Datorgrafik DT3025

March 18, 2016

Jourhavande lärare Martin Magnusson, phone 073 667 2258, available during the whole exam.

Tools No tools (books or calculators) are allowed.

There are three types of questions on this exam, targeted for the goals that correspond to a passing grade (3) and a pass with distinction (grades 4–5).

- To pass with grade 3 (G), you need 12 out of 16 points (75%) from the {3} questions *or* 20 points in total (from all questions).
- To pass with grade 4 (VG), you additionally need 75% from the {4} questions.
- To pass with grade 5, you additionally need 75% from the {5} questions.

You may answer in Swedish or English.
Good luck!

Question 1

4 points {3} Describe the following four terms: gamut, hue, saturation, refraction.

4 points {4} A common colour space is sRGB, where colours to be displayed are modelled as a weighted sum of three specific R, G, B components. It has a smaller nominal gamut than, e.g., the Adobe RGB colour space. Consider extending sRGB to a "sRGBC" colour space that also includes a cyan component. Would that affect the gamut? In what way, and why?

4 points {5} It is convenient to assume a single pin-hole camera model with zero exposure time when rendering computer graphics. List advantages and disadvantages of this model. Are the assumptions behind using this model valid? Explain why or why not.

Solution

{3} One point each:

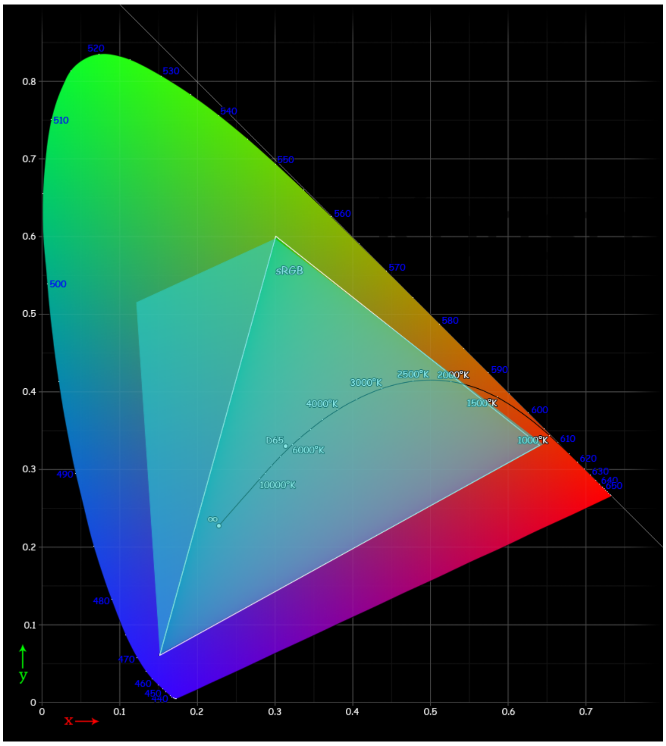
Gamut The set of colours that can be represented in a certain colour space or by a certain device.

Hue E.g., blue, orange, red. The appearance of a colour that is independent of brightness or saturation, and is associated with the dominant frequency, or combination of frequencies, of light.

Saturation The relative colour purity, or how little white light is mixed in. Monochromatic light has full saturation, while white light has zero saturation.

Refraction Bending of light as it passes the boundary between two materials in which the speed of light is different.

{4} Yes, it would increase the gamut — at least if the C component is chosen appropriately. The R, G, B components (base colours) of sRGB form a triangle in the CIE chromaticity diagram. If the C base colour is chosen outside of this triangle, the gamut of the sRGBC colour space will be bigger. However, if the cyan base colour is chosen to be anywhere within the sRGB triangle, then it would not affect the gamut. The image below illustrates the sRGB gamut (triangle) and a possible sRBGC gamut overlayed (the shaded quad).



Original image credit: Spiget, via Wikipedia

{5}

- Pin-hole camera (in other words, an aperture of size zero) means we have no depth of field (out-of-focus blur).
- Zero exposure time means we have no motion blur.
- Zero aperture (pin hole) and zero exposure time also means we would have zero photons, so no image at all, if the image were rendered in a physically correct way.
- Single camera means no depth perception.

Avoiding focus blur and motion blur can be advantageous, since in many cases such effects are not desirable, but the inability to render these effects even if we want to is a disadvantage. A clear advantage is that the computations are simpler.

Single-camera model makes sense as long as we are only interested in photo-realism, as opposed to 3D perception.

Pin-hole camera may not be physically strictly valid (it does not make sense to have an aperture big enough for a single photon) but is a sensible approximation to cameras with small apertures that render both near and far objects sharply.

Zero exposure time, by a similar argument, is a sensible approximation to a camera with a high shutter speed that captures an image quicker than the time it takes for an object to move one pixel in the image.

Question 2

4 points {3} Describe the difference between Phong or Gouraud shading and the Phong reflectance model.

