which is the subject of the first paper, deals with the top flow.
3.2) The second paper we present deals with the second flow, Intrinsic images.



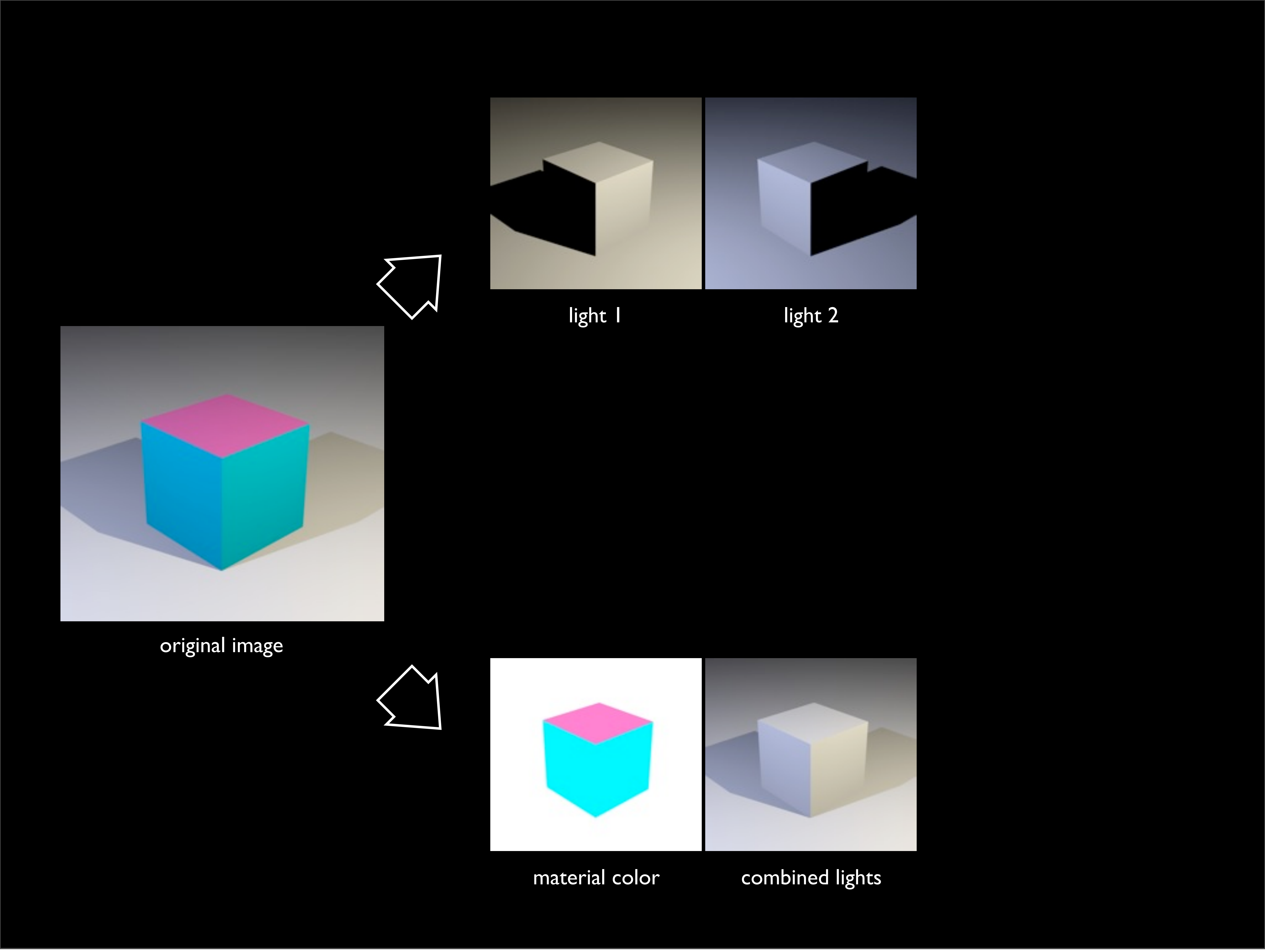
original image

Taking this image and going back to the basic image formation model we can do two things
1) Determine the light color/intensity for each pixel
2) Determine the material color for each pixel
1.1) If we succeed in obtaining the first a neat application is to modify color/intensity of the lights, perhaps making them all white.
2.1) In the other case an interesting application is to modify the material colors without modifying the lights.
3.1) White balance, which is the subject of the first paper, deals with the top flow.
3.2) The second paper we present deals with the second flow, Intrinsic images.



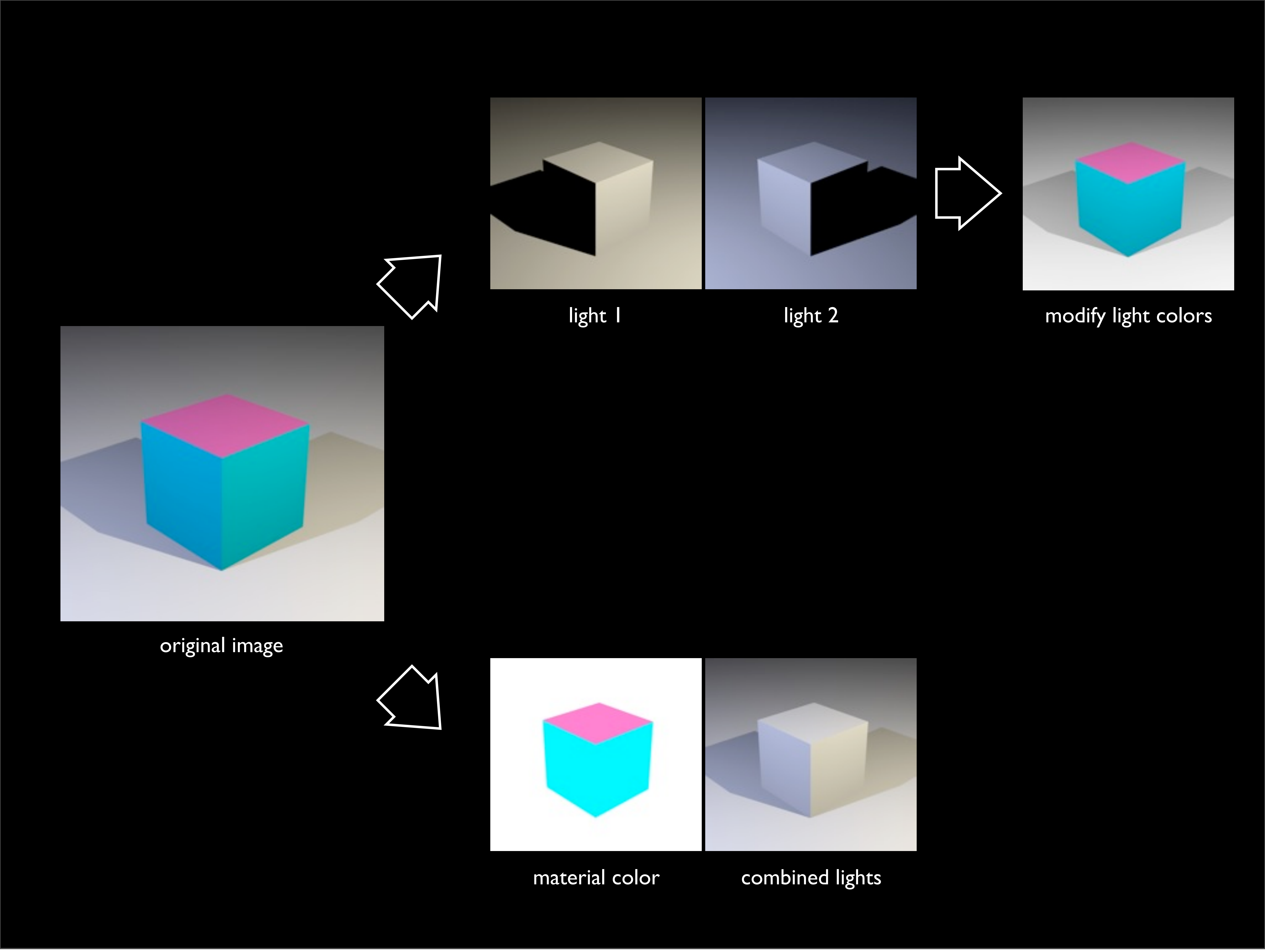
Thursday, September 29, 11

Taking this image and going back to the basic image formation model we can do two things
1) Determine the light color/intensity for each pixel
2) Determine the material color for each pixel
1.1) If we succeed in obtaining the first a neat application is to modify color/intensity of the lights, perhaps making them all white.
2.1) In the other case an interesting application is to modify the material colors without modifying the lights.
3.1) White balance, which is the subject of the first paper, deals with the top flow.
3.2) The second paper we present deals with the second flow, Intrinsic images.



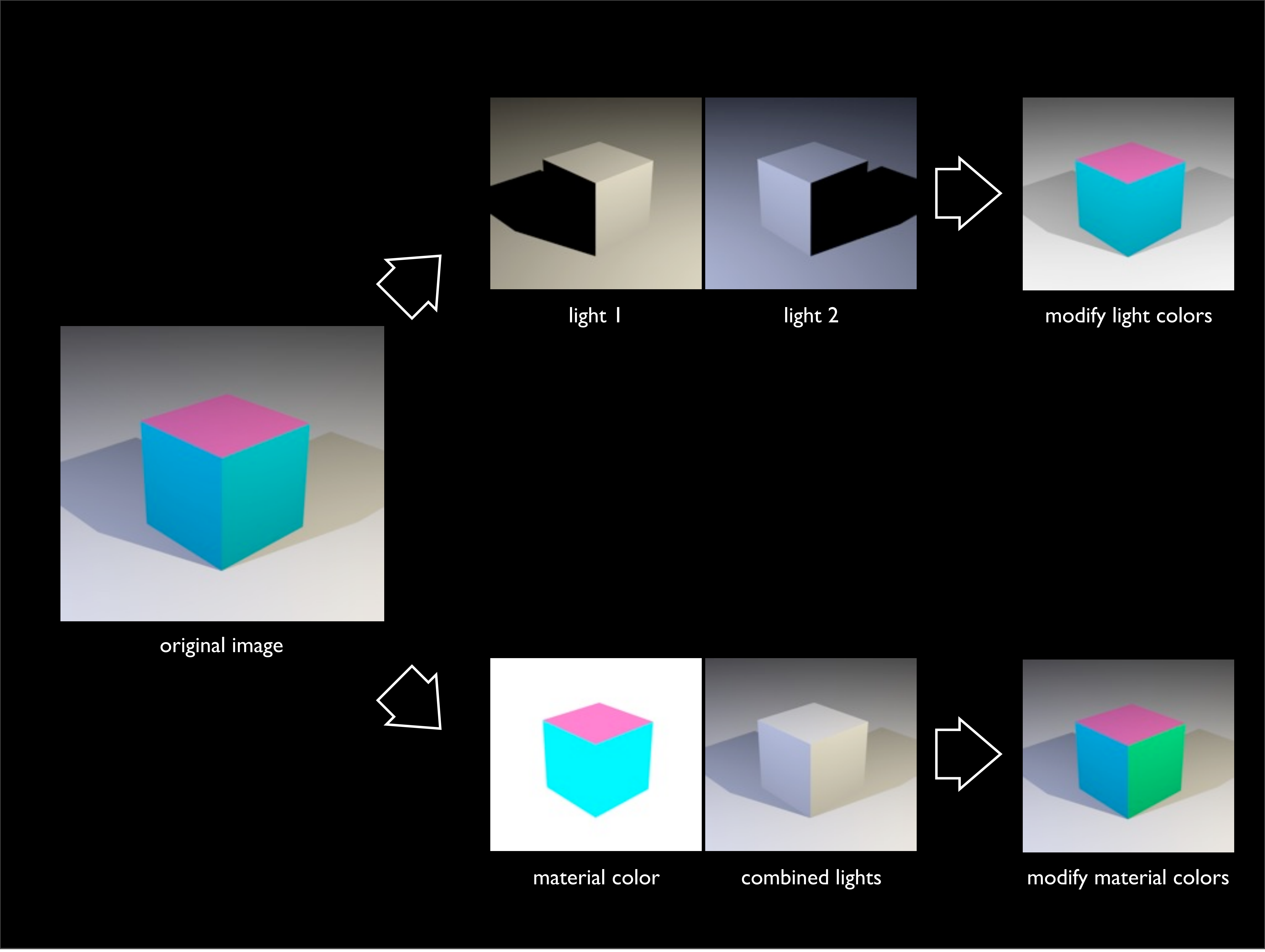
Thursday, September 29, 11

Taking this image and going back to the basic image formation model we can do two things
1) Determine the light color/intensity for each pixel
2) Determine the material color for each pixel
1.1) If we succeed in obtaining the first a neat application is to modify color/intensity of the lights, perhaps making them all white.
2.1) In the other case an interesting application is to modify the material colors without modifying the lights.
3.1) White balance, which is the subject of the first paper, deals with the top flow.
3.2) The second paper we present deals with the second flow, Intrinsic images.



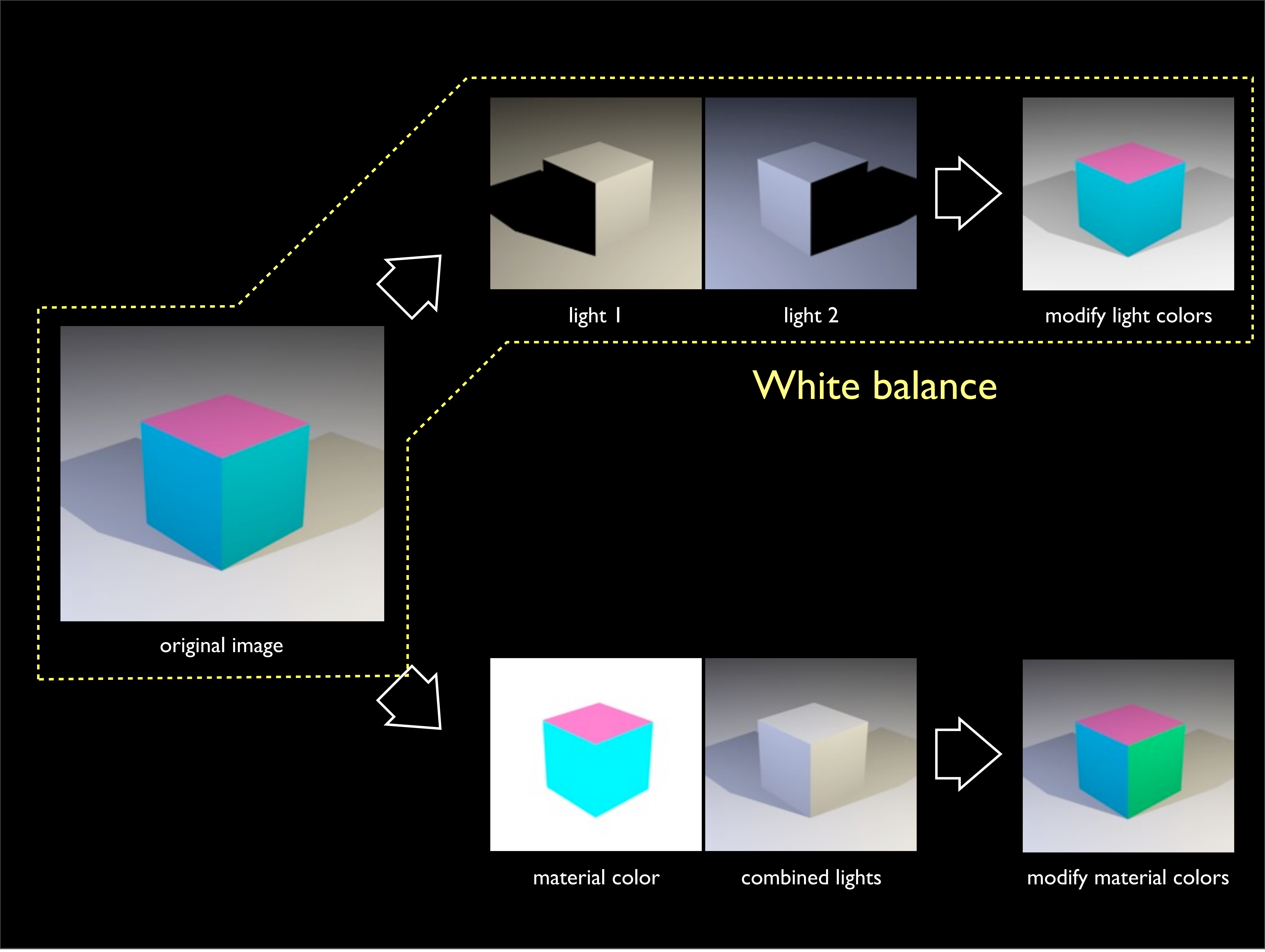
Thursday, September 29, 11

Taking this image and going back to the basic image formation model we can do two things
1) Determine the light color/intensity for each pixel
2) Determine the material color for each pixel
1.1) If we succeed in obtaining the first a neat application is to modify color/intensity of the lights, perhaps making them all white.
2.1) In the other case an interesting application is to modify the material colors without modifying the lights.
3.1) White balance, which is the subject of the first paper, deals with the top flow.
3.2) The second paper we present deals with the second flow, Intrinsic images.



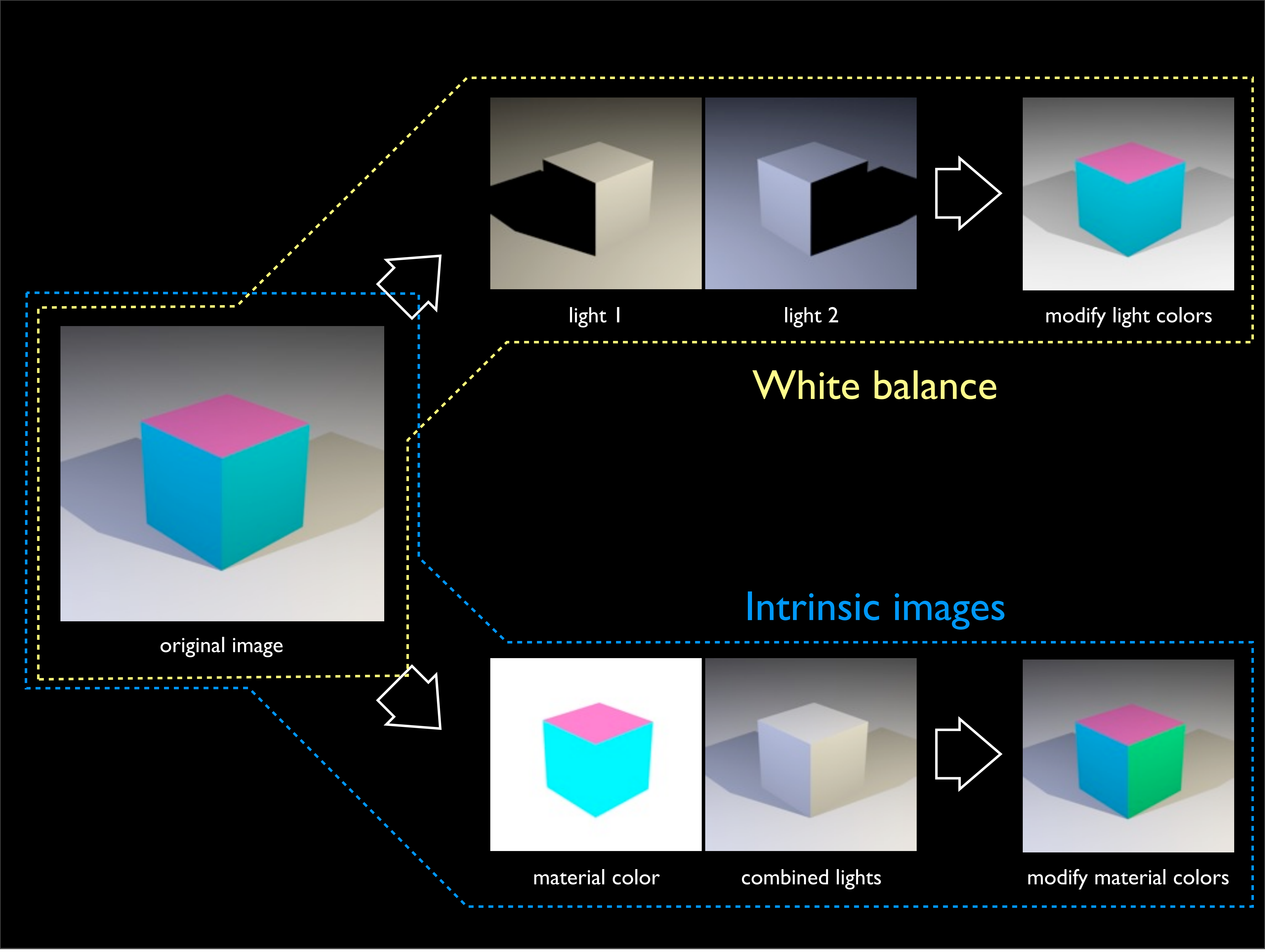
Thursday, September 29, 11

Taking this image and going back to the basic image formation model we can do two things
1) Determine the light color/intensity for each pixel
2) Determine the material color for each pixel
1.1) If we succeed in obtaining the first a neat application is to modify color/intensity of the lights, perhaps making them all white.
2.1) In the other case an interesting application is to modify the material colors without modifying the lights.
3.1) White balance, which is the subject of the first paper, deals with the top flow.
3.2) The second paper we present deals with the second flow, Intrinsic images.



Thursday, September 29, 11

Taking this image and going back to the basic image formation model we can do two things
1) Determine the light color/intensity for each pixel
2) Determine the material color for each pixel
1.1) If we succeed in obtaining the first a neat application is to modify color/intensity of the lights, perhaps making them all white.
2.1) In the other case an interesting application is to modify the material colors without modifying the lights.
3.1) White balance, which is the subject of the first paper, deals with the top flow.
3.2) The second paper we present deals with the second flow, Intrinsic images.



Thursday, September 29, 11

Taking this image and going back to the basic image formation model we can do two things
1) Determine the light color/intensity for each pixel
2) Determine the material color for each pixel
1.1) If we succeed in obtaining the first a neat application is to modify color/intensity of the lights, perhaps making them all white.
2.1) In the other case an interesting application is to modify the material colors without modifying the lights.
3.1) White balance, which is the subject of the first paper, deals with the top flow.
3.2) The second paper we present deals with the second flow, Intrinsic images.

Light Mixture Estimation for Spatially Varying White Balance

Eugene Hsu
Tom Mertens
Sylvain Paris
Shai Avidan
Frédo Durand

http://www.squicky.org/work/sig08lme/

Thursday, September 29, 11

Hey everyone! What's up? Wow, what a great audience!

Thursday, September 29, 11

Digital photographers use the white balance function to remove unsightly color casts due to lighting. This photo, for instance, has a strong orange color cast because it’s taken under tungsten illumination.

