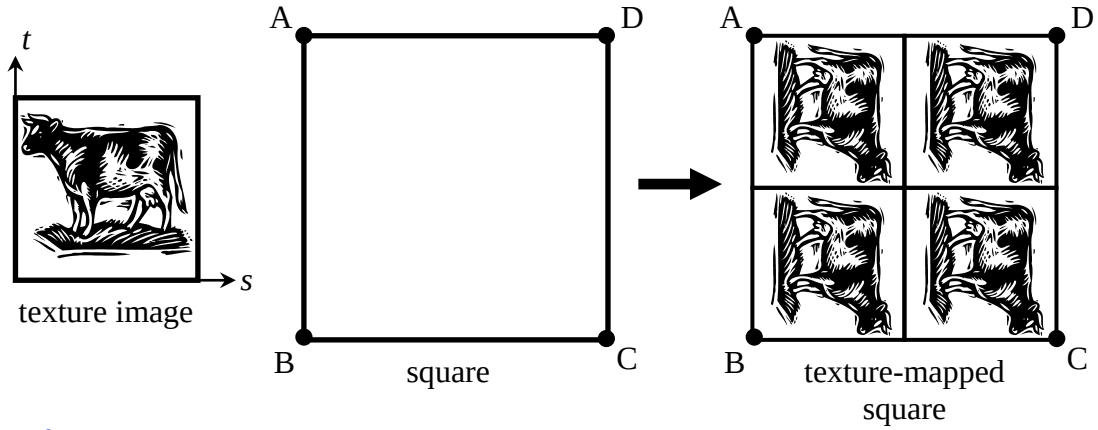
RI3004A 3D Graphics Rendering

Discussion 5 (Answers)

For **Lecture 8: Texture Mapping & Shadows**

Please attempt the following questions before you go to your discussion class. Some of the questions may be quite open-ended and some may be even ambiguous. In those cases, you are encouraged to make your own (reasonable) assumptions.

(1) Suppose the texture coordinate wrapping mode has been set to `GL_REPEAT` for both the *s* and *t* texture coordinates. Given a texture image and a square as shown in the following diagram, what are the 2D texture coordinates assigned to vertices A, B, C and D so that texture-mapped square appears as shown below?



One of the multiple answers:
A: [2, 0]
B: [0, 0]
C: [0, 2]
D: [2, 2]

(2) Given a 512×512 texture image, we want to create a mipmap from it and use the appropriate mipmap level during rendering.

(a) How many levels (including the original texture image) are in the mipmap?

$\log_2(512) + 1 = 10$

(b) The mipmap is used to texture-map a 3D square that appears in a 100×100 region on the screen. What is the best integer mipmap level to use to texture-map the square? Assume that we prefer a more blurred result if the exact level is not an integer. The highest-resolution texture image is level 0, the next is level 1, and so on.

$\text{ceil}(\log_2(512/100)) = 3$

(c) Some rendering systems can actually take non-integer mipmap level and compute the result by interpolating between two mipmap levels. In the case of Part (b), what is the exact mipmap level to use to texture-map the square? Round your answer to 2 decimal places. You can use the formula $\log_2(x) = \log_{10}(x) / \log_{10}(2)$.

$\log_2(512/100) \approx 2.36$

(3) **(a)** A reflective object can be rendered using reflection mapping or ray tracing. List two situations where there will be obvious differences between the images produced by the two methods.

(i) When there should be self-reflection. Reflection mapping cannot produce self-reflection.
(ii) When the reflective object is quite large compared to the size of its surrounding.

(b) Describe a way you can use to detect that the features on an object are actually bump-mapped instead of real geometry.

When we look at the silhouette of the object, if the features appear flat, most likely it is bump-mapped.

(c) Suppose you can extend the functionalities in any stage of the raster graphics pipeline. We want to render polygons with bump-mapping. **(i)** Should the lighting computation be performed per fragment, per vertex, or per polygon? **(ii)** At which pipeline stage should the lighting computation be performed?

(i) Per fragment.
(ii) Fragment processing stage. (Rasterization stage is OK too.)

——— End of Document ———