2 points {4} Figure 1a shows a white Lambertian sphere lit by a white point light source (located behind the camera in this scene). The BRDF of a Lambertian surface is a constant value. Why is it then that this surface does not appear flat white, but is darker towards the silhouette?

2 points Assume we want to model the moon as a smooth sphere. Suggest a lighting model that would be more suited than purely Lambertian scattering (to achieve a result closer to Figure 1b). Disregard the texture.

4 points {5} The rendering equation can be written in the following two, slightly different, forms.

$$L(P, \omega_o) = L^e(P, \omega_o) + \int_{\omega_i \in S^2(P)} L(P, -\omega_i) f(P, \omega_i, \omega_o)(|\omega_i \cdot \mathbf{n}_P|)\, d\omega_i \quad (1)$$

$$L(P, \omega_o) = L^e(P, \omega_o) + \int_{\omega_i \in S^2_+(P)} L(P, -\omega_i) f(P, \omega_i, \omega_o)(\omega_i \cdot \mathbf{n}_P)\, d\omega_i \quad (2)$$

What is the difference between Equation (1) and (2), in terms of which materials they can model? Describe the implications of the difference in the limits used in the integral and the use of an absolute value in the last factor.

Solution

2 points {3} "Phong shading" is just a way to interpolate the rendered colour over a polygon face, which (as opposed to Gouraud shading) interpolates the normals, and computes the colour using a per-pixel normal. (Gouraud shading, on the other hand, simply evaluates the colour at the corner pixels of the polygon and linearly interpolates the colour over the face.)

2 points Phong's reflection model is the traditional model that computes the reflected light as a sum of an ambient, specular, and diffuse term (with particular choices for how to compute the specular and diffuse terms).

{4} Because the amount of reflected light is also multiplied by the cosine of the angle between the surface normal and the direction of the incoming light. (The answer should somehow mention this; that is, Lambert's cosine law.)

Phong reflection with a large ambient (or emissive) term would be an easy solution. More physically correct would be to model the surface using the Cook–Torrance model (or better yet, the Oren–Nayar model).

(The explanation for the moon being almost equally bright at the edges is that the surface is rough, so there are always some surface that is perpendicular to the light direction. Cook–Torrance models rough surfaces as a collection of small mirrors, and Oren–Nayar models them as a collection of small Lambertian reflectors — which is a reasonable model for the moon.)

{5} Equation (1) integrates incoming light over all the full sphere, so it can also be used to render transparent materials, with incoming light from below the surface. Equation (2), on the other hand, integrates over the positive hemisphere; in other words, only light from outside the surface. Therefore it does not take into account light that has passed through the object. That is to say that it can only model opaque surfaces. (Equation (1) uses the *reflectance equation* while Equation (2) uses the *scattering equation*.)

Question 3

Figure 2 shows an indoor scene with two light sources, rendered with path tracing.

4 points {3} Describe the difference between ray tracing and path tracing in terms of rendering indirect illumination, and how the difference would affect the appearance of Figure 2.

4 points {4} Path tracing implementations typically make use of *Monte Carlo integration*. Use Monte Carlo integration, with one sample, to compute the integral $\int_0^1 x^2 \, \mathrm{d}x$ (using appropriate assumptions when needed).

4 points {5} Some rendering methods produce a *biased* value of each pixel and some rendering methods produce an unbiased value, but with a larger *variance*. Explain where such bias and variance come from, and how they affect the appearance of the rendered image.

Solution

{3} Ray tracing typically traces one ray per pixel to its first intersection with a surface, and emits one recursive ray from the intersection point towards each light source to determine whether it is shadowed or not. Additionally, ray tracing emits a recursive ray if the material has a reflective component, and another if it has a translucent component. (Stochastic ray tracing optionally emits several rays with slight perturbations; for example to simulate blurred reflections [several rays from a reflection point], or to mitigate aliasing artifacts [several rays through a pixel].)

Path tracing, on the other hand, traces many rays per pixel, where each ray is traced until its "end"; that is, the whole path of light. (In practice it is not possible to trace the path until it ends up at the light source, so the path can be terminated with probability proportional to the reflectance of the material at each bounce.) Additionally, at each bounce, secondary rays are traced to all light sources in order to determine whether the point is shadowed or not.

With ray tracing, only surfaces that are directly lit by a light source would be non-black in Figure 2; that is, the corner of the ceiling above the floor lamp, and part of the table and the left wall, and the glass egg.

{4} Monte Carlo integration, in its basic form, works as follows. Given n samples, a function $g(x)$ to integrate, and limits i_1 and i_2:

- For each sample:
  - select a point $p \in [i_1, i_2]$,
  - compute $s = g(p)$,
  - multiply by the size of the region: $s \leftarrow s \cdot (i_2 - i_1)$,

- add it to the total sum $S \leftarrow S + s$.
- Return S/n.