Thursday, September 29, 11

We can use white balance to compensate for these conditions, as shown here.

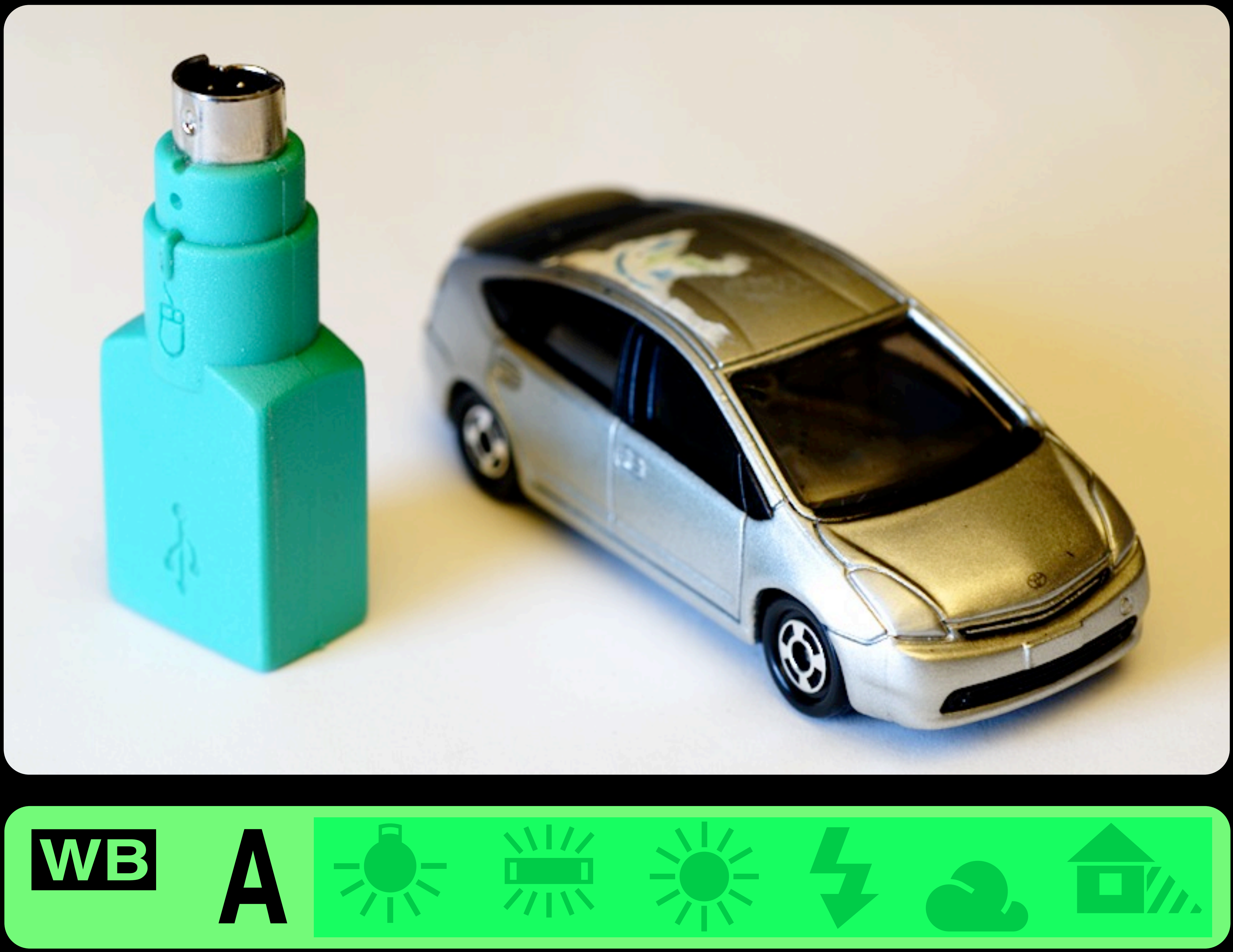
Now, this sort of functionality is almost universal in digital cameras and photo editing packages. But sometimes, these traditional white balance techniques are insufficient.

Thursday, September 29, 11

Take a look at this photo, for instance. It has two types of light: daylight (coming from the left) and tungsten (coming from above). Because the lighting is mixed, the photo’s going to look bad whether we white-balance for daylight (as shown here) …

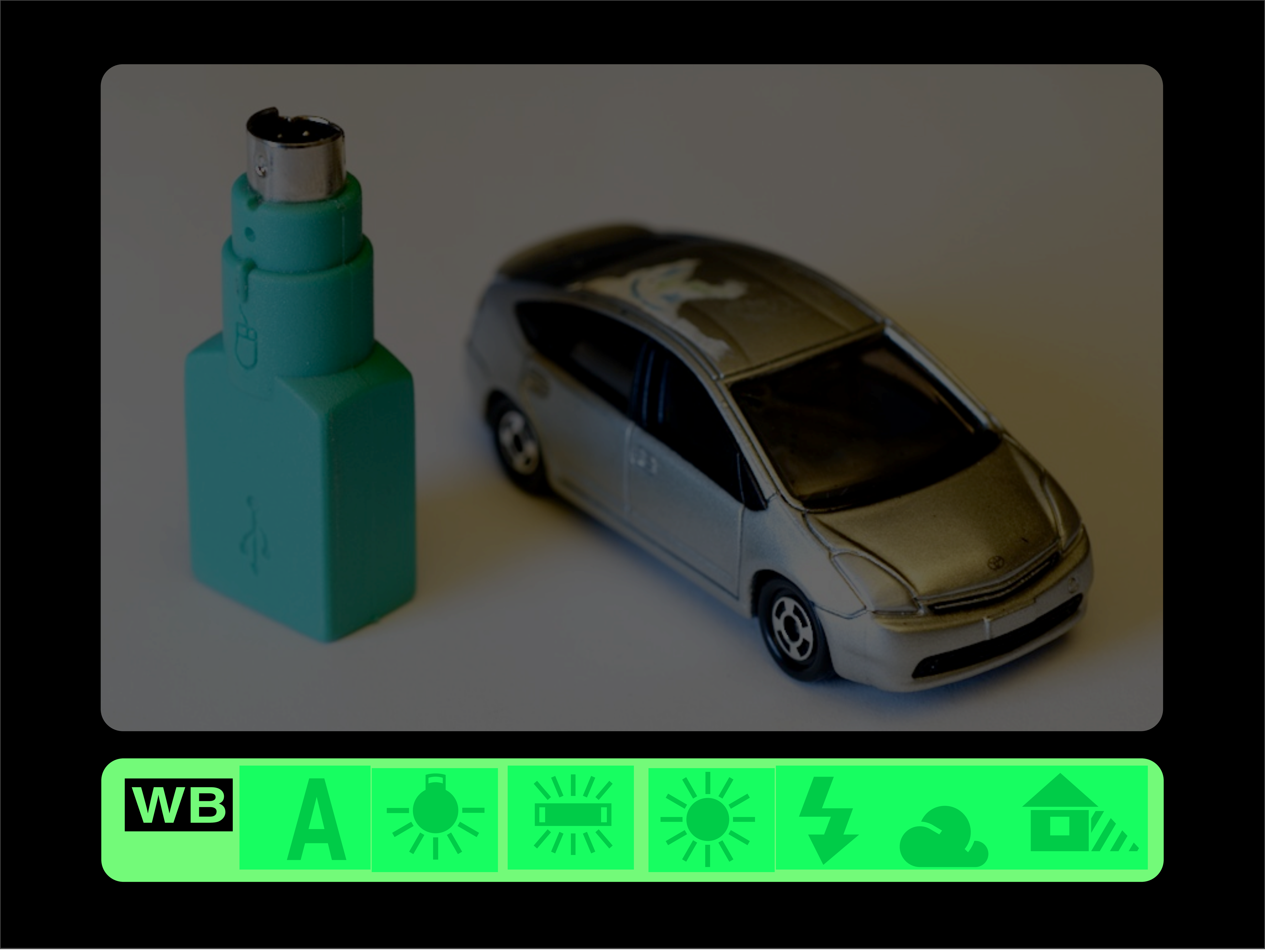
Thursday, September 29, 11

… or tungsten (as shown here) …



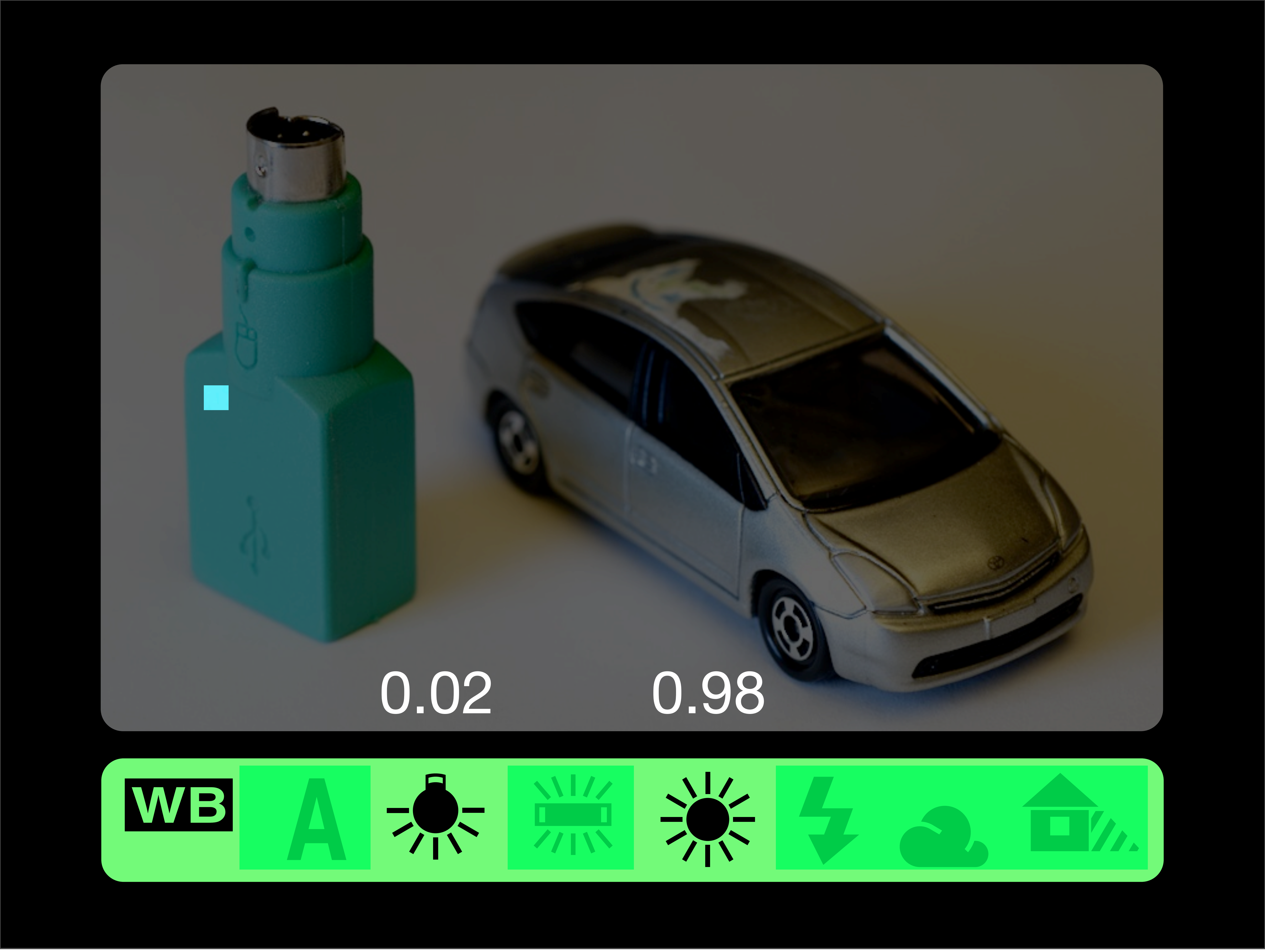
… or something in between. Here, you can see a blue shadow under the car and an orange shadow behind it, although they're both supposed to be gray.

Given this image and the two light types, our technique estimates the spatially varying light mixture at each pixel.



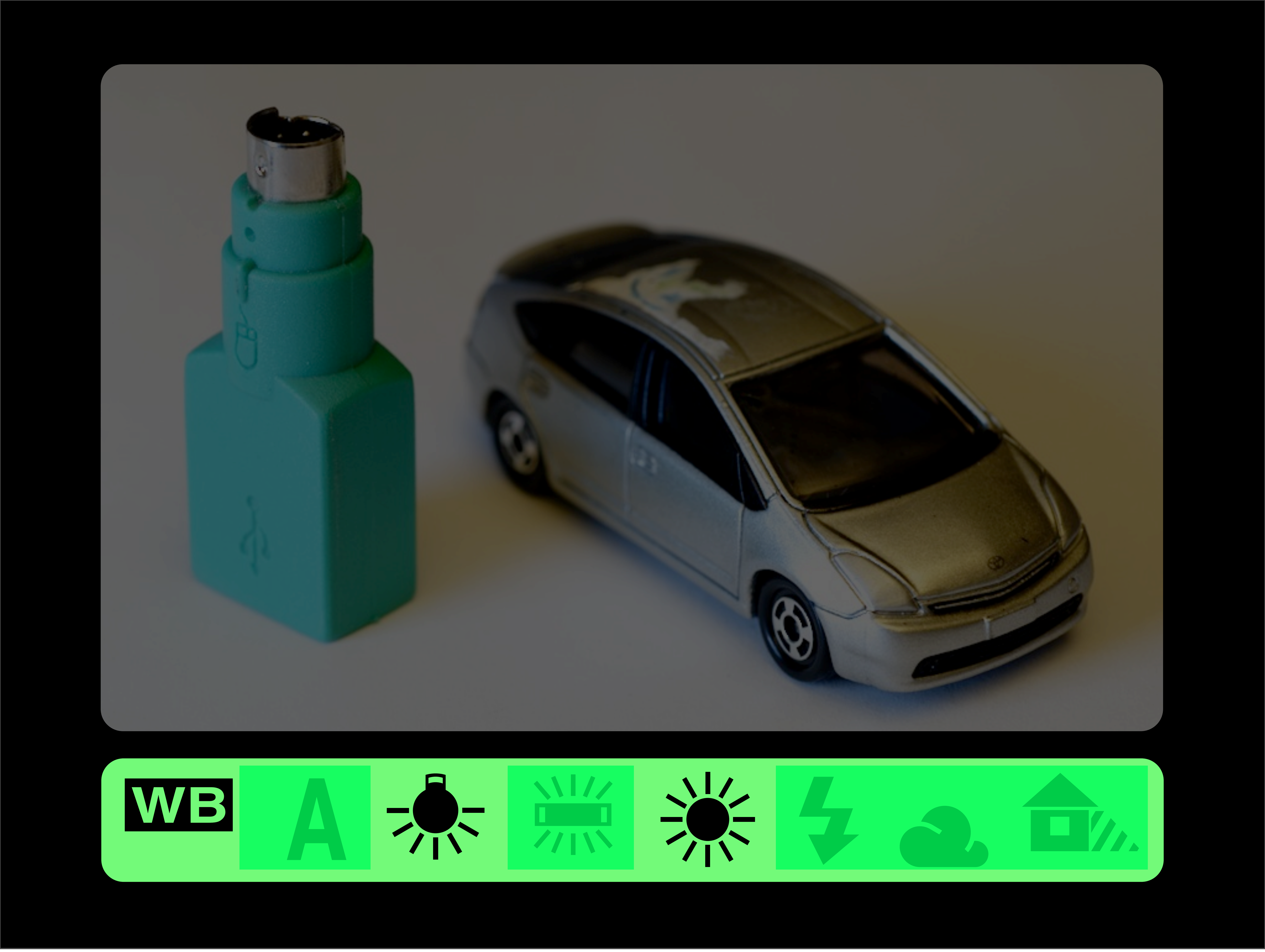
Thursday, September 29, 11

This pixel, for instance, is 2 percent tungsten and 98 percent daylight.



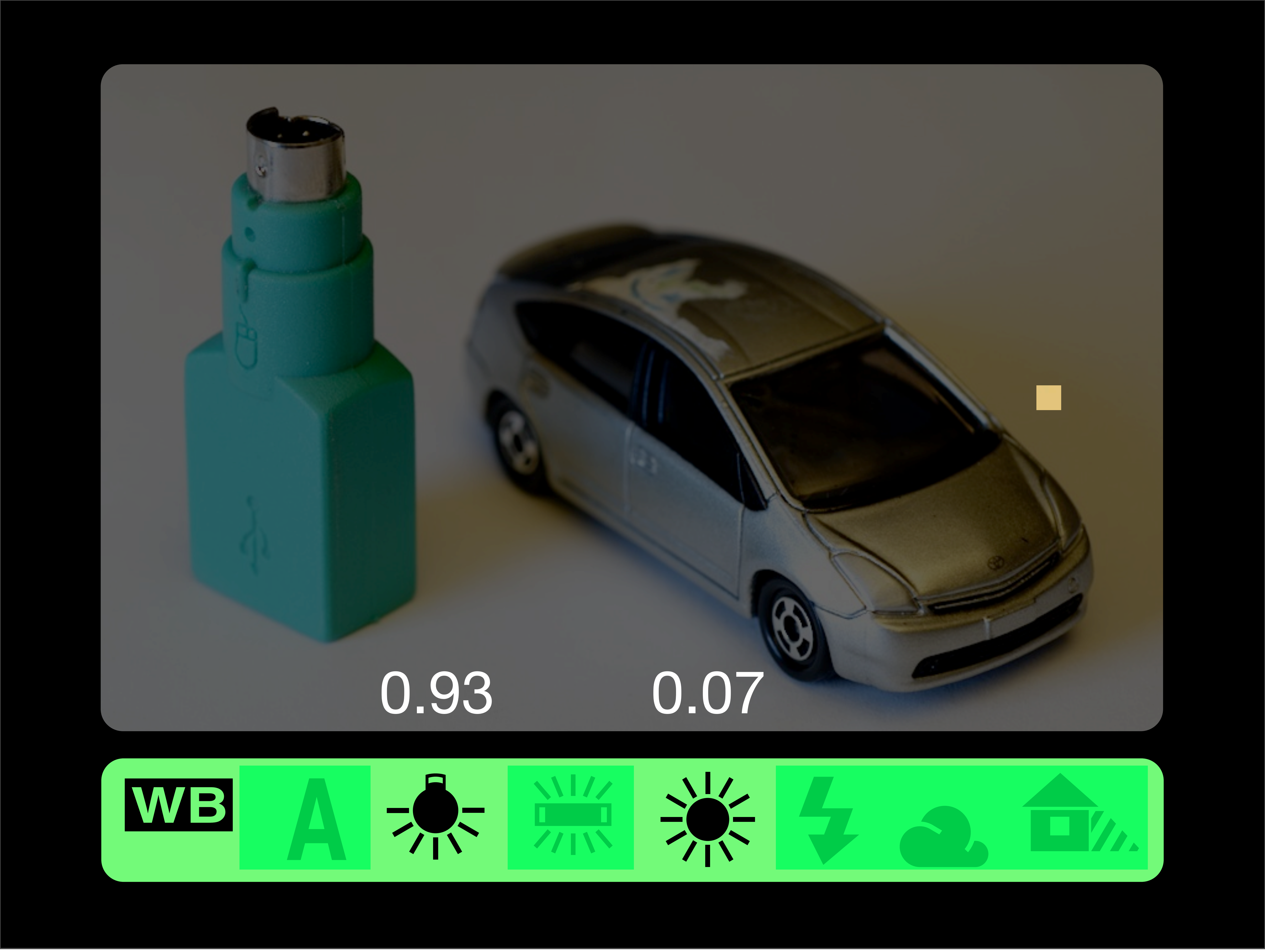
Thursday, September 29, 11

This pixel, for instance, is 2 percent tungsten and 98 percent daylight.



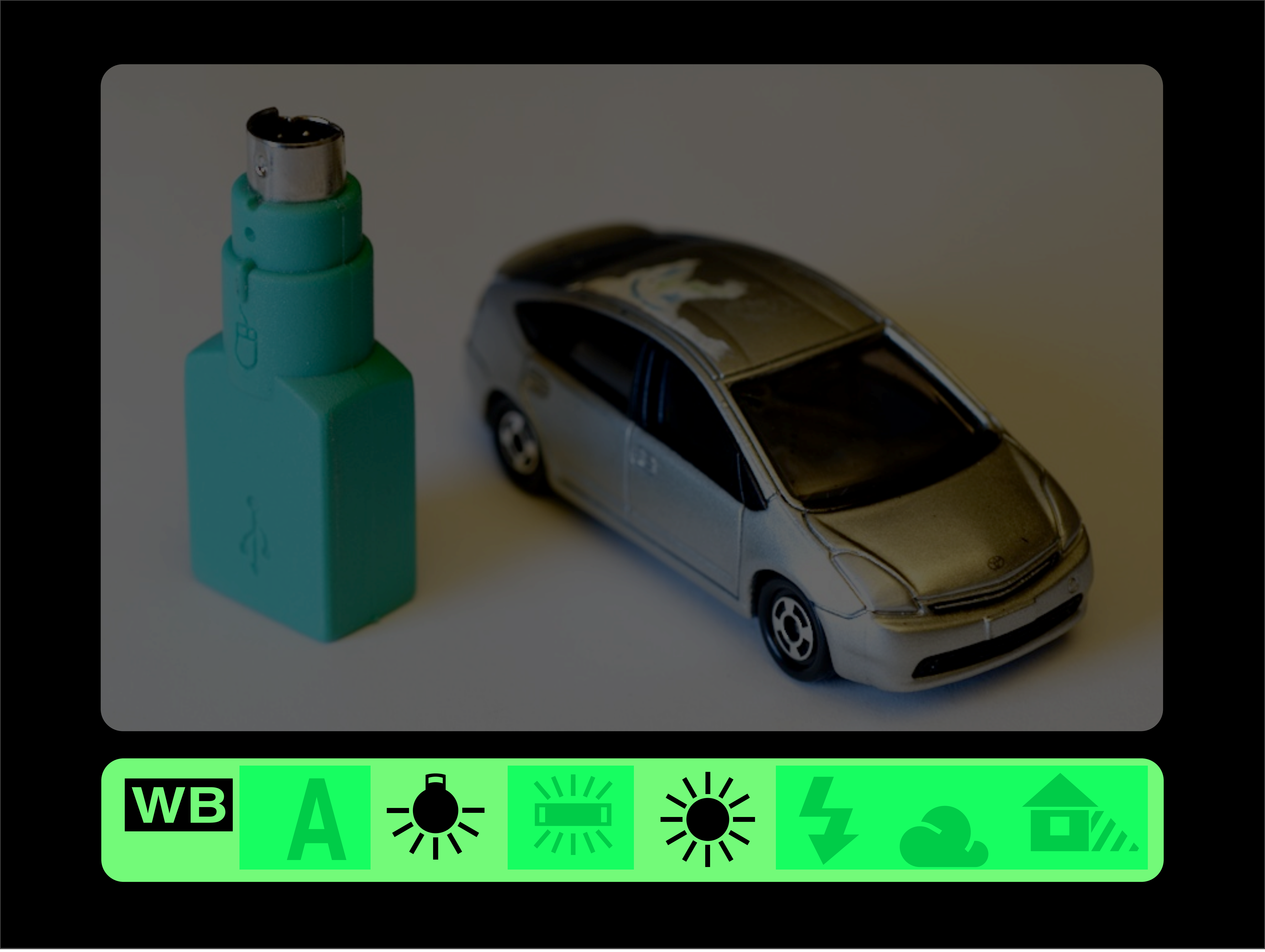
Thursday, September 29, 11

This one is 93 percent tungsten and 7 percent daylight.

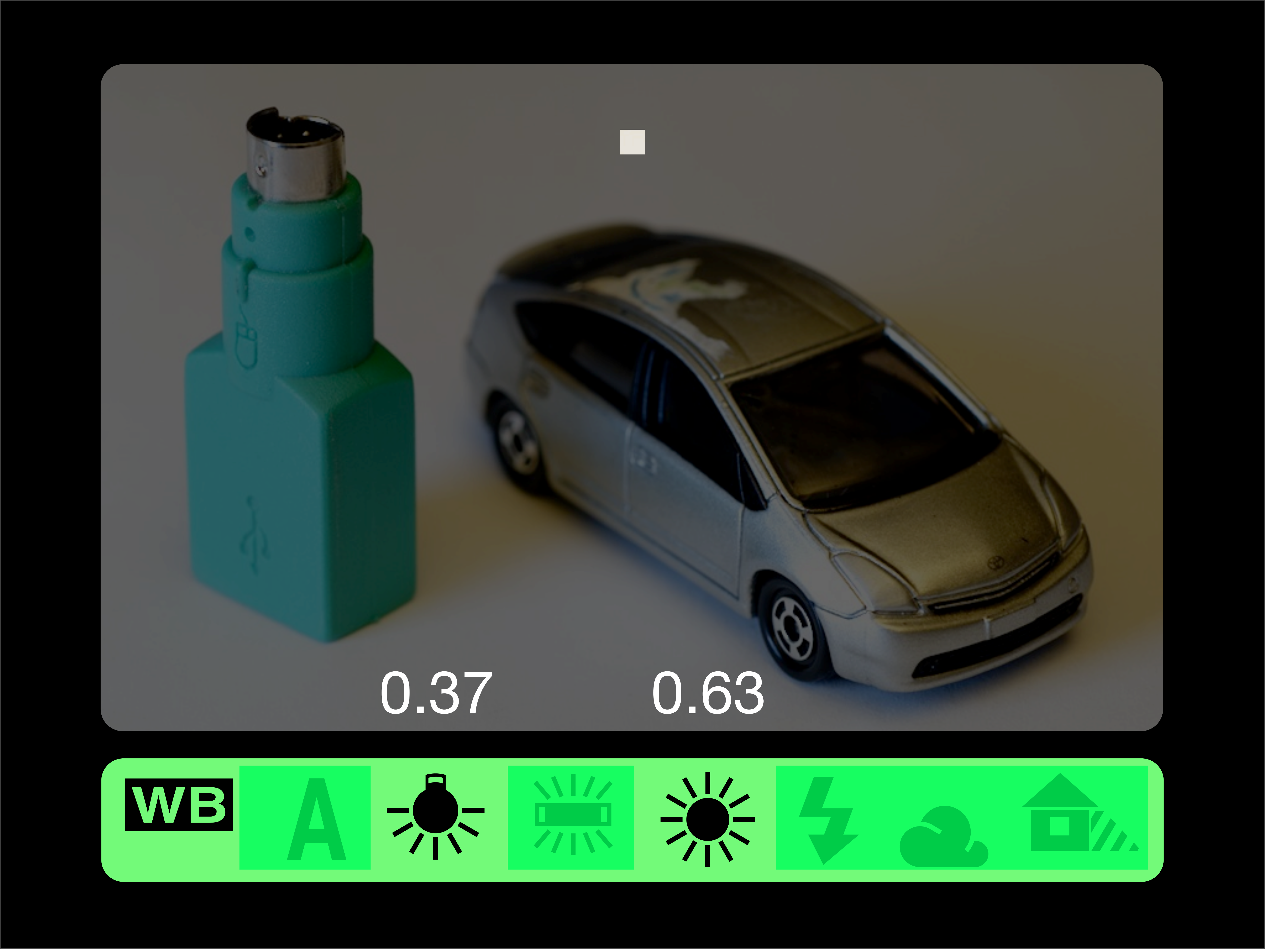


Thursday, September 29, 11

This one is 93 percent tungsten and 7 percent daylight.



And this one is 37 percent tungsten and 63 percent daylight.



Thursday, September 29, 11

And this one is 37 percent tungsten and 63 percent daylight.

By estimating this mixture at every pixel in the image ...

Thursday, September 29, 11

… we can perform white balance. Notice that all the ugly color casts are gone. And later on, we'll also see how this mixture lets us do some interesting relighting effects.

Background

Thursday, September 29, 11

First, I’d like to provide some background about our problem.

Film photographers use color filters during exposure or printing.

Thursday, September 29, 11

White balance has been an issue from the beginnings of color photography.

With film, photographers have to do manual color filtration during exposure or printing. For instance, the magenta filter on the left compensates for greenish fluorescent illumination.

It used to be that photographers needed to carry around a pouch full of these filters to achieve accurate color reproduction.

Digital cameras simplify this process, but they don't handle mixed lighting.

Digital cameras effectively replace that pouch full of filters with a few matrix multiplies. They also provide automatic white balance functions that estimate the type of light in the scene.

The thing about all of these techniques is this: they only assume a single type of light — and as we've seen before, they don't work if there are multiple types of light.

For mixed lighting, filter each source to emulate single illuminant case.

The typical way to handle mixed lighting is fairly primitive. Photographers just try to filter all the light sources to match each other, and then apply standard white balance.

So, for instance, the flash unit on the left has a green filter so that it matches the color temperature of fluorescent lighting. This is what you might use if you're taking flash photos in an office environment.

Now, as you can imagine, this can be a very cumbersome process. The goal of our work is to make it easier.

Barnard [1997]
… smooth lighting

Ebner [2004]
… locally neutral

Thursday, September 29, 11

There have been several papers in graphics and vision that have attempted to solve our problem.

One of the first ones is by Barnard and colleagues in 1997. Their work assumes that lighting variation in the scene is all smooth. Unfortunately, this doesn't hold for a lot of real-world scenes because of geometric discontinuities, hard shadows, and so on.

Ebner's work from 2004 assumes that the scene is locally neutral in color. This technique works well in some cases, but it has a hard time handling scenes with large regions of color.

Kawakami [2005]
… hard shadows

Lischinski [2006]
… user scribbles

Kawakami and colleagues presented a technique in 2005 for “large objects in outdoor scenes. They rely on a number of assumptions to achieve their goal — the main one being that all shadows are hard. Now, this is a pretty restrictive assumption that doesn’t usually hold indoors.

And finally, there’s a number of techniques that ask for user interaction. In particular, Lischinski’s work from SIGGRAPH 2006 uses scribbles for tonal adjustments. Of course, there are also commercial tools like Photoshop that can more or less handle the task given enough user assistance. In contrast, our technique aims for greater automation.

Ultimately, the differences in these techniques all boil down to the assumptions that they make …

Overview

… and there are two key assumptions that differentiate our technique from the others. So I'd like to start our overview with those assumptions.

The input is a scene illuminated by two light types given by the user.

Thursday, September 29, 11

We assume that we're given a single photo illuminated by two light types given by the user. For instance, this scene has mixed tungsten and daylight illumination.

This two-light assumption covers a lot of typical scenarios. For instance, indoor pictures often show a mix of interior light and daylight from the windows. And flash pictures often show a mix of flash and ambient lighting.

Thursday, September 29, 11

Our key assumption is that the scene is dominated by a few material colors. This is similar to what Omer and Werman noted in their 2004 paper. This scene, for instance, has a lot of colorful stuff on the shelves, but it's predominantly white and brown.

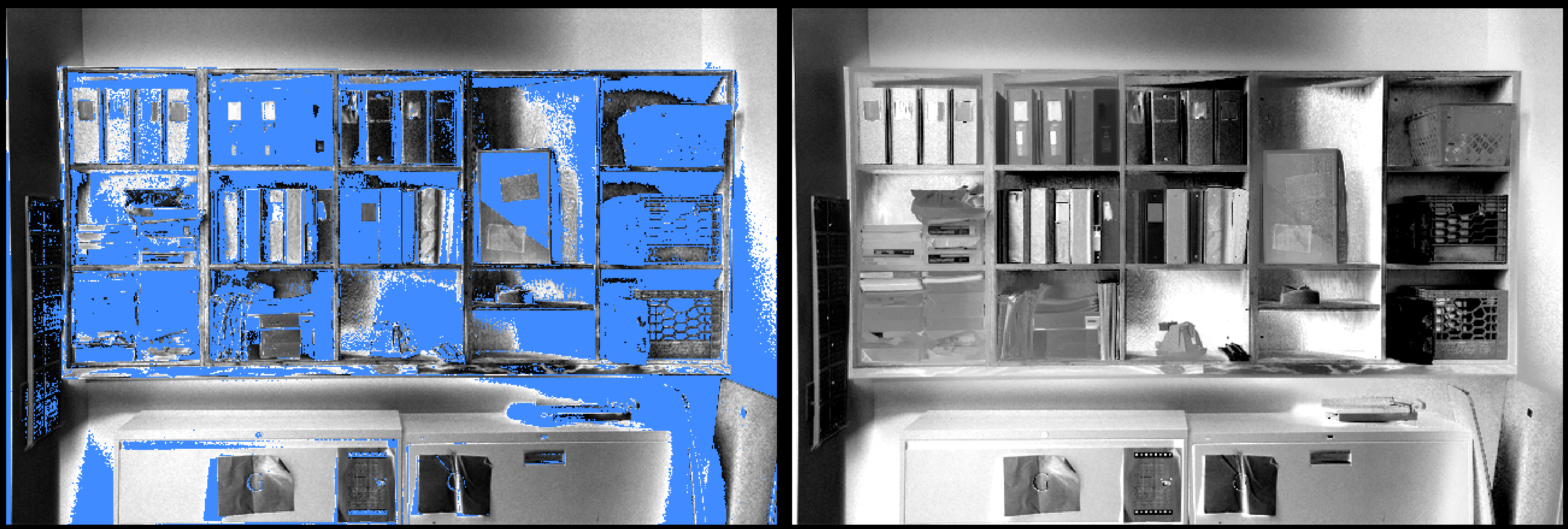
We assume that a few material colors dominate [Omer and Werman 2004].

Thursday, September 29, 11

Our key assumption is that the scene is dominated by a few material colors. This is similar to what Omer and Werman noted in their 2004 paper. This scene, for instance, has a lot of colorful stuff on the shelves, but it’s predominantly white and brown.

We vote on these dominant material colors and label pixels accordingly.

Based on this assumption, we use a voting scheme to determine these dominant material colors and label pixels accordingly.



We then estimate the light mixture and interpolate missing values.

Thursday, September 29, 11

These pixels are used to estimate the light mixture. On the left, the bright pixels correspond to more tungsten, the dark pixels correspond to more daylight, and the blue pixels are undetermined. We fill in these missing blue pixels using an interpolation scheme inspired by matting techniques, as shown on the right.

We use the light mixture to achieve spatially varying white balance.

Thursday, September 29, 11

Finally, we use the estimated light mixture to achieve spatially varying white balance. So, let's move on to the details of our technique.

[CHECKPOINT 5:00]

White Balance

Thursday, September 29, 11

Let’s start off with a description of white balance.

$$\begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix} = \begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k \begin{bmatrix} L_R \\ L_G \\ L_B \end{bmatrix}}$$

Observed pixel color is material color multiplied by scaled light color.

Your typical white balance technique assumes that the observed pixel color I equals the material color R multiplied by the light color L scaled by a factor k. This scale factor accounts for things like shadowing, attenuation, and so on. The dot here means element by element multiplication.

In this model, the goal of white balance is to show the scene as if it were taken under neutral illumination. To do this, we define a color transformation W …

$$\begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix} = \begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k \begin{bmatrix} L_R \\ L_G \\ L_B \end{bmatrix}}$$

… that scales each color channel by the inverse of the light source color. And this gives us the image as if it were taken …

$$\begin{bmatrix}W_R\\W_G\\W_B\end{bmatrix} \bullet \begin{bmatrix}I_R\\I_G\\I_B\end{bmatrix} = \begin{bmatrix}R_R\\R_G\\R_B\end{bmatrix} \bullet \boxed{k \begin{bmatrix}L_R\\L_G\\L_B\end{bmatrix}} \div \begin{bmatrix}L_R\\L_G\\L_B\end{bmatrix}$$

Proper white balance is achieved by inverting the light source color.

… that scales each color channel by the inverse of the light source color. And this gives us the image as if it were taken …

$$\begin{bmatrix} W_R \\ W_G \\ W_B \end{bmatrix} \cdot \begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix} = \begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \cdot \boxed{\begin{matrix} k & & \\ & & \\ & & \end{matrix}}$$

Proper white balance is achieved by inverting the light source color.

… under pure white light. And that’s how white balance is usually done.

$$\begin{bmatrix} W_R \\ W_G \\ W_B \end{bmatrix} \bullet \begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix} = \begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}$$

Proper white balance is achieved by inverting the light source color.

… under pure white light. And that's how white balance is usually done.

$$\begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix} = \begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{ k_1 \begin{bmatrix} L_{1R} \\ L_{1G} \\ L_{1B} \end{bmatrix} + k_2 \begin{bmatrix} L_{2R} \\ L_{2G} \\ L_{2B} \end{bmatrix} }$$

Our technique assumes two lights.

Thursday, September 29, 11

Our technique assumes two lights instead of one. But, as you can see here, it's pretty similar to the previous scenario.

$$\begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix} = \begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{\quad k_1 \qquad\qquad + \quad k_2 \qquad}$$

Again, the goal of white balance is to show the scene as if it were taken under neutral illumination. And like before, we can do this by scaling each pixel by a vector W …

$$\begin{bmatrix} W_R \\ W_G \\ W_B \end{bmatrix} \bullet \begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix} = \begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + k_2 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}$$

The goal of white balance is to show the scene under neutral illumination.

Again, the goal of white balance is to show the scene as if it were taken under neutral illumination. And like before, we can do this by scaling each pixel by a vector W ...

$$\begin{matrix} W_R \\ W_G \\ W_B \end{matrix}$$

But this time, W is defined as one divided by some mixture alpha of the two lights. Here, alpha is just k1 divided by k1 plus k2. And this is the key unknown — once we estimate alpha, we can perform white balance.

$$\begin{bmatrix} W_R \\ W_G \\ W_B \end{bmatrix} = 1 \div \boxed{\alpha \begin{bmatrix} L_{1R} \\ L_{1G} \\ L_{1B} \end{bmatrix} + (1-\alpha) \begin{bmatrix} L_{2R} \\ L_{2G} \\ L_{2B} \end{bmatrix}}$$

$$\alpha = k_1 \div \boxed{k_1 + k_2}$$

Proper white balance is defined by the relative mixture of the lights.

But this time, W is defined as one divided by some mixture alpha of the two lights. Here, alpha is just k1 divided by k1 plus k2. And this is the key unknown — once we estimate alpha, we can perform white balance.

$$\alpha = k_1 \div \boxed{k_1 + k_2}$$

Thursday, September 29, 11

Unfortunately, solving for alpha is a pretty underconstrained problem. Since we aren't given the material color R, the two-light image model has fewer equations than unknowns. And this difficulty is compounded by the fact that we have to solve for different alphas …

$$\begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix} = \begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k_1 \begin{bmatrix} L_{1R} \\ L_{1G} \\ L_{1B} \end{bmatrix} + k_2 \begin{bmatrix} L_{2R} \\ L_{2G} \\ L_{2B} \end{bmatrix}}$$

$$\alpha = k_1 \div \boxed{k_1 + k_2}$$

Solving for α is underconstrained since the material color is not given.

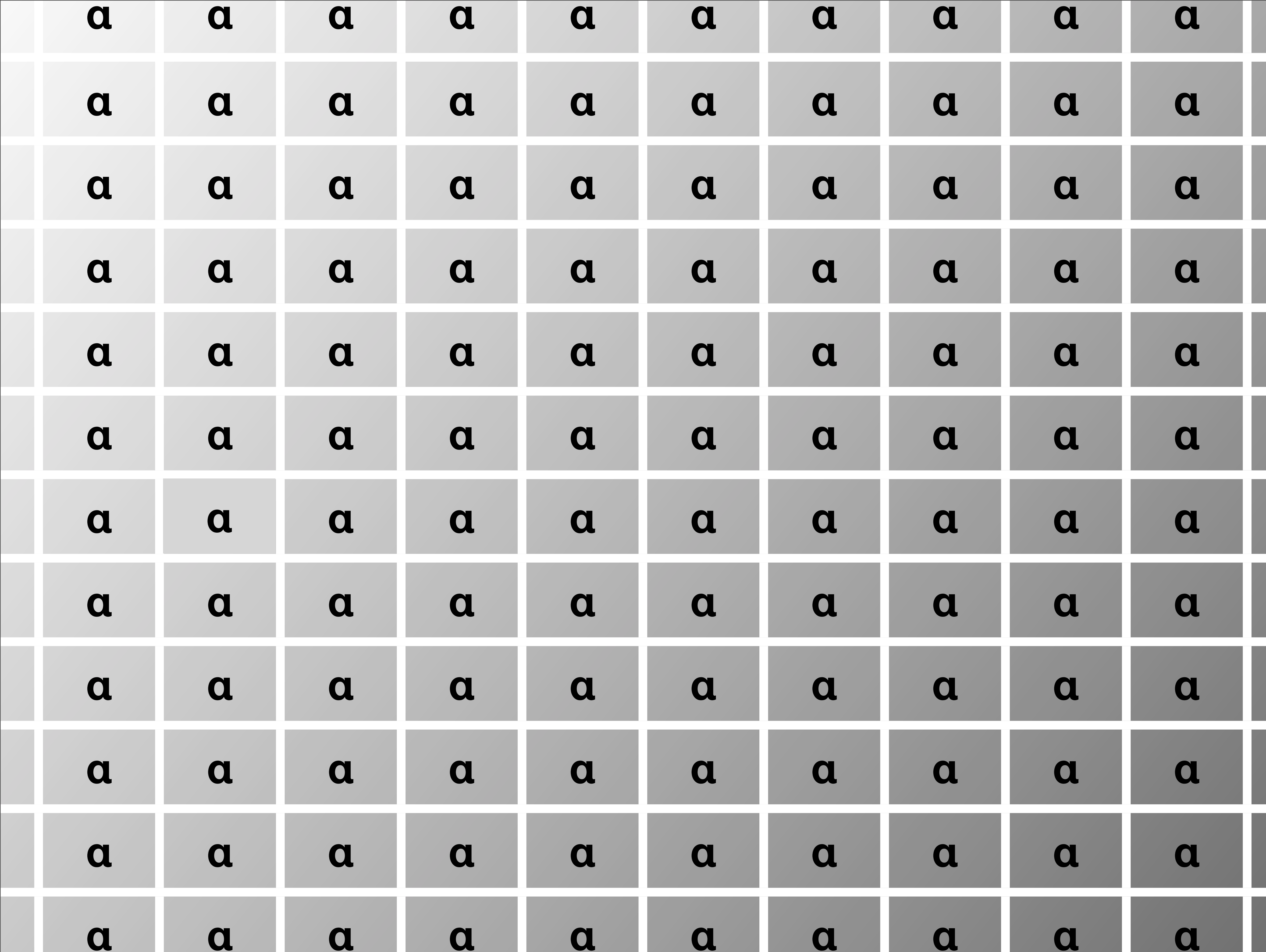
Unfortunately, solving for alpha is a pretty underconstrained problem. Since we aren't given the material color R, the two-light image model has fewer equations than unknowns. And this difficulty is compounded by the fact that we have to solve for different alphas …

a

Thursday, September 29, 11

… at every pixel of the image.

[CHECKPOINT 7:00]



… at every pixel of the image.

[CHECKPOINT 7:00]

Material Color Estimation

Thursday, September 29, 11

To resolve this issue, we use a voting scheme to estimate material colors in the scene.

Thursday, September 29, 11

The fundamental problem that we have to get around is that individual pixel colors are ambiguous.

Thursday, September 29, 11

The fundamental problem that we have to get around is that individual pixel colors are ambiguous.

For instance, this pixel is obviously not white. Now, it could be a colored object lit by white light, or a white object lit by colored light. It just happens to be the latter, but how can we figure this out?

Thursday, September 29, 11

For instance, this pixel is obviously not white. Now, it could be a colored object lit by white light, or a white object lit by colored light. It just happens to be the latter, but how can we figure this out?