In this case, we have $n = 1$, $g(x) = x^2$, $i_1 = 0$, $i_2 = 1$. If we pick $p = 0.5$, we have $g(0.5) = 0.25$, multiplied by one, so the approximation of the integral is $\int_0^1 x^2 \,\mathrm{d}x \approx 0.25$. (The true value is $1/3$.)

{5} The variance comes from stochastic sampling used in path tracing. Each sample is offset from the correct value, but the mean of all samples is at the correct value. The variance describes the spread around the mean.

The bias on ray tracing comes from the fact that not all information is used, which means that there is a systematic error (bias) between the computed value and the physically correct value.

An image with large variance will be noisy (looks grainy). An image with a large bias but small variance will be smooth, but have (slightly) incorrect colours (compared to a more correct physical model).

Question 4

Figure 3 shows an example of a hard shadow rendered in real-time using shadow volumes.

4 points

{3} Outline the steps of a naive *shadow volume* algorithm for rendering hard shadows cast by a polygon model onto a terrain object, given a point light source, such as the one in Figure 3.

4 points

{4} Discuss performance issues of your algorithm and give details of how it can be amended to be more efficient. Still only on a pseudocode level. (Alternatively, give details of what measures you took in the previous algorithm.)

4 points

{5} Shadow volumes are typically used for rendering hard shadows. Discuss how your shadow volume algorithm could be amended to render soft shadows (from an area light source), and discuss pros and cons relative to implementing soft shadows with shadow mapping instead.

Solution

{3} To get full points, the answer should include the following:

- Generate shadow volumes (by creating shadow triangles that extend from the edges of the actual geometry and away from the light source).
- For each pixel, count how many shadow volumes a ray through this pixel *enters*.
- Count how many shadow volumes the ray *exits*.
- If it enters more volumes than it exits, then the pixel is in shadow, otherwise not.

A simple shadow volume algorithm that uses the stencil buffer might look like this:

1. (Optionally, first render ambient light only.)
2. Turn of frame buffer and depth buffer.

3. Generate shadow volumes.
4. Render only the forward-facing shadow triangles, incrementing the stencil buffer with 1 for each triangle.
5. Render only the backward-facing shadow triangles, decrementing the stencil buffer with 1 for each triangle.
6. Turn on the frame and depth buffer, and render the scene as usual, using the stencil mask and rendering only those pixels where then mask $= 0$.

{4} A naive shadow volume algorithm generates more shadow triangles than necessary. One step to decrease the amount of extra geometry generated is to first extract the *silhouette edges* of the objects and only generate shadow volumes from them.

A naive algorithm might also do ray tracing for each pixel. Implementing it using the GPU stencil buffer (as above) makes it more efficient.

{5} For example, one could sample many point lights over the area light's surface and render shadow volumes for all of them. This is a brute-force way of rendering soft shadows, and would be slow, since it would require generating a lot of extra geometry.

Another idea would be to generate two sets of shadow volumes from the same point light (one inside the other), where points that are inside the innermost volume are in the umbra, and points that are inside the outermost volume only are in the penumbra. This might be faster than the above, but would (1) require additional work to compute the distance to the inner and outer shadow volume and (2) be susceptible to artefacts when a surface is shaded by two nearby or overlapping objects.

The first solution is (in the limit) physically correct, but slow. With shadow mapping, plausible (though not correct) soft shadows can be generated quickly with, for example, percentage closer filtering.

(a) Lambertian sphere.

(b) Full moon.

Figure 1: Comparing the light reflected off a Lambertian sphere and the moon.

Figure 2: Indoor scene rendered with bidirectional path tracing.

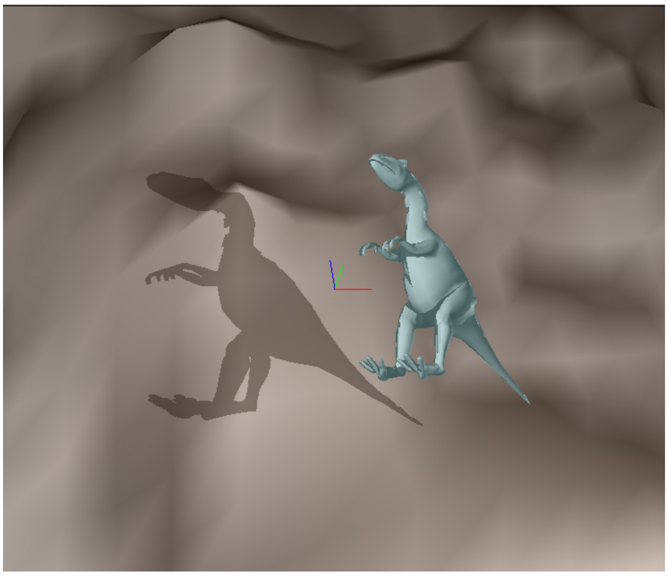


Figure 3: Hard shadow rendered with shadow volume.