Thursday, September 29, 11

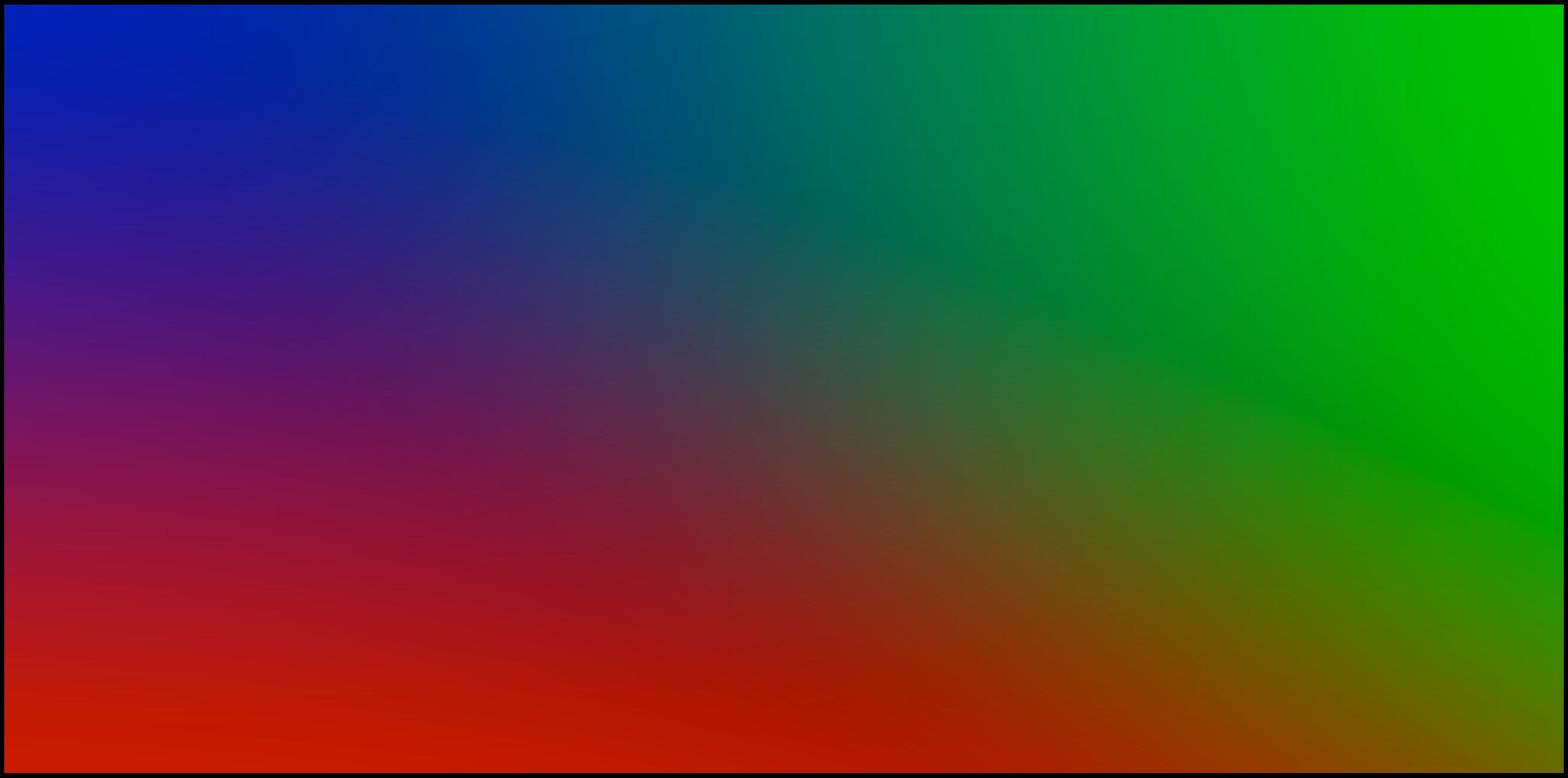
Well, the idea is to look for multiple pixels that share the same material color. Alone, these pixels are ambiguous even when we know the light source colors. But together, they all can be explained by a single material color.

So, the key idea behind our technique is this. We want to identify a small set of dominant material colors that can explain a large number of pixels in the image.

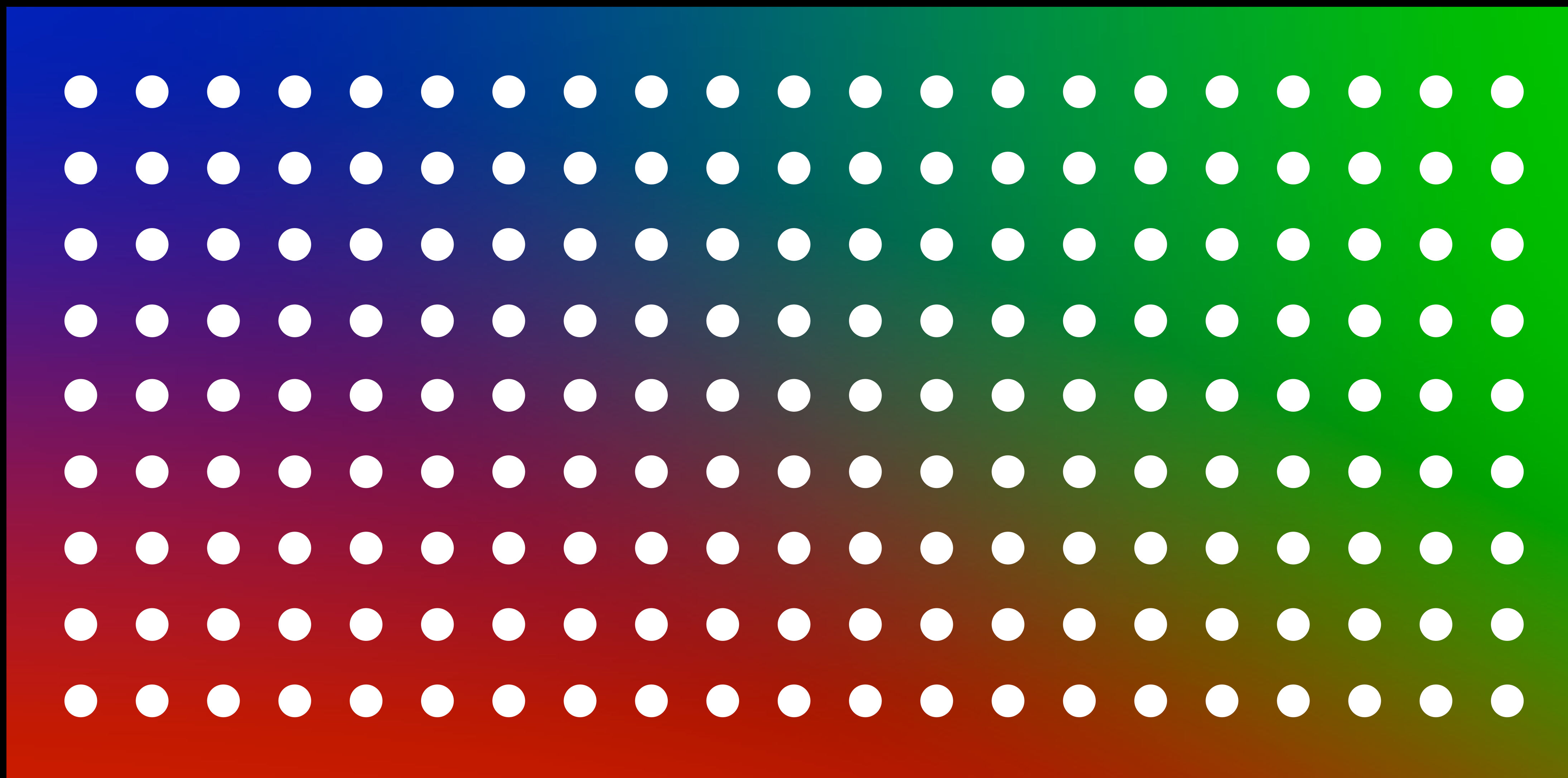
Thursday, September 29, 11

Well, the idea is to look for multiple pixels that share the same material color. Alone, these pixels are ambiguous even when we know the light source colors. But together, they all can be explained by a single material color.

So, the key idea behind our technique is this. We want to identify a small set of dominant material colors that can explain a large number of pixels in the image.



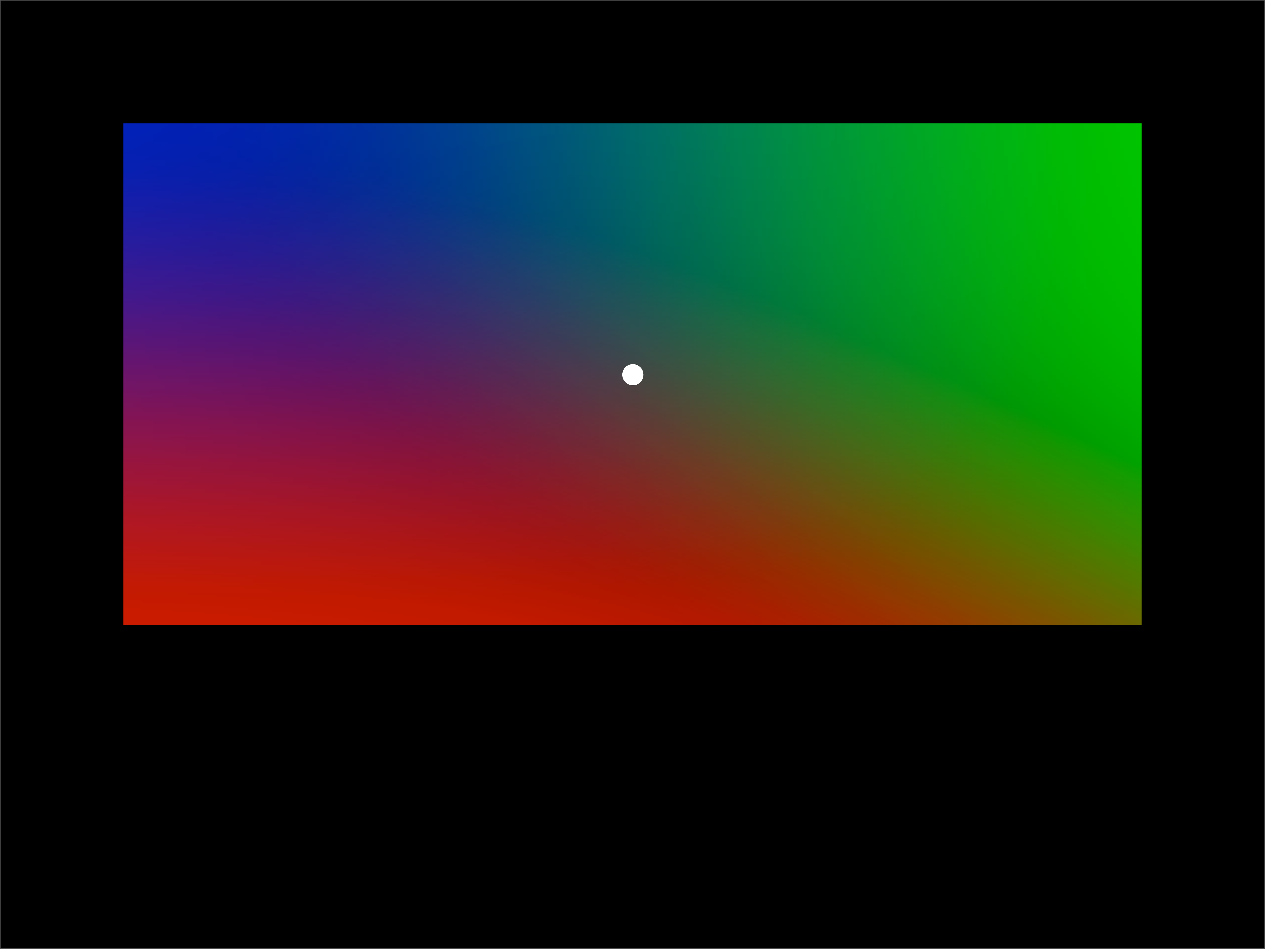
To do this, we start by sampling the set of possible material colors and finding the one that accounts for the most pixels.



Sample material colors and find the one that accounts for the most pixels.

Thursday, September 29, 11

To do this, we start by sampling the set of possible material colors and finding the one that accounts for the most pixels.

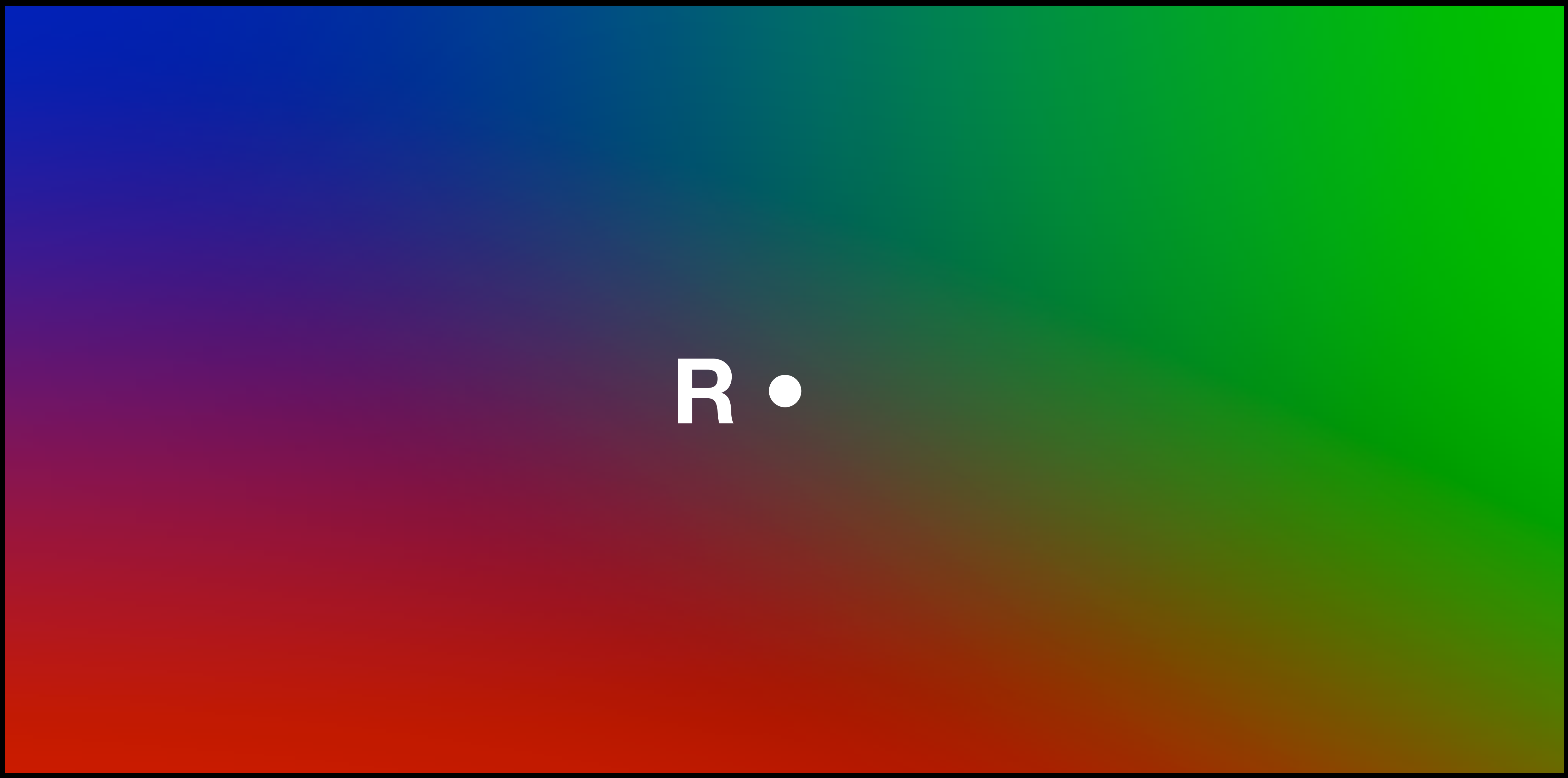


Thursday, September 29, 11

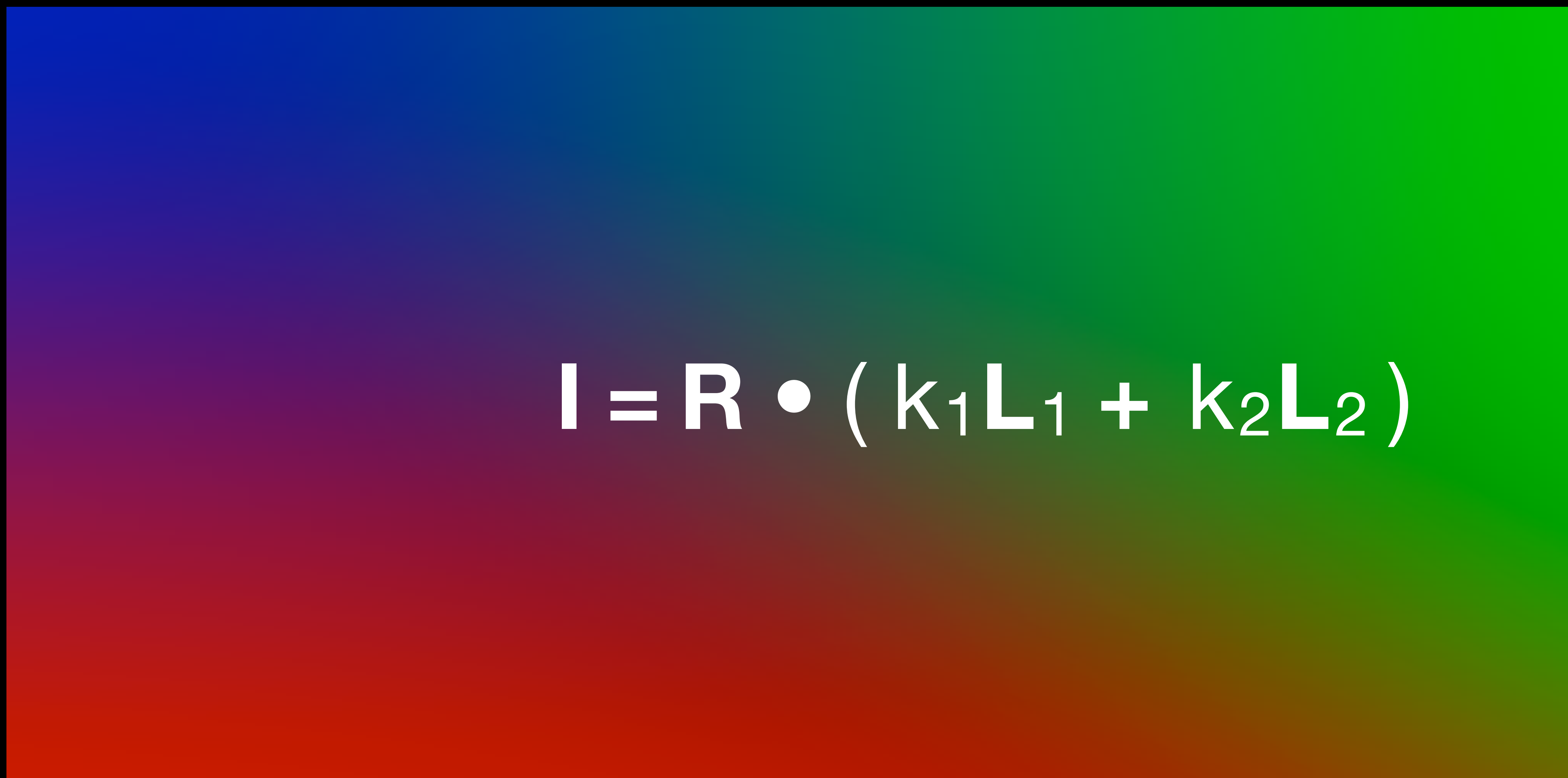
Let’s say that we’re given a candidate material color R.

Given a candidate material color …

Let’s say that we’re given a candidate material color R.



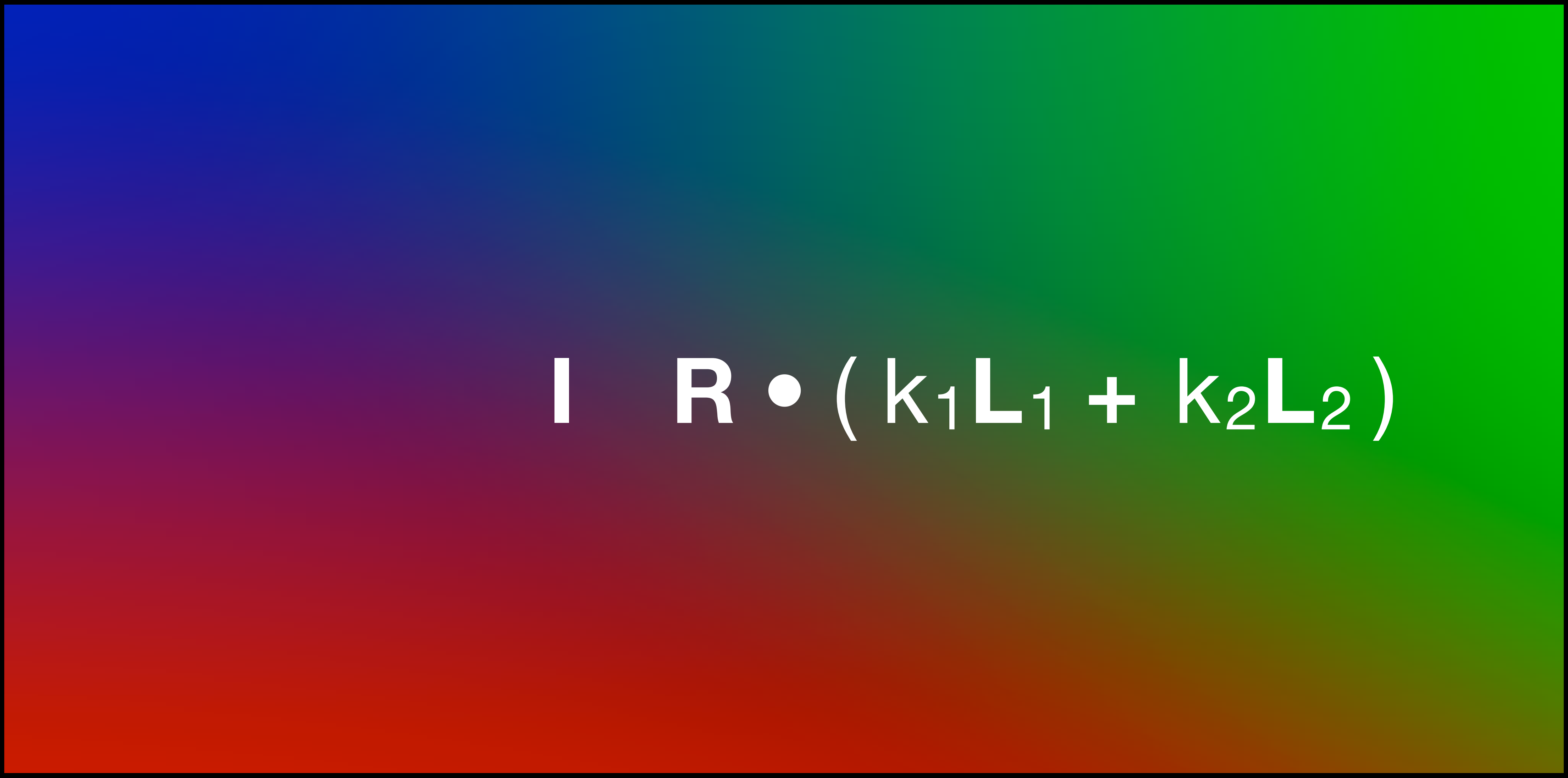
We can use our image model to determine how this material color could appear in the image.



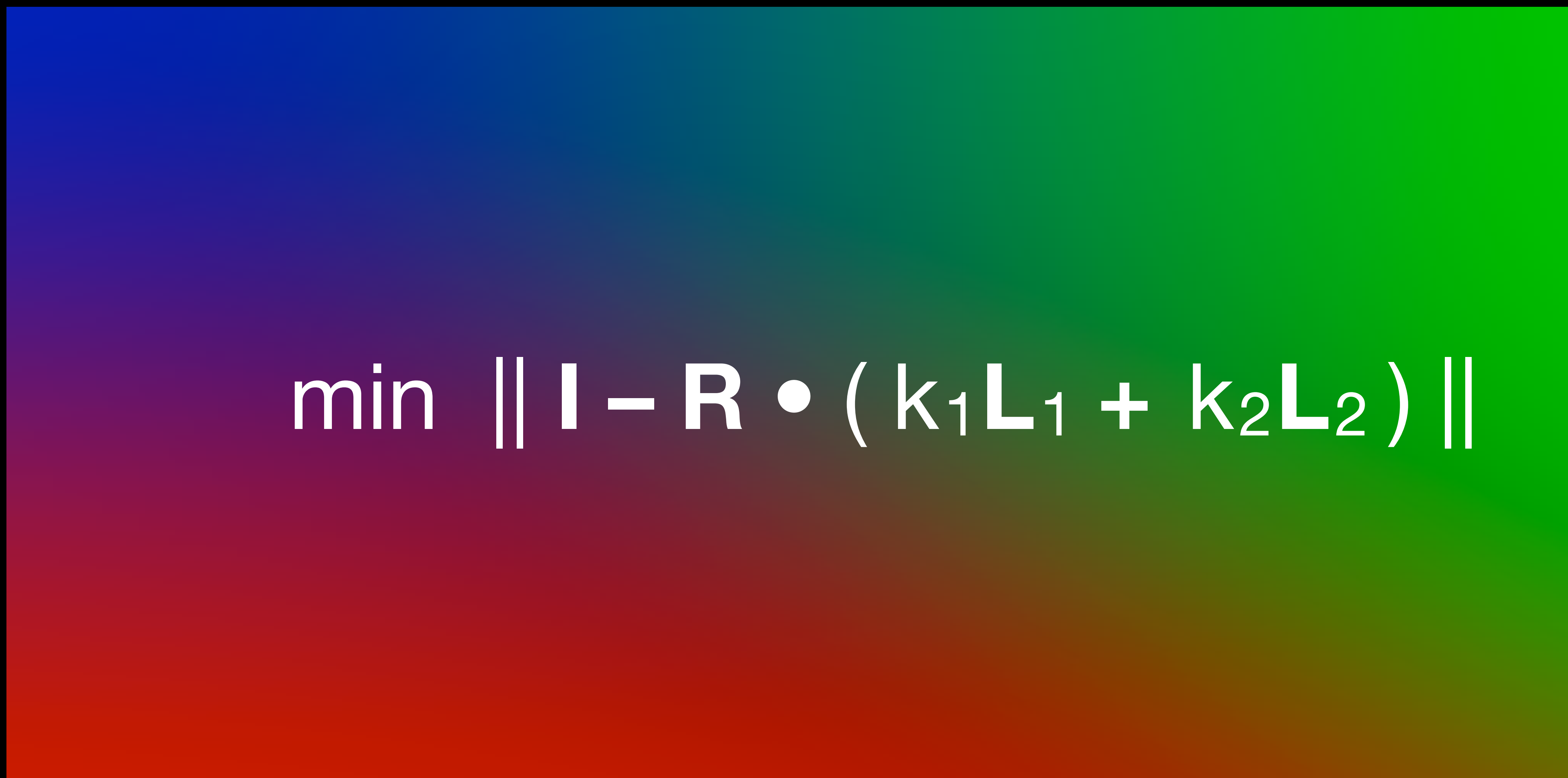
$$\mathbf{I} = \mathbf{R} \bullet (k_1\mathbf{L}_1 + k_2\mathbf{L}_2)$$

… how it could appear in the image?

We can use our image model to determine how this material color could appear in the image.



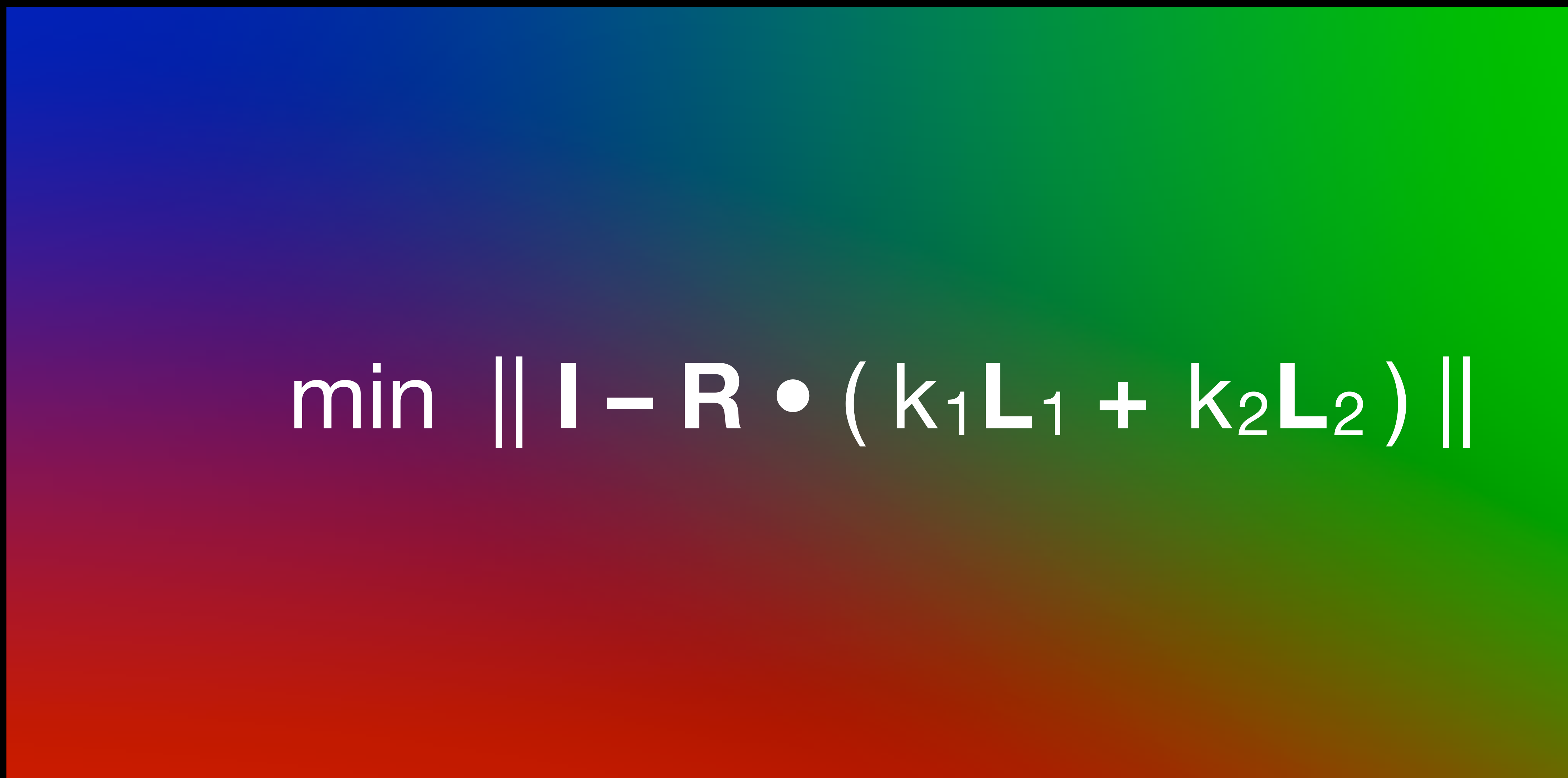
We can also figure out if the material color R could appear as pixel color I. And we do this by performing a least-squares estimation of k1 and k2 and computing the residual, as shown here.



… what could be that material color?

Thursday, September 29, 11

We can also figure out if the material color R could appear as pixel color I. And we do this by performing a least-squares estimation of k1 and k2 and computing the residual, as shown here.



$$\min \; \| \mathbf{I} - \mathbf{R} \bullet (k_1 \mathbf{L}_1 + k_2 \mathbf{L}_2) \|$$

Thursday, September 29, 11

If this residual falls below some threshold t, we say that the pixel I votes for the material color R. The dominant material color, then, is just the one that that gets the most votes.

So, let me go ahead and illustrate this.

$$t \geq \min \| \mathbf{I} - \mathbf{R} \bullet (k_1 \mathbf{L}_1 + k_2 \mathbf{L}_2) \|$$

If this expression holds, we say that the pixel votes for the material color.

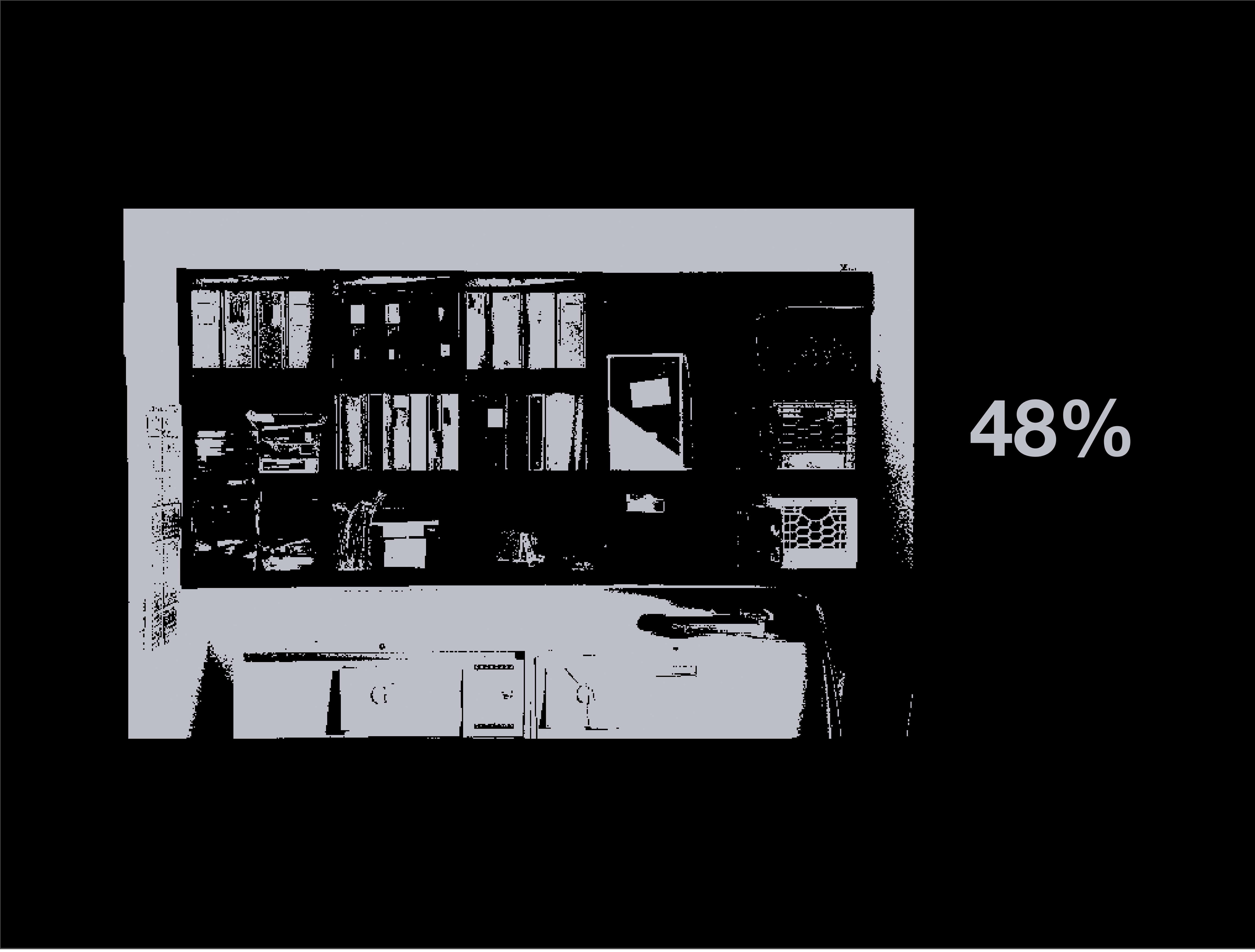
If this residual falls below some threshold t, we say that the pixel I votes for the material color R. The dominant material color, then, is just the one that that gets the most votes.

So, let me go ahead and illustrate this.

Given this scene …

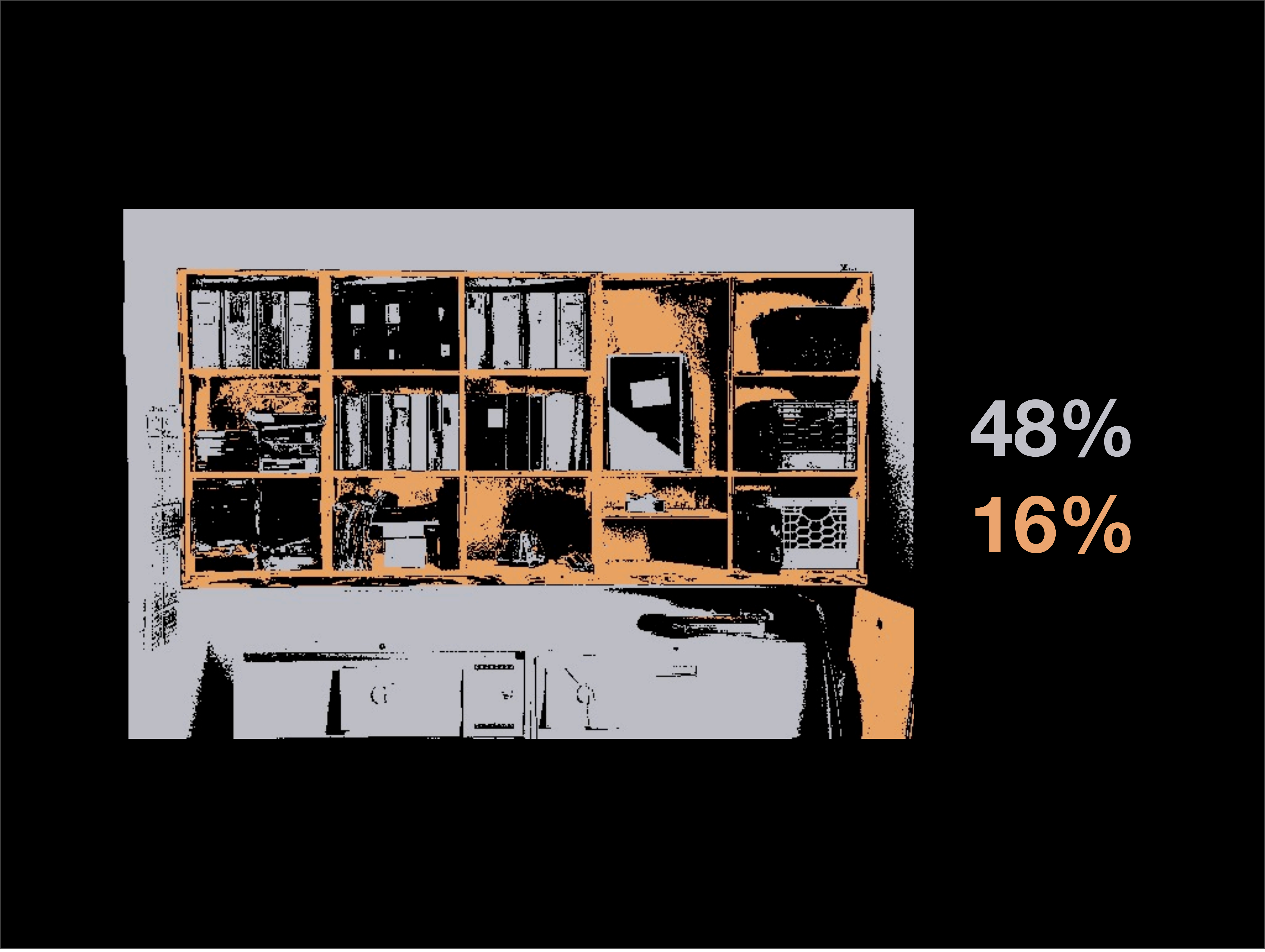
Thursday, September 29, 11

Given this scene …



Thursday, September 29, 11

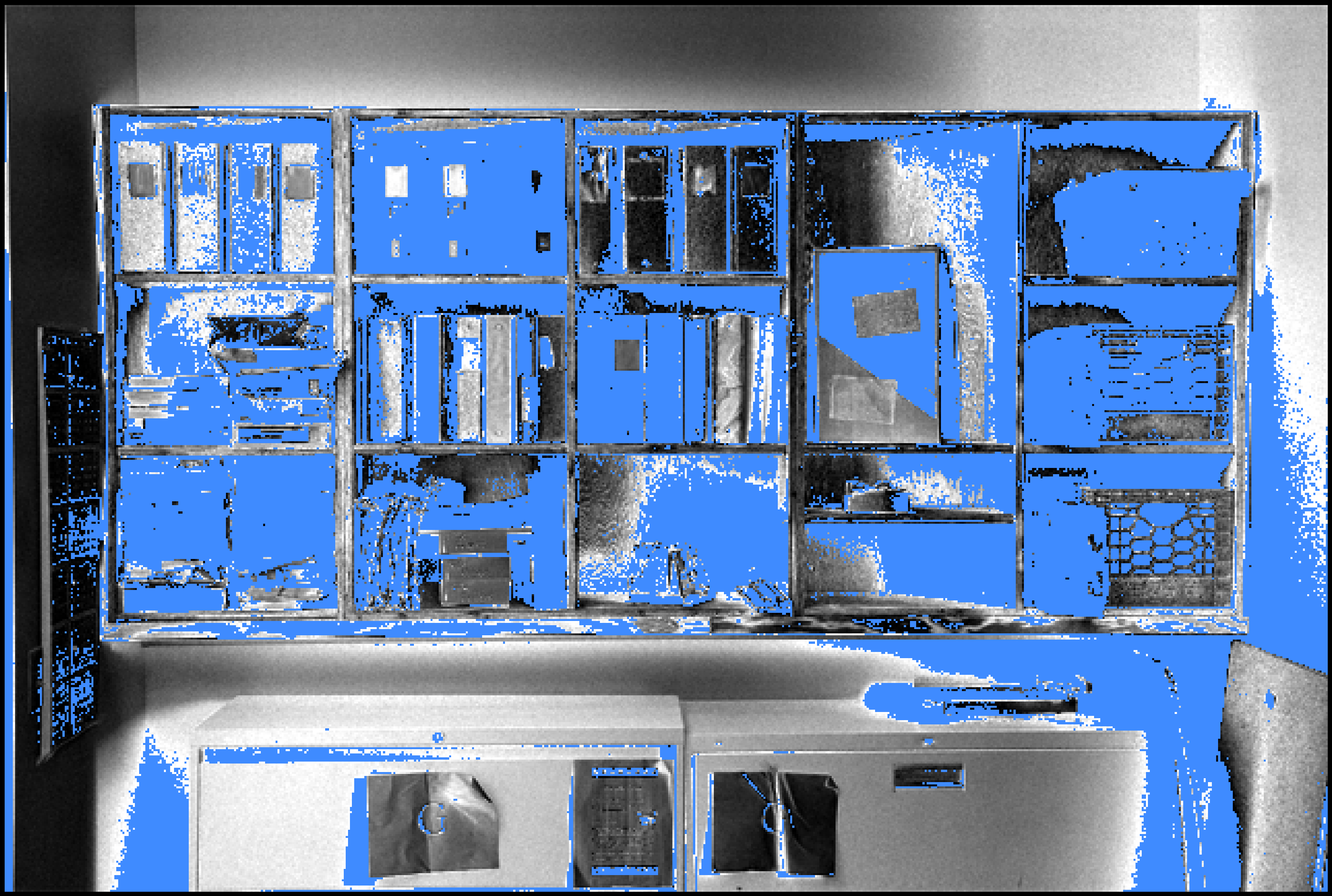
… the voting scheme determines that 48 percent of the pixels can be explained by a gray material color. For the remaining black pixels in this image, we take another vote for the next most dominant material color …



… which happens to be brown. This color accounts for 16 percent of the total pixels in the image. So, this process is repeated until the percentage drops below a certain threshold.

48%
16%

At this point, we can do white balance on large portions of the image, as shown here. The thing is, though, there's a lot of missing data. So the next step of our technique is to interpolate this data. But we have to be a bit careful about how we do it. For instance, we can't just interpolate the material colors shown here, because then the entire image would just turn out white and brown.



Thursday, September 29, 11

What we do instead is to use these material colors to determine the mixture of the two lights at each pixel. This is the alpha term that we saw earlier. We then use an interpolation technique to fill in the missing alpha values shown in blue.

Mixture Interpolation

Thursday, September 29, 11

So now, I'm going to describe how exactly we do that.

$$\begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix} = \begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k_1 \begin{bmatrix} L_{1R} \\ L_{1G} \\ L_{1B} \end{bmatrix} + k_2 \begin{bmatrix} L_{2R} \\ L_{2G} \\ L_{2B} \end{bmatrix}}$$

As you recall, this is our image model. It turns out that we can come up with an equivalent expression in a chromaticity space by doing two things.

[CLICK] First, we assume that the blue channel values of both lights are one. We can do this without loss of generality since we have these scale factors k1 and k2 to play with.

[CLICK] Second, we divide out the blue channels. Essentially, we just treat the RGB colors as homogeneous coordinates. And what we get is this [CLICK] expression for the image in chromaticity space. Here, alpha is what we defined before in our image model.

$$\begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix} = \begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k_1 \begin{bmatrix} L_{1R} \\ L_{1G} \\ L_{1B} \end{bmatrix} + k_2 \begin{bmatrix} L_{2R} \\ L_{2G} \\ L_{2B} \end{bmatrix}}$$

… assume that $\mathbf{L_{1B}}$ and $\mathbf{L_{2B}}$ are 1.

As you recall, this is our image model. It turns out that we can come up with an equivalent expression in a chromaticity space by doing two things.

[CLICK] First, we assume that the blue channel values of both lights are one. We can do this without loss of generality since we have these scale factors k1 and k2 to play with.

[CLICK] Second, we divide out the blue channels. Essentially, we just treat the RGB colors as homogeneous coordinates. And what we get is this [CLICK] expression for the image in chromaticity space. Here, alpha is what we defined before in our image model.

$$\begin{pmatrix} I_R \\ I_G \\ I_B \end{pmatrix} = \begin{pmatrix} R_R \\ R_G \\ R_B \end{pmatrix} \bullet \boxed{ k_1 \begin{pmatrix} L_{1R} \\ L_{1G} \\ L_{1B} \end{pmatrix} + k_2 \begin{pmatrix} L_{2R} \\ L_{2G} \\ L_{2B} \end{pmatrix} }$$

… assume that $\mathbf{L_{1B}}$ and $\mathbf{L_{2B}}$ are 1.

… divide out the blue channels.

As you recall, this is our image model. It turns out that we can come up with an equivalent expression in a chromaticity space by doing two things.

[CLICK] First, we assume that the blue channel values of both lights are one. We can do this without loss of generality since we have these scale factors k1 and k2 to play with.

[CLICK] Second, we divide out the blue channels. Essentially, we just treat the RGB colors as homogeneous coordinates. And what we get is this [CLICK] expression for the image in chromaticity space. Here, alpha is what we defined before in our image model.

$$\begin{bmatrix} I_R \\ I_G \\ I_B \end{bmatrix} = \begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{ k_1 \begin{bmatrix} L_{1R} \\ L_{1G} \\ L_{1B} \end{bmatrix} + k_2 \begin{bmatrix} L_{2R} \\ L_{2G} \\ L_{2B} \end{bmatrix} }$$

... assume that L_{1B} and L_{2B} are 1.

... divide out the blue channels.

$$\begin{bmatrix} I_{R \div B} \\ I_{G \div B} \end{bmatrix} = \begin{bmatrix} R_{R \div B} \\ R_{G \div B} \end{bmatrix} \bullet \boxed{ \alpha \begin{bmatrix} L_{1R \div B} \\ L_{1G \div B} \end{bmatrix} + (1-\alpha) \begin{bmatrix} L_{2R \div B} \\ L_{2G \div B} \end{bmatrix} }$$

Thursday, September 29, 11

As you recall, this is our image model. It turns out that we can come up with an equivalent expression in a chromaticity space by doing two things.

[CLICK] First, we assume that the blue channel values of both lights are one. We can do this without loss of generality since we have these scale factors k1 and k2 to play with.

[CLICK] Second, we divide out the blue channels. Essentially, we just treat the RGB colors as homogeneous coordinates. And what we get is this [CLICK] expression for the image in chromaticity space. Here, alpha is what we defined before in our image model.

$$\begin{matrix} I_{R\div B} \\ I_{G\div B} \end{matrix} = \begin{matrix} R_{R\div B} \\ R_{G\div B} \end{matrix} \bullet \boxed{\alpha \begin{matrix} L_{1R\div B} \\ L_{1G\div B} \end{matrix} + 1-\alpha \begin{matrix} L_{2R\div B} \\ L_{2G\div B} \end{matrix}}$$

The key thing to realize here is that this looks exactly like an image matting problem, where you're trying to solve for the alpha given some constraints.

So, in matting, these constraints are usually scribbles. But for us, these constraints are given by our voting algorithm.

$$\begin{matrix} I_{R\div B} \\ I_{G\div B} \end{matrix} = \begin{matrix} R_{R\div B} \\ R_{G\div B} \end{matrix} \bullet \boxed{\alpha \begin{matrix} L_{1R\div B} \\ L_{1G\div B} \end{matrix} + 1-\alpha \begin{matrix} L_{2R\div B} \\ L_{2G\div B} \end{matrix}}$$

This looks exactly like image matting!

The key thing to realize here is that this looks exactly like an image matting problem, where you're trying to solve for the alpha given some constraints.

So, in matting, these constraints are usually scribbles. But for us, these constraints are given by our voting algorithm.

$$\begin{matrix} I_{R\div B} \\ I_{G\div B} \end{matrix} = \begin{matrix} R_{R\div B} \\ R_{G\div B} \end{matrix} \bullet \boxed{\alpha \begin{matrix} L_{1R\div B} \\ L_{1G\div B} \end{matrix} + 1-\alpha \begin{matrix} L_{2R\div B} \\ L_{2G\div B} \end{matrix}}$$

Now that we've reduced our problem to matting, we can use existing matting techniques. We chose to use the matting Laplacian, as described by Levin and her colleagues in 2006. And the precise details of this are in our paper.

So, let me jump ahead and show some examples of this technique in action.

$$\begin{bmatrix} I_{R \div B} \\ I_{G \div B} \end{bmatrix} = \begin{bmatrix} R_{R \div B} \\ R_{G \div B} \end{bmatrix} \bullet \boxed{ \alpha \begin{bmatrix} L_{1R \div B} \\ L_{1G \div B} \end{bmatrix} + 1-\alpha \begin{bmatrix} L_{2R \div B} \\ L_{2G \div B} \end{bmatrix} }$$

We perform interpolation using the matting Laplacian [Levin et al. 2006].

Thursday, September 29, 11

Now that we’ve reduced our problem to matting, we can use existing matting techniques. We chose to use the matting Laplacian, as described by Levin and her colleagues in 2006. And the precise details of this are in our paper.

So, let me jump ahead and show some examples of this technique in action.

This scene was shot using multiple exposures, so we have ground truth.

Thursday, September 29, 11

This photo was shot using multiple exposures, which allows us to compute this [CLICK] ground truth alpha.

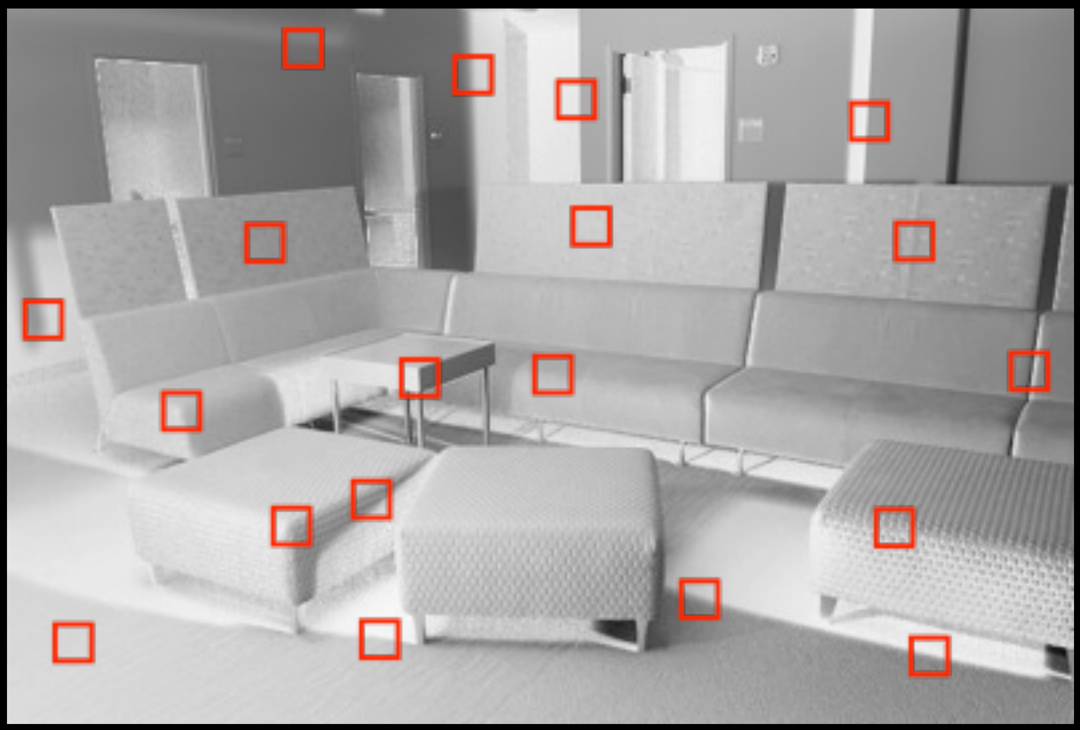
This scene was shot using multiple exposures, so we have ground truth.

Thursday, September 29, 11

This photo was shot using multiple exposures, which allows us to compute this **[CLICK]** ground truth alpha.

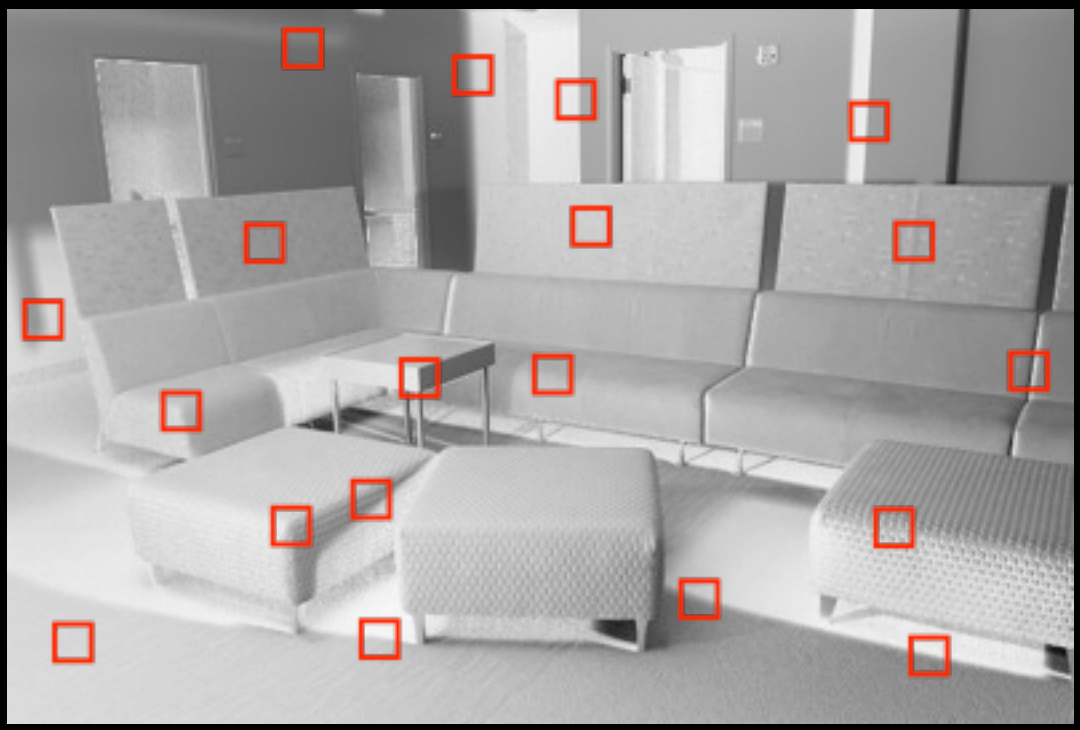
Thursday, September 29, 11

We set up a difficult test by only constraining the parts in the red squares. Then, we tried interpolating these parts to get the rest of the alpha.



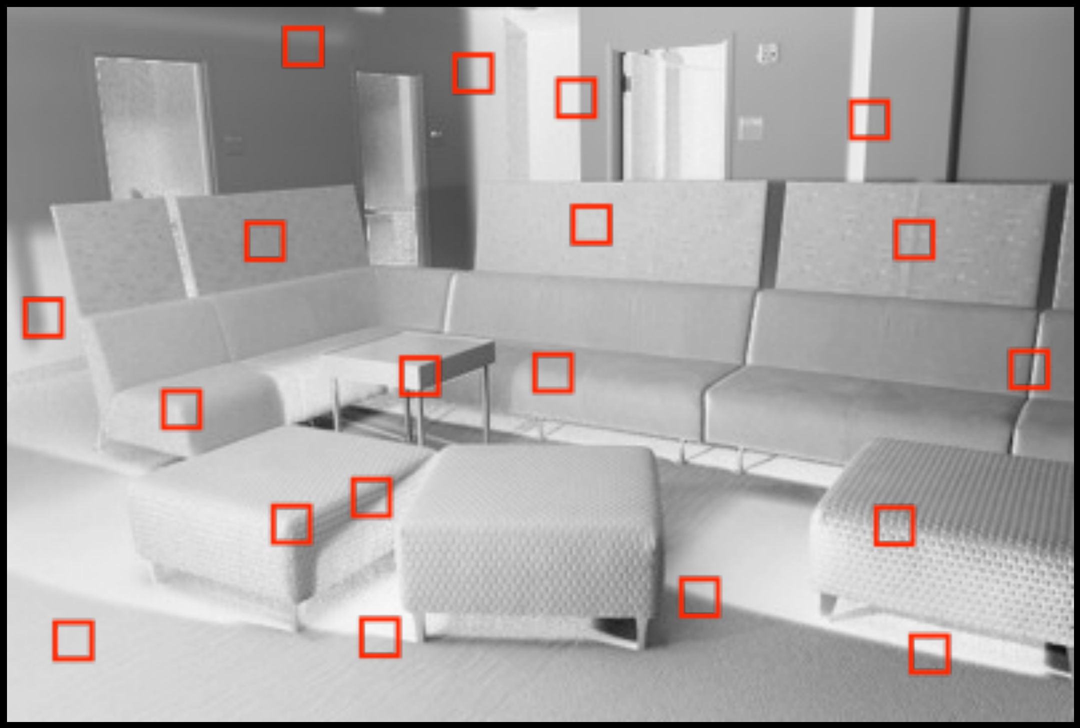
We constrain the points in the red squares and interpolate the rest.

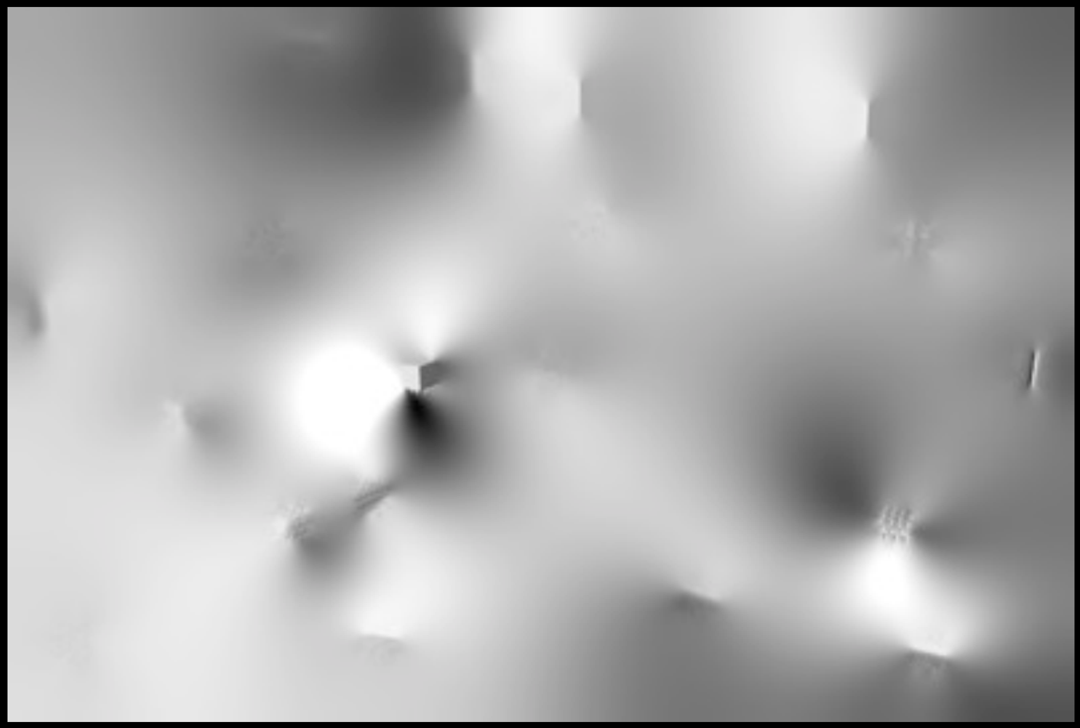
We set up a difficult test by only constraining the parts in the red squares. Then, we tried interpolating these parts to get the rest of the alpha.



Thursday, September 29, 11

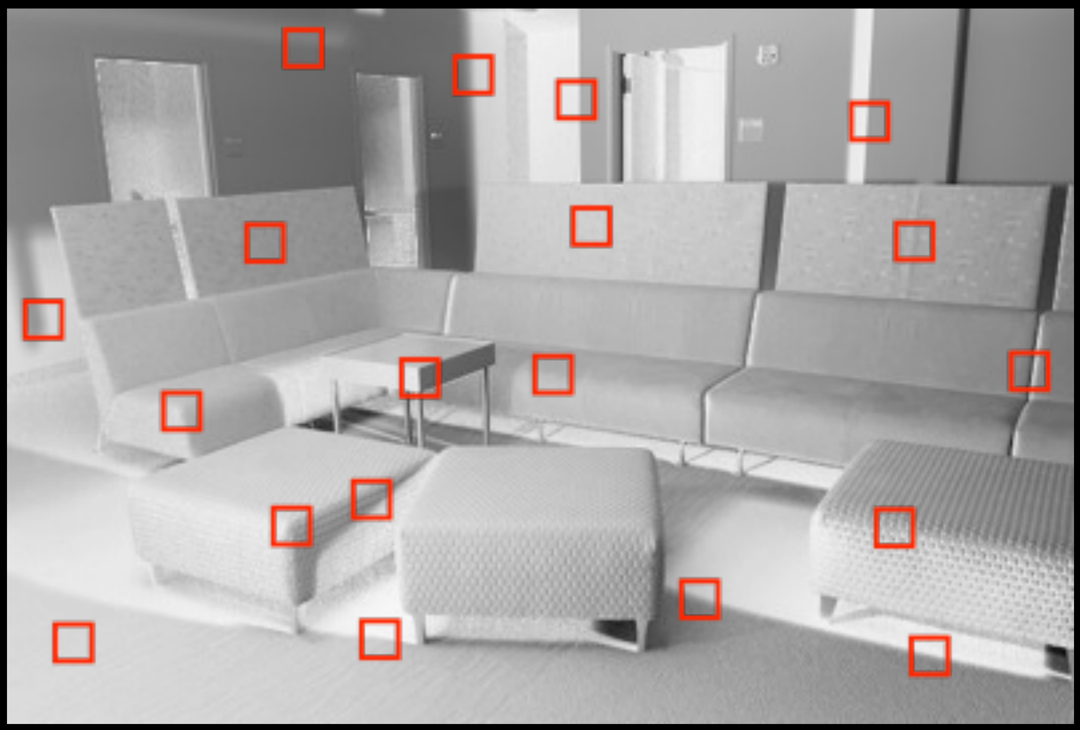
Standard smooth interpolation gives bad results.





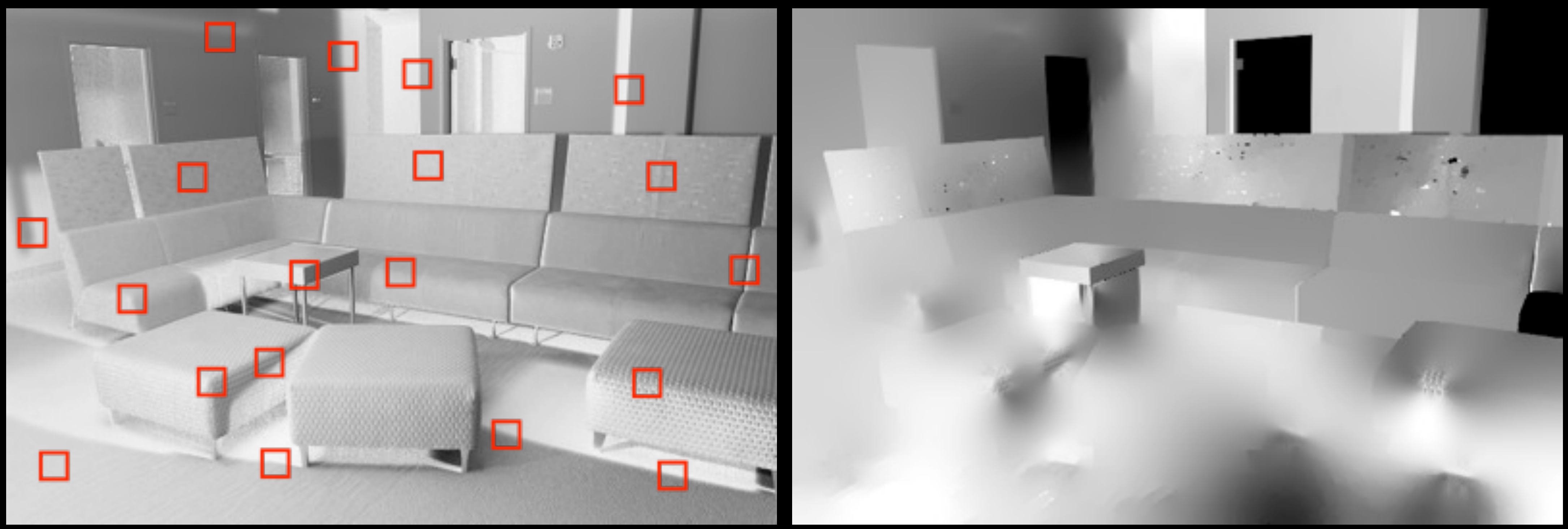
Smooth interpolation is pretty bad.

Standard smooth interpolation gives bad results.



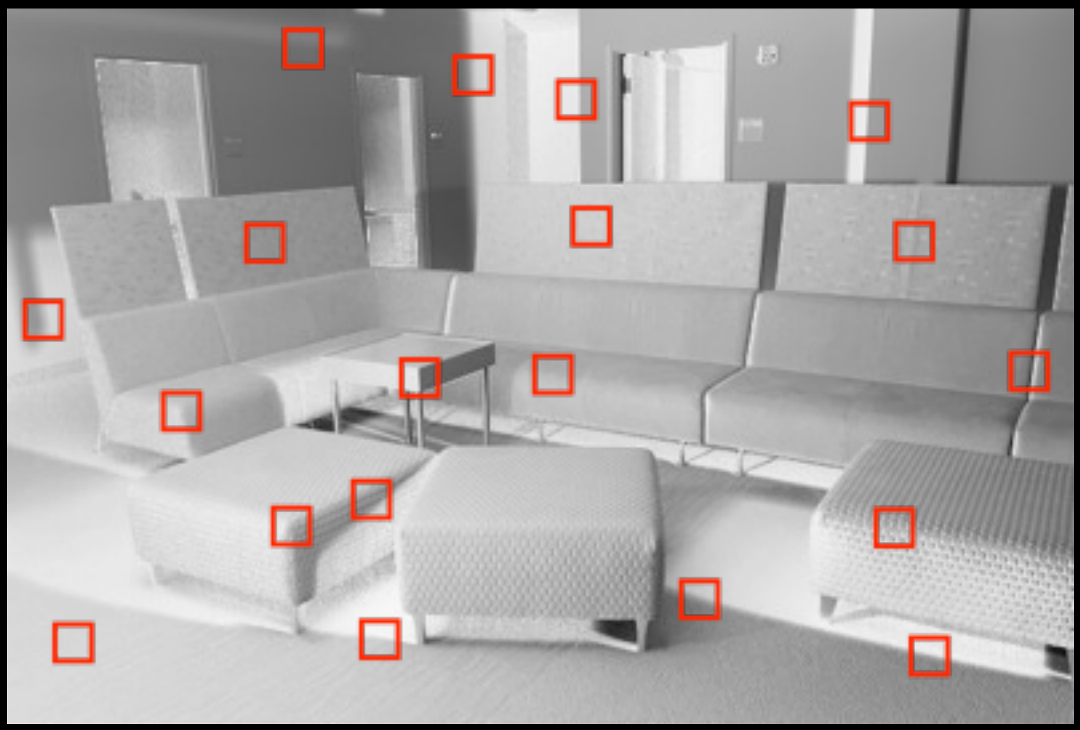
Thursday, September 29, 11

Edge-aware interpolation does a bit better, but the results are still pretty far from the desired solution.



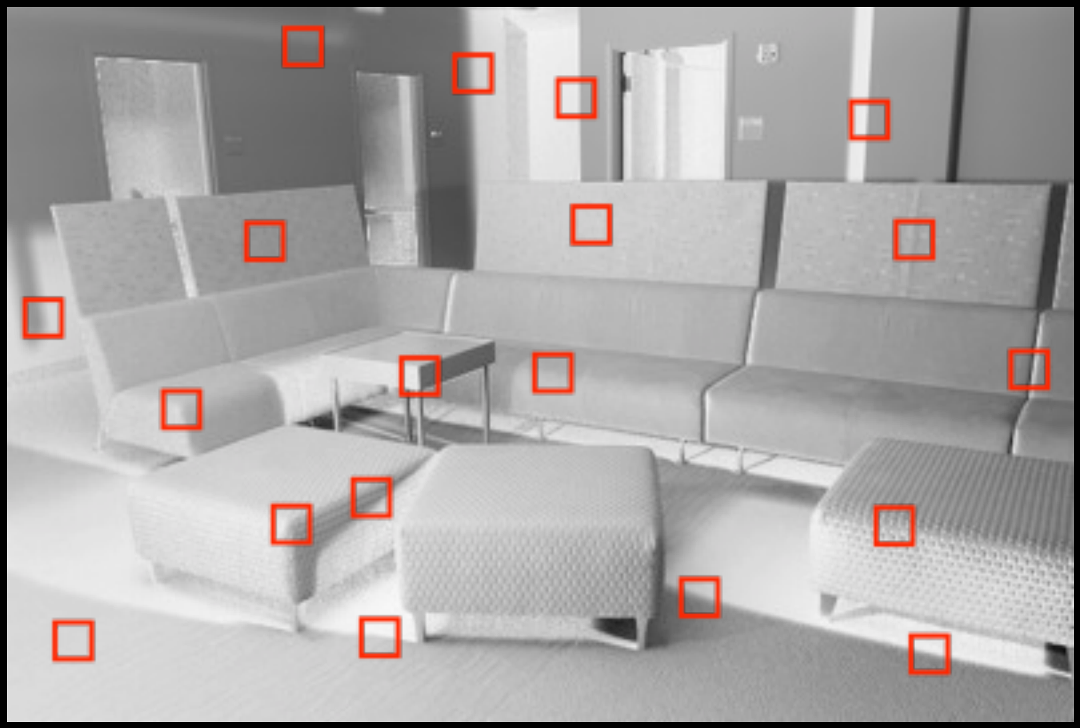
Edge-aware interpolation is also bad.

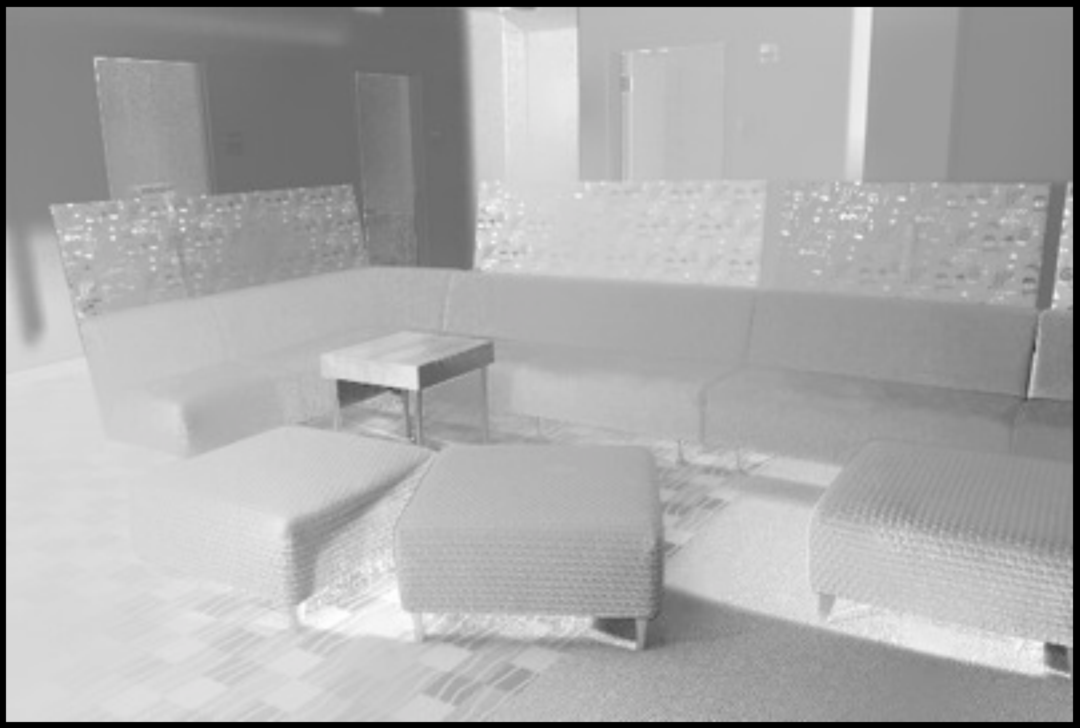
Edge-aware interpolation does a bit better, but the results are still pretty far from the desired solution.



When we apply the matting Laplacian, the results are a lot closer to the true solution.

This is actually a more difficult interpolation scenario than we'll usually encounter. The reason for this somewhat contrived example is to emphasize the quality of this technique.



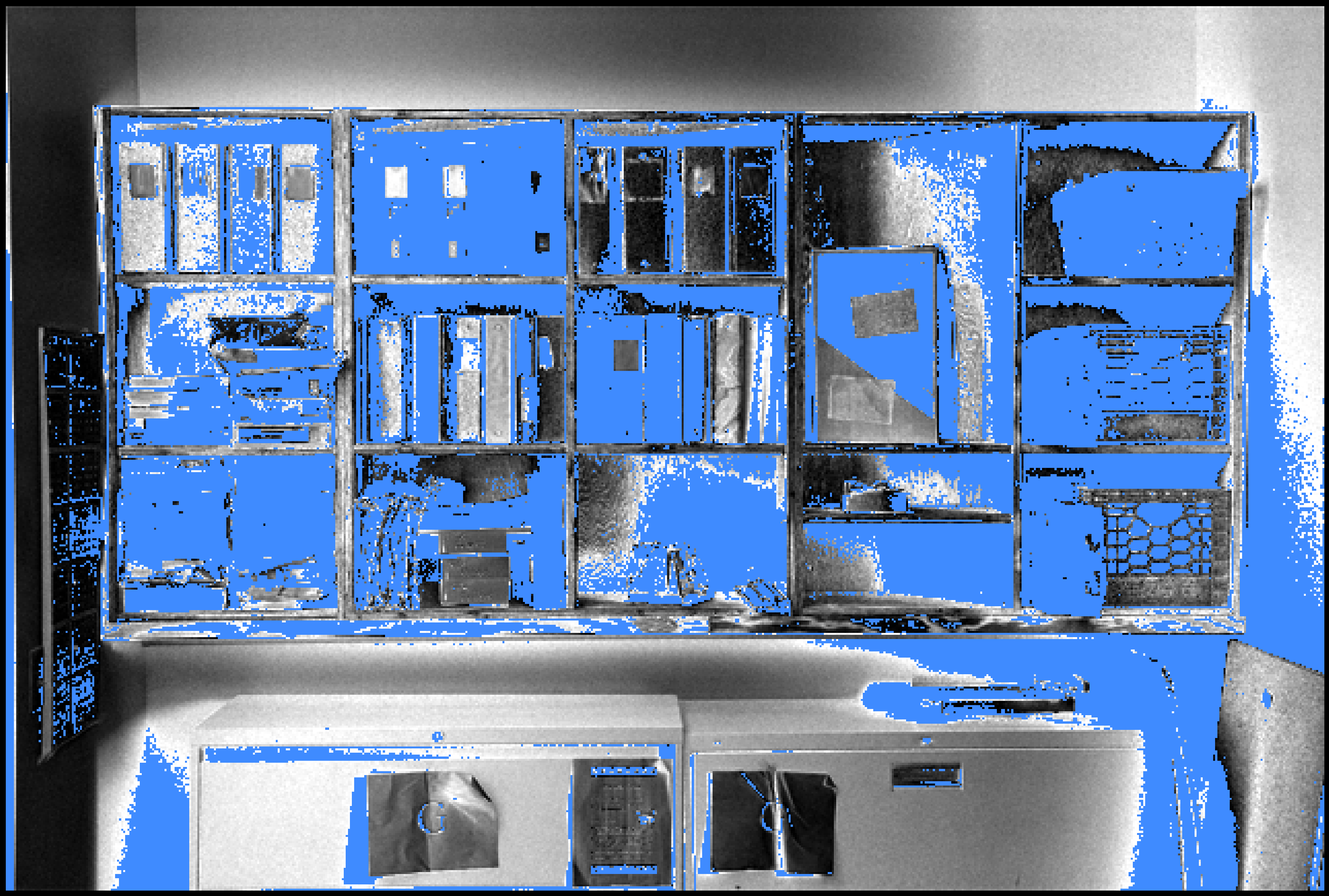


Matting Laplacian is much better.

Thursday, September 29, 11

When we apply the matting Laplacian, the results are a lot closer to the true solution.

This is actually a more difficult interpolation scenario than we'll usually encounter. The reason for this somewhat contrived example is to emphasize the quality of this technique.



Thursday, September 29, 11

So for completeness, let me just show you the interpolation result from this example again.

Thursday, September 29, 11

And here we go.

Thursday, September 29, 11

And here’s the final white-balanced result.

[CHECKPOINT 12:00]

Results

Thursday, September 29, 11

So let show you some results now.

The following results use synthetic inputs from multiple exposures to allow ground truth evaluations.

The first results I'd like to show use synthetic inputs constructed from multiple exposures. This allows us to compute ground truth results for evaluation.

Input

Here’s our first test, which has pretty obvious color artifacts from mixed lighting. Pay close attention to the color differences on the cardboard box and in the shadow areas.

[HOLD 3s]

Output

Thursday, September 29, 11

Our result eliminates those color differences.

[HOLD 3s]

Let’s go ahead and compare this to ground truth.

Ground Truth

[HOLD 3s]

And let’s see our result again.

Output

[HOLD 6s]

The following results are from single exposures with real mixed lighting.

The next set of results I'd like to show you are from single exposures with real mixed lighting.

Input

Thursday, September 29, 11

Here’s a portrait with mixed fluorescent lighting and off-camera flash. Notice the color variation across the face, especially in the highlight areas.

[HOLD 3s]

Thursday, September 29, 11

... and this is our result.

[HOLD 6s]

Input

Thursday, September 29, 11

This is a another test that mixes flash and tungsten lighting. Notice the color variations in the shadows.

[HOLD 3s]

Output

Thursday, September 29, 11

And this is out result.

[HOLD 6s]

Relighting

Another interesting thing we can do is relighting. To do this …

$$\begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + k_2 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}$$

From the white balanced image, separate the lighting contributions.

… we start from the white balanced image, which is represented by this expression. What we want to do is separate the contributions of the two light types. To do this …

$$\begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + k_2 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}$$

Thursday, September 29, 11

… we just scale the white-balanced image by alpha. Now, from our image model, [CLICK] we know that alpha is just k1 divided by k1 plus k2. So, a lot of things cancel out here …

$$\begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + k_2 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}} \bullet \alpha$$

Multiply the white balanced image by α for the first contribution.

… we just scale the white-balanced image by alpha. Now, from our image model, [CLICK] we know that alpha is just k1 divided by k1 plus k2. So, a lot of things cancel out here …

$$\begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + k_2 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}} \bullet k_1 \div \boxed{k_1 + k_2}$$

Multiply the white balanced image by α for the first contribution.

… we just scale the white-balanced image by alpha. Now, from our image model, [CLICK] we know that alpha is just k1 divided by k1 plus k2. So, a lot of things cancel out here …

R_R

R_G •

R_B

Multiply the white balanced image by α for the first contribution.

… and we get the contribution of the first light type.

$$\begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}$$

Multiply the white balanced image by α for the first contribution.

… and we get the contribution of the first light type.

$$\begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}$$

In the same way, we can multiply the white-balanced image by 1 minus alpha to get the second lighting contribution.

Multiply the white balanced image by 1–α for the second contribution.

In the same way, we can multiply the white-balanced image by 1 minus alpha to get the second lighting contribution.

$$\begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}} \qquad \begin{bmatrix} R_R \\ R_G \\ R_B \end{bmatrix} \bullet \boxed{k_2 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}}$$

At this point, we can go ahead and choose new lights. Adding the images together gets the desired effect.

So, let's see some examples.

$$\begin{bmatrix}R_R\\R_G\\R_B\end{bmatrix} \bullet \boxed{k_1 \begin{bmatrix}1\\1\\1\end{bmatrix}} \bullet \begin{bmatrix}L_{1R}\\L_{1G}\\L_{1B}\end{bmatrix} + \begin{bmatrix}R_R\\R_G\\R_B\end{bmatrix} \bullet \boxed{k_2 \begin{bmatrix}1\\1\\1\end{bmatrix}} \bullet \begin{bmatrix}L_{2R}\\L_{2G}\\L_{2B}\end{bmatrix}$$

We can choose new lights and add the images to get the desired effect.

At this point, we can go ahead and choose new lights. Adding the images together gets the desired effect.

So, let's see some examples.

Input

Thursday, September 29, 11

Here’s an input image. We saw this at the beginning of the talk.

Output

Thursday, September 29, 11

… or we can emphasize the sidelight and cut the ambient light for a sunset effect. And just a reminder that all of this is done from a single exposure.

Input

Thursday, September 29, 11

Here’s an indoor scene with mixed fluorescent lighting and daylight.

Output

This is what happens when we apply our white balance technique.

Output

And we can also use our technique to dim the interior lights.

Discussion

Thursday, September 29, 11

So, to wrap things up, I'd like revisit the key assumptions of our technique.

Thursday, September 29, 11

Our technique relies on the fact that there are two types of light in the scene which the user must identify.

It's not hard to pick the proper light types, but it still requires a bit of user interaction. So it's a natural question is **[CLICK]** whether we can automate this process, perhaps by extending current techniques for auto white balance. And that's something we'd like to address in future work.

Another question that comes up here is: **[CLICK]** why not more types of light? Why two? Well, practically speaking, two covers a lot of typical cases. Furthermore, introducing a third light type adds an extra degree of freedom to the image model. So, if we really wanted to handle that case, we'd have to introduce some other assumptions to make up for that.

There are two types of light in the scene which the user must identify …

Thursday, September 29, 11

Our technique relies on the fact that there are two types of light in the scene which the user must identify.

It's not hard to pick the proper light types, but it still requires a bit of user interaction. So it's a natural question is [CLICK] whether we can automate this process, perhaps by extending current techniques for auto white balance. And that's something we'd like to address in future work.

Another question that comes up here is: [CLICK] why not more types of light? Why two? Well, practically speaking, two covers a lot of typical cases. Furthermore, introducing a third light type adds an extra degree of freedom to the image model. So, if we really wanted to handle that case, we'd have to introduce some other assumptions to make up for that.

There are two types of light in the scene which the user must identify …

… can we automate identification?

Our technique relies on the fact that there are two types of light in the scene which the user must identify.

It's not hard to pick the proper light types, but it still requires a bit of user interaction. So it's a natural question is **[CLICK]** whether we can automate this process, perhaps by extending current techniques for auto white balance. And that's something we'd like to address in future work.

Another question that comes up here is: **[CLICK]** why not more types of light? Why two? Well, practically speaking, two covers a lot of typical cases. Furthermore, introducing a third light type adds an extra degree of freedom to the image model. So, if we really wanted to handle that case, we'd have to introduce some other assumptions to make up for that.

There are two types of light in the scene which the user must identify …

… can we automate identification?

… why not allow more light types?

Thursday, September 29, 11

Our technique relies on the fact that there are two types of light in the scene which the user must identify.

It's not hard to pick the proper light types, but it still requires a bit of user interaction. So it's a natural question is **[CLICK]** whether we can automate this process, perhaps by extending current techniques for auto white balance. And that's something we'd like to address in future work.

Another question that comes up here is: **[CLICK]** why not more types of light? Why two? Well, practically speaking, two covers a lot of typical cases. Furthermore, introducing a third light type adds an extra degree of freedom to the image model. So, if we really wanted to handle that case, we'd have to introduce some other assumptions to make up for that.

The scene is dominated by a small set of material colors …

Of course, the big assumption of our technique is that the scene is dominated by a few material colors. We've shown that this assumption is reasonable for a variety of typical images. But it's certainly not a foolproof assumption. For our technique to work, those material colors have to appear under different mixtures of light.

The scene is dominated by a small set of material colors …

… which must appear under different mixtures of light.

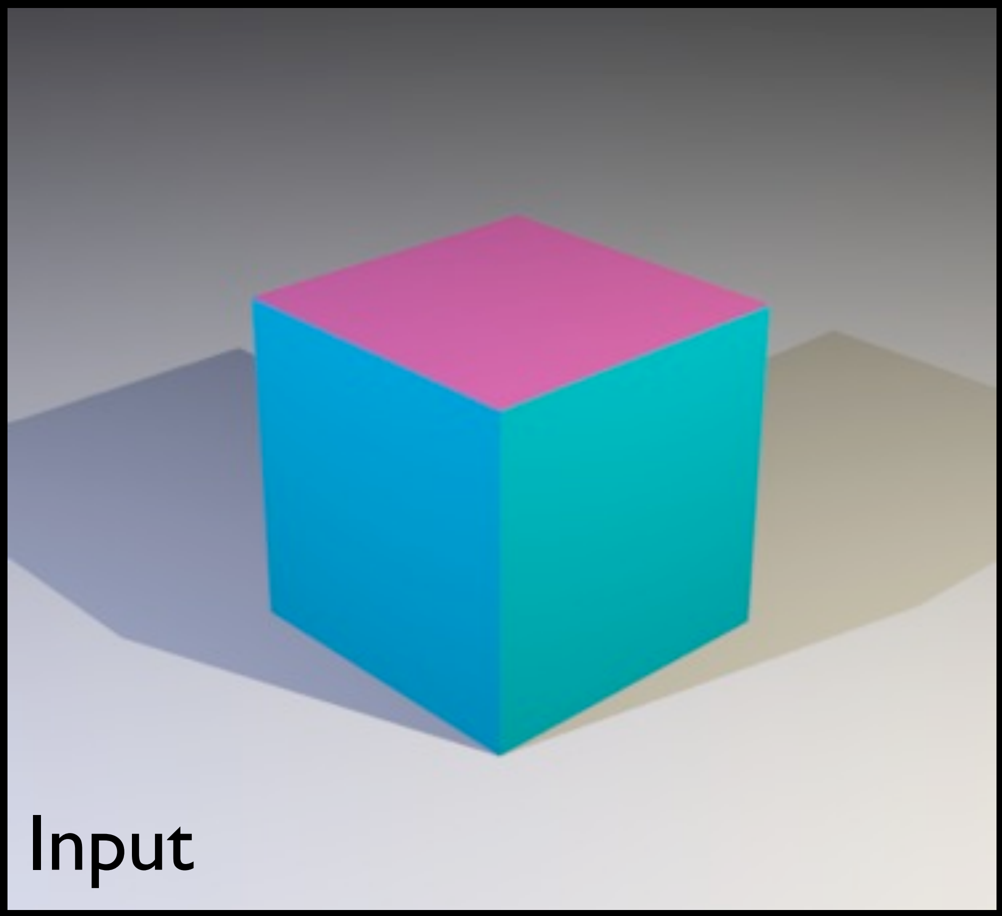
Thursday, September 29, 11

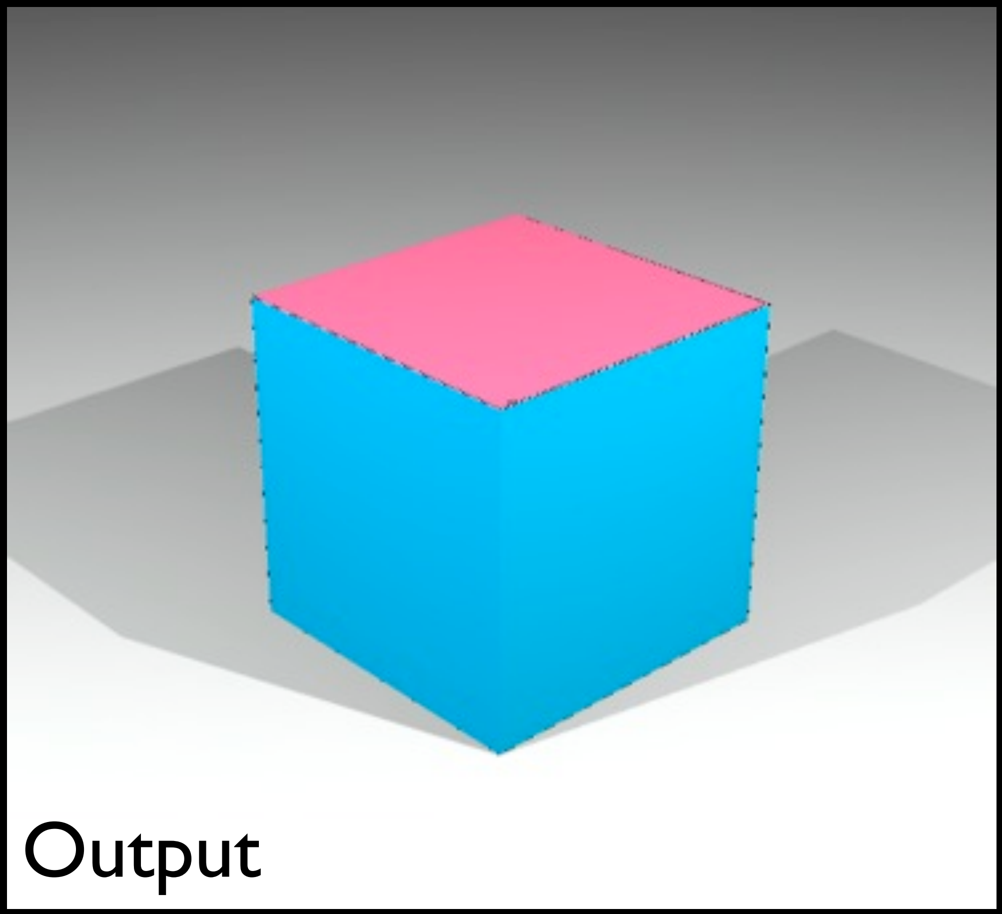
In some cases, this might not hold. For instance, if the two light types are both very diffuse, the true mixture might be almost constant in the scene. There also might be cases where there's a nearly binary foreground-background separation. For instance, a direct flash in the face with a distant background.

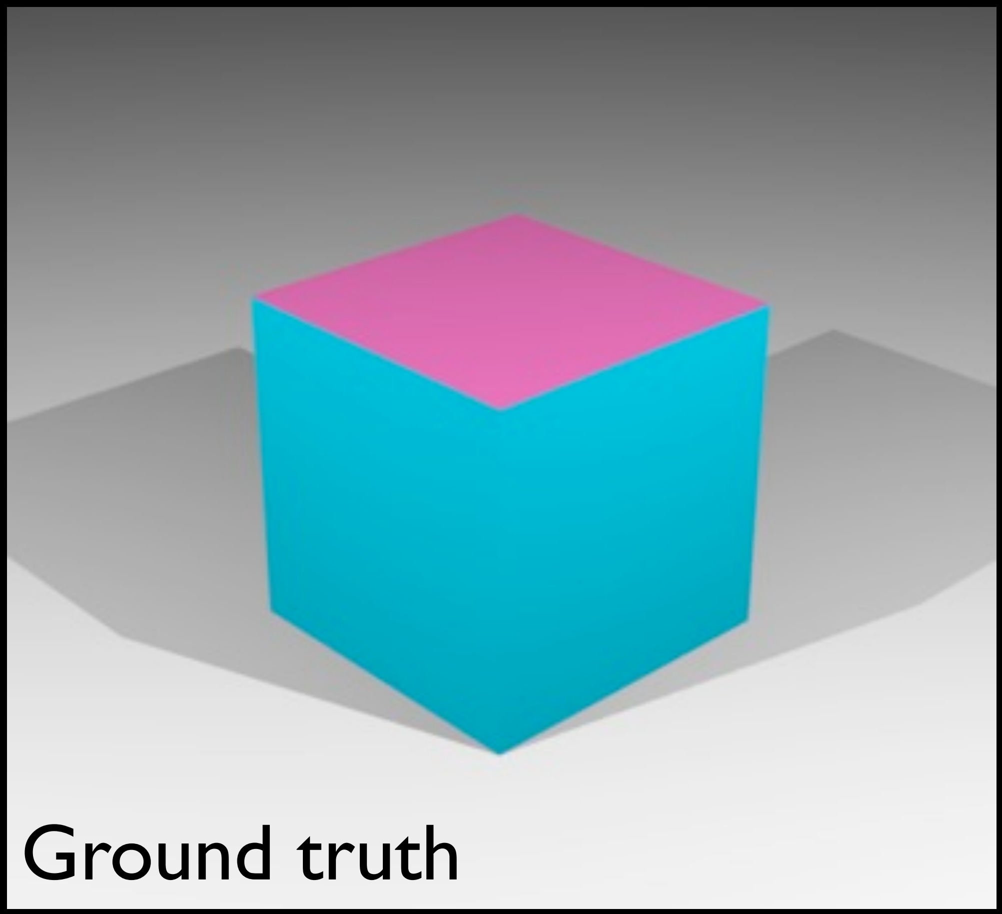
Now, it's pretty unlikely that our technique would work too well in these scenarios. Thankfully, they don't seem to come up too often.

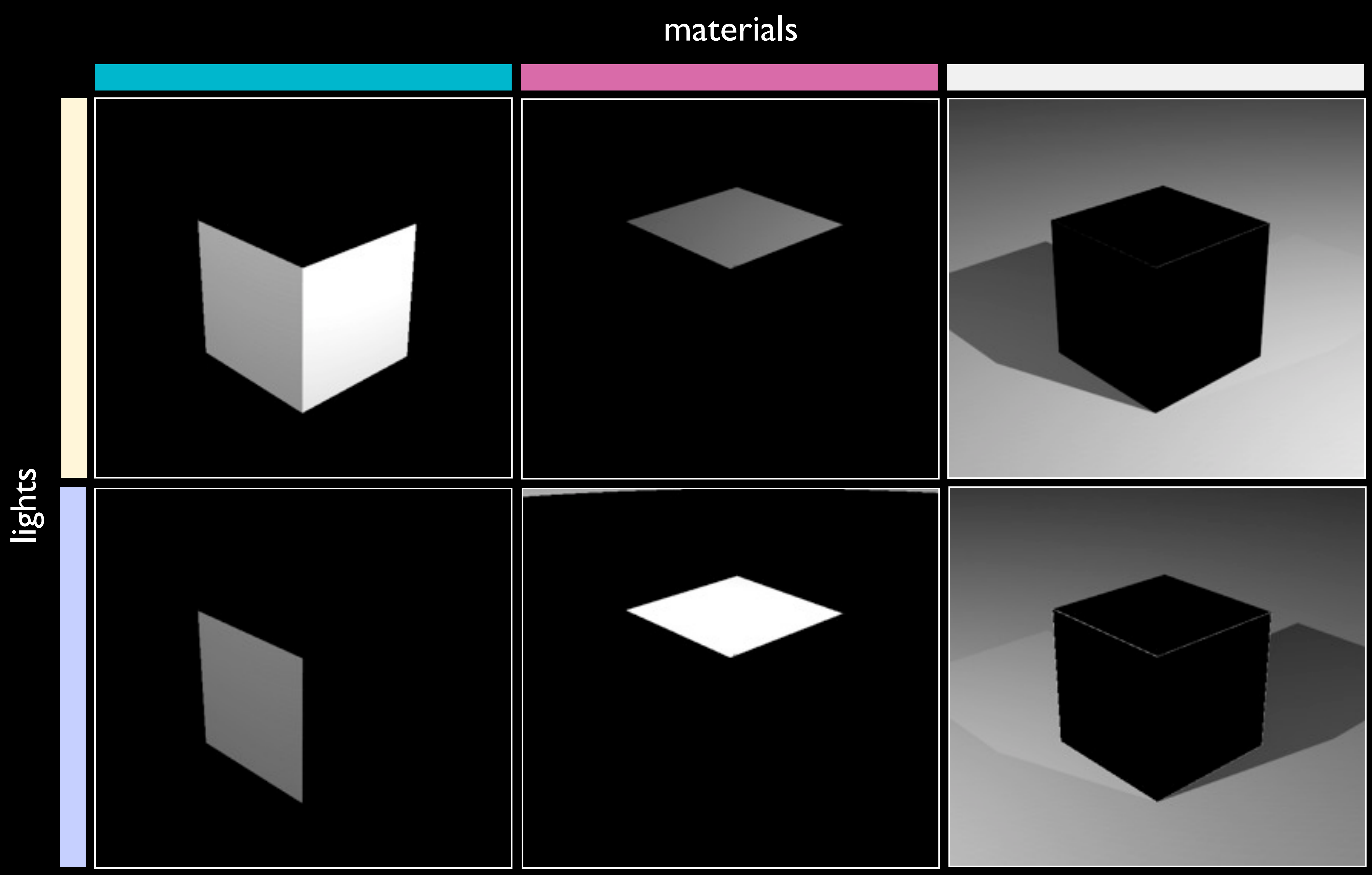
So, let me summarize what I’ve shown you today.

Demo


Input

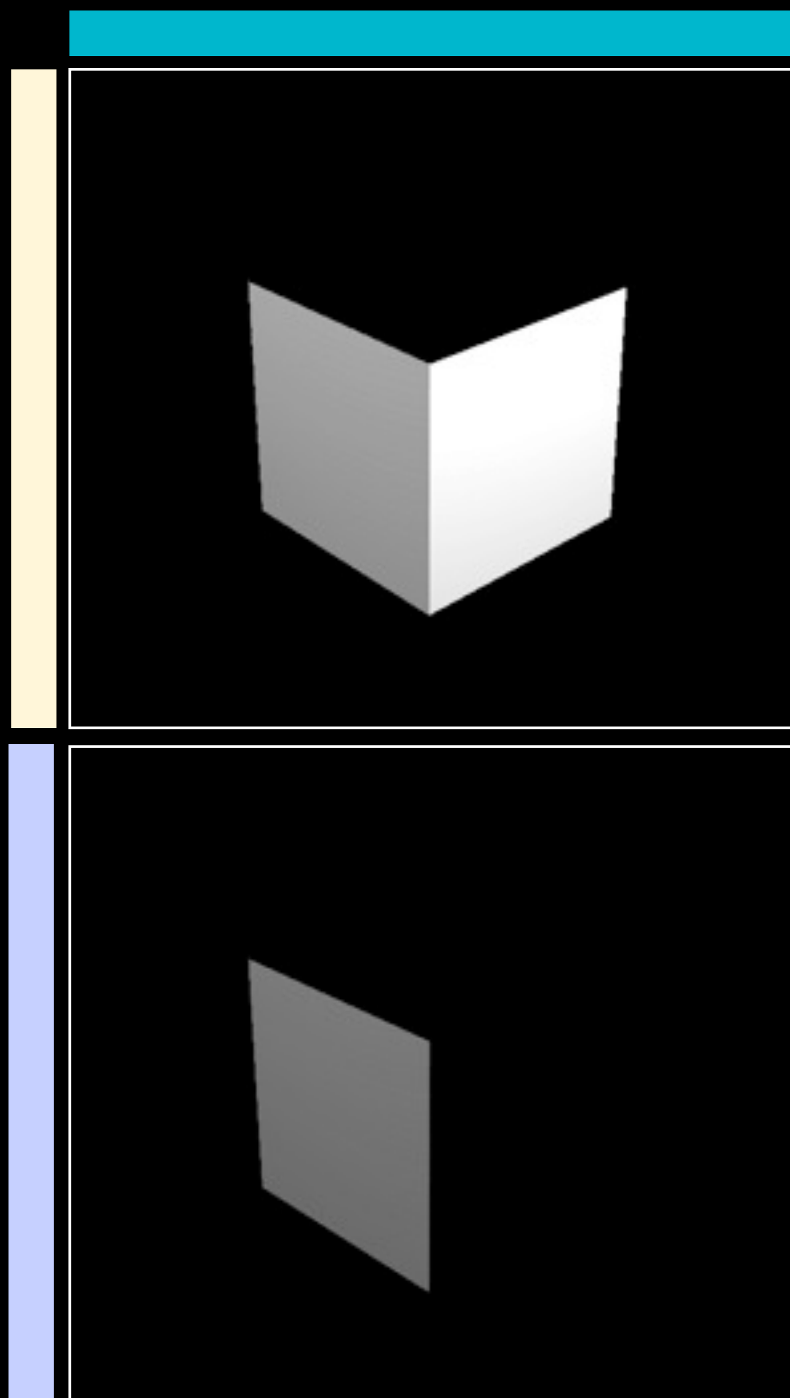

Output


Ground truth

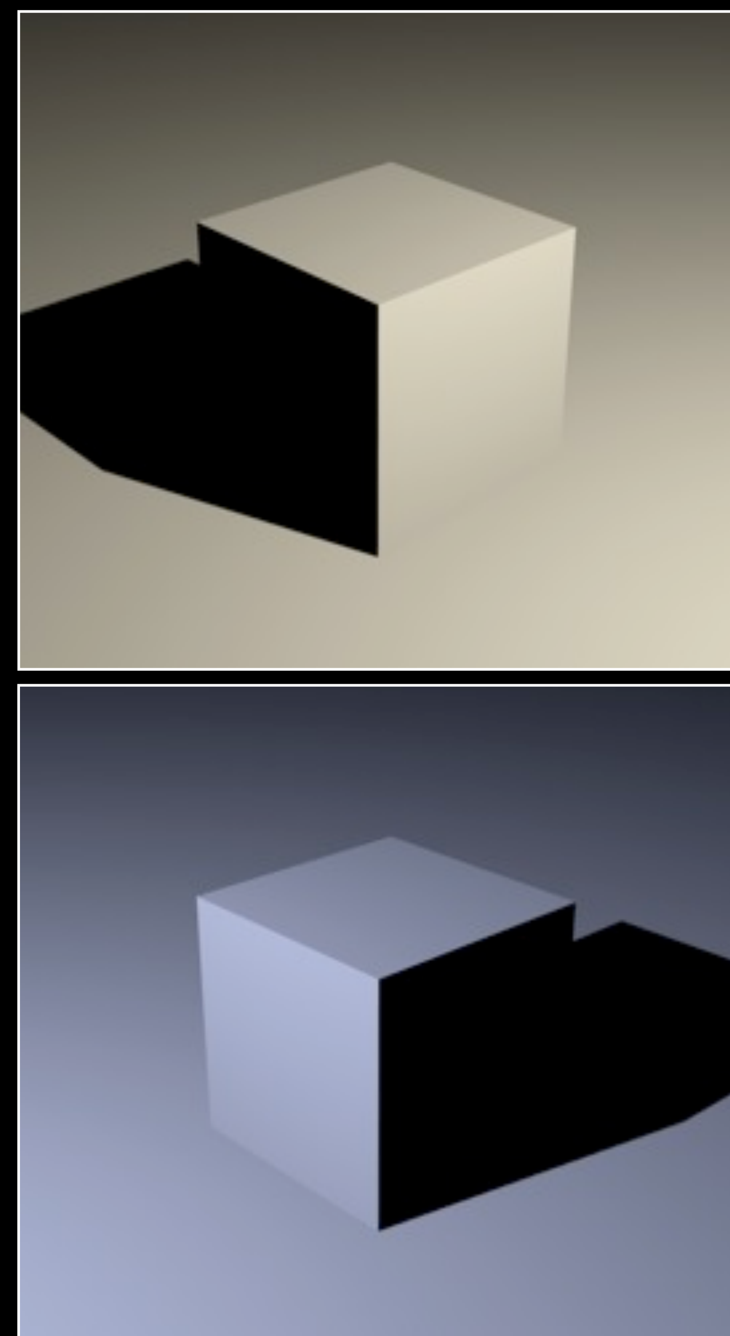

materials
lights

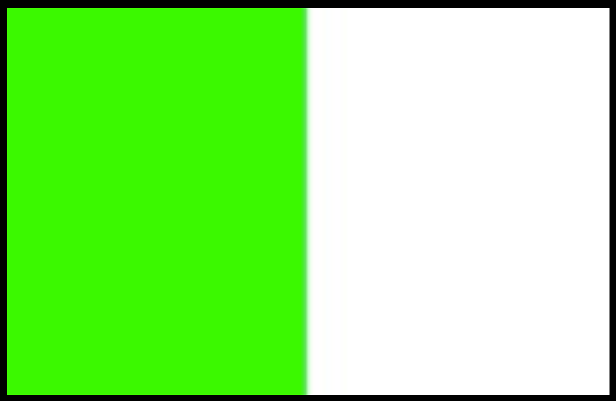
materials

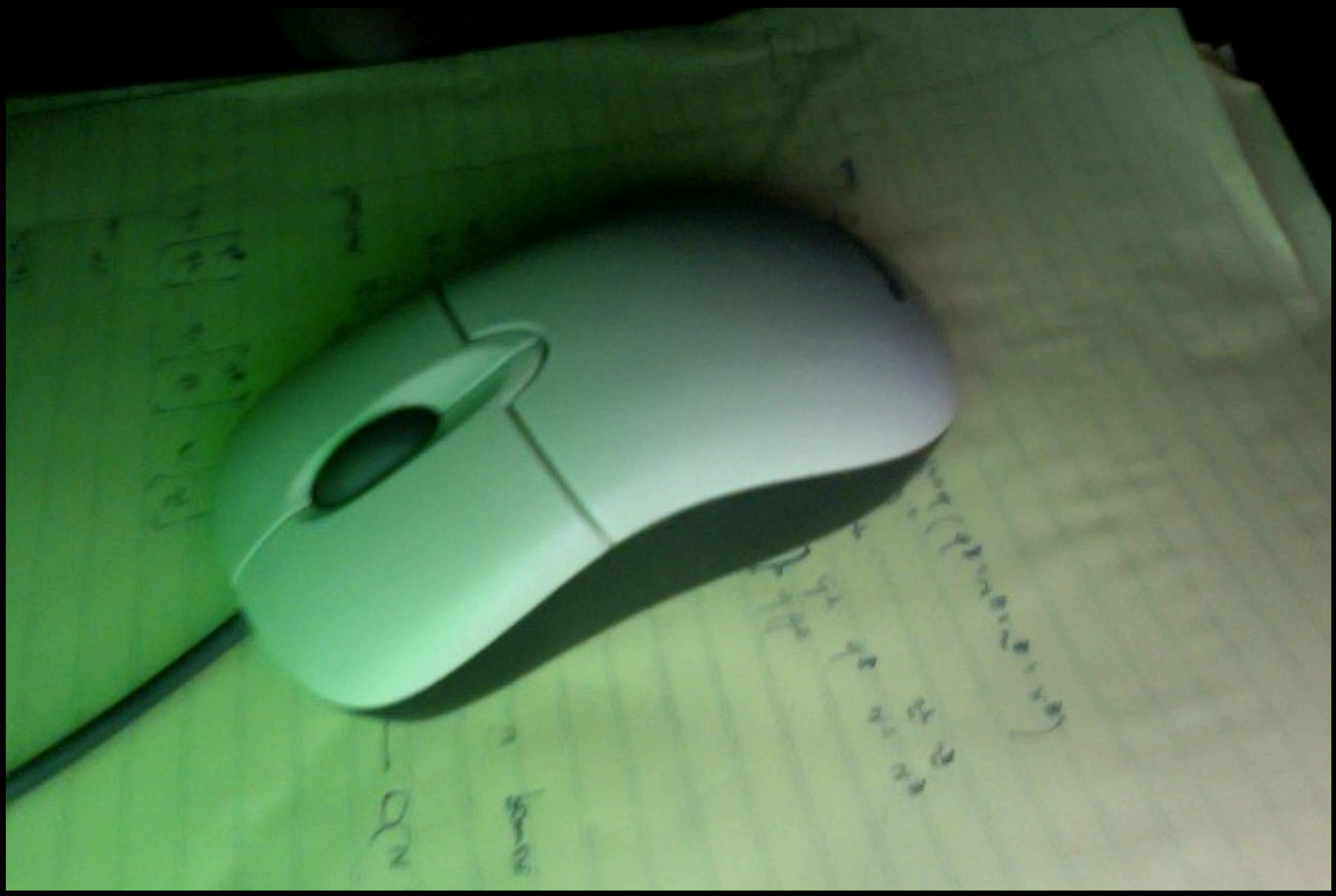
lights

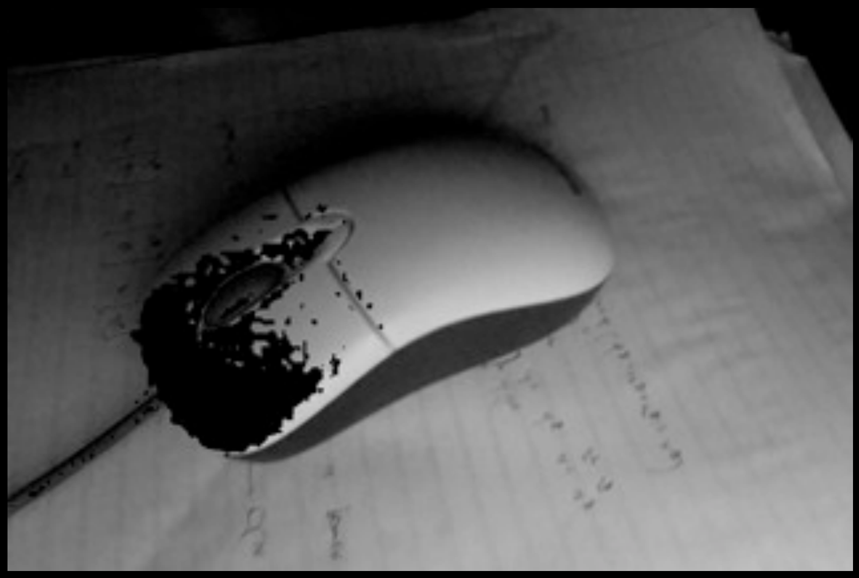


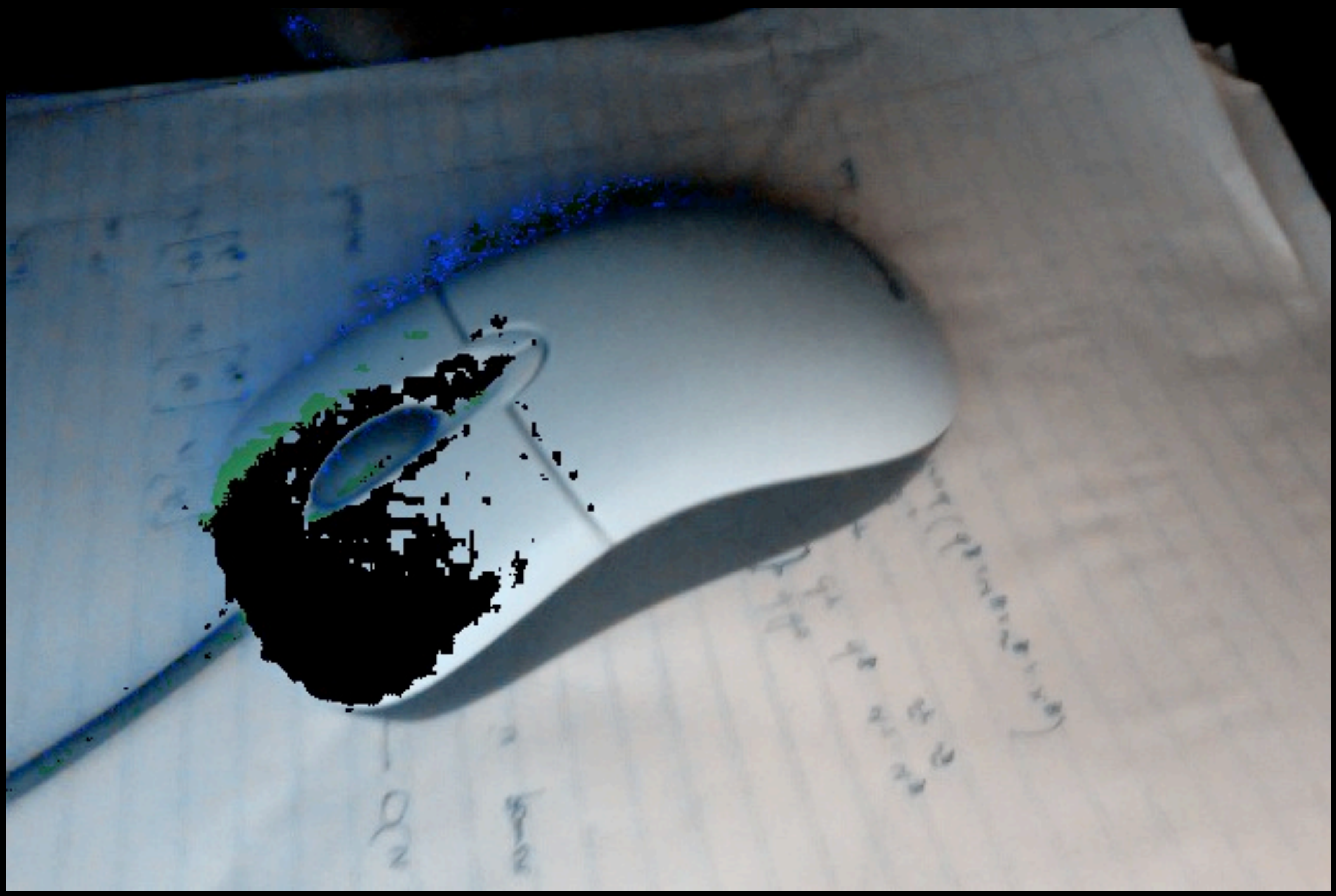
all materials white, one light source



light source: 







Thursday, September 29, 11