HW 1a Solutions

Andy Scukanec

September 15, 2003

1 Image Conversion - Problem 1

1.1 How much space is occupied by I_8?

$$(1280 * 1024) \text{ pixels } * \frac{8 \text{ bits}}{\text{pixel}} * \frac{1 \text{ byte}}{8 \text{ bits}} = 1.25 * 1024 * 1024 \text{ bytes}$$
$$= 1.25 \text{ MB } \approx 1.3 \text{ MB}$$

1.2 Questions about I_4

1.2.1 How much memory does I_4 occupy?

$$(1280 * 1024) \text{ pixels } * \frac{4 \text{ bits}}{\text{pixel}} * \frac{1 \text{ byte}}{8 \text{ bits}} = 1.25 * 1024 * 1024 * .5 \text{ bytes}$$
$$= .625 \text{ MB } \approx .6 \text{ MB}$$

1.2.2 How does it look different?

It has quantization bands visible because there are only 16 gray levels.

1.2.3 What technique might improve this?

Dithering.

1.3 Questions about I_{16}

1.3.1 How much memory does I_{16} occupy?

$$(1280 * 1024) \text{ pixels } * \frac{16 \text{ bits}}{\text{pixel}} * \frac{1 \text{ byte}}{8 \text{ bits}} = 1.25 * 1024 * 1024 * 2 \text{ bytes}$$
$$= 2.5 \text{ MB } \approx 2.5 \text{ MB}$$

1.3.2 What is the maximum pixel value that will result from this conversion?

In binary:

$$I8_{max} = 11111111_2$$
$$I8_{max} * 256 = 1111111100000000_2$$

In base 10:

$$(2^{16} - 1) - (2^8 - 1) = 65280$$

1.3.3 Give an alternate expression that will map full white to full white.

One decent answer is:

$$I_{16}[x, y] = \frac{(2^{16} - 1) * I_8[x, y]}{2^8 - 1}$$

which is another way of saying:

$$I_{16}[x, y] = 257 * I_8[x, y]$$

Any answer that maps full white to full white, full black to full black, and all other colors to something reasonable will be accepted. In particular, the function must be linearly increasing so that a brighter color represented in I_8 does not appear relatively darker in I_{16}.

Some people asked about mappings that would result in a mapping of full white to full white, but then failed to map full black to full black. These will not receive full credit, although they will receive some partial credit. According to the exact problem description, you could actually just map all colors to full white and technically solve the problem, but that would render the act of expanding from 8 bits to 16 bits useless!

2 Gamma Correction - Problem 2

2.1 What are the first and last five representable color values in each scheme represented as a percentage (to 2 digits) of L_M?

Given n, we must find a^γ for each scheme. This will be the percentage of L_M that comes through the monitor. I is a function whose exact value is known, and it is up to us to do something clever so that values of I are displayed correctly, or nearly correctly, on the monitor. I does *not* factor directly into this problem though, it was merely given so as to provide background information for the student to understand the real world applicability of this problem. One obvious problem in trying to work with I, is that it does not have some 'first 5 values'. This can only have any sort of meaning with a discrete variable. The only variable like that in the problem is n. n is the integer value from 0 to 255, that discretely approximates the value of I. How it does this differs depending on which scheme you are using. In the table below, I have calculated out a as well as a^γ. The student only needed to worry about the a^γ columns.

$$a(n_1) = \sqrt{\frac{n_1}{255}}, \gamma = 2 \Rightarrow a^\gamma(n_1) = \frac{n_1}{255}$$
$$a(n_2) = \frac{n_2}{255}, \gamma = 2 \Rightarrow a^\gamma(n_2) = \frac{n_2^2}{255^2}$$

	Scheme 1		Scheme 2	
n	a	a^γ	a	a^γ
0	0.0%	0.0%	0.0%	0.0%
1	6.3%	.39%	.39%	.0015%
2	8.9%	.78%	.78%	.0062%
3	11%	1.2%	1.2%	.014%
4	13%	1.6%	1.6%	.025%
251	99%	98%	98%	97%
252	99%	99%	99%	98%
253	100%	99%	99%	98%
254	100%	100%	100%	99%
255	100%	100%	100%	100%

2.2 Dark vs. Light representation

2.2.1 In each scheme, what pixel values straddle the intensities at 1% and 99%?

In order to actually 'straddle' an intensity, we need to find an n which lies immediately below and above the appropriate intensity. This problem is essentially the opposite of the above problem. Rather than being given an n and told to find a percentage, we are being given a percentage (a^γ) and told to find n.

Scheme 1:

$$a^2 = .01, n_1(a) = 255 * a^2 \Rightarrow n_1 = 255 * .01 = 2.55 \Rightarrow n = 2, 3$$
$$a^2 = .99, n_1(a) = 255 * a^2 \Rightarrow n_1 = 255 * .99 = 252.45 \Rightarrow n = 252, 253$$

Scheme 2:

$$a^2 = .01, n_2(a) = 255 * a \Rightarrow n_2 = 255 * \sqrt{.01} \Rightarrow n = 25, 26$$
$$a^2 = .99, n_2(a) = 255 * a \Rightarrow n_2 = 255 * \sqrt{.99} \Rightarrow n = 253, 254$$

2.2.2 Which scheme can distinguish smaller variations in dark shadows?

Scheme 2 can. The difference between n_2=25 and n_2=26, the two values that straddle 1% intensity in Scheme 2 $\approx$ .08%. The difference between n_1=2 and n_1=3, the two values that straddle 1% in Scheme 1 $\approx$ .39%.

2.2.3 Which scheme can distinguish smaller variations in bright lights?

Scheme 1 has the percentage values more closely bunched up towards the 100% end of light intensity, and can distinguish brighter lights better than Scheme 2. The same argument in the above question can be applied here.

2.2.4 In which case can the human visual system make more use of precision and why?

The human visual system can distinguish *percentage* change in light, and hence can distinguish changes in darker lights much better. Thus, scheme two is the better choice.

2.3 Assume we use 8-bit pixels in scheme 2. How many bits of precision would we need in scheme 1 to match the performance of scheme 2 at 1% intensity?

We need to add bits to scheme 1 until the difference in the two representable intensities that straddle 1% is less than that of Scheme 2, or .08% (call this number δ). In scheme 1 with b bits, $L = n_2/(2^b - 1)$, so the difference between subsequent values of n_2 is always $1/(2^b - 1)$. We want this difference to be less than δ so we have:

$$\begin{aligned} \delta &> 1/(2^b - 1) \\ 2^b &> 1/\delta - 1 \\ b &> \log_2(1/\delta - 1) = \log_2(1/0.0008 - 1) = 10.3 \end{aligned}$$

Therefore the number of bits must be greater than 10.3, or at least 11.

3 Alpha Channel - Problem 3

3.1 What will the pixel values after 1, 2, 3, 4, and 5 iterations be with the an alpha channel of 0, 15, 128, 240, and 255?

The answer to this question is a 5 by 5 grid. In my grid, I have the number of the compositing operation going down, and the changing alpha values going across. The only pertinent formula for this problem is $c = \alpha * c_f + (1 - \alpha) * c_b$. $c_f = 255$ since the problem description stated that it was full white, and $c_b = 0$ initially, since it started as full black. c_b will change each iteration. The other important thing to remember is that $\alpha \in [0, 1]$. Thus, we have to divide α by 255 before we actually use it in the equation. Below is the table that emerges. I rounded to the nearest to resolve fractional values, although any consistent rounding scheme is fine.

Iteration	Alpha Values				
	0	15	128	240	255
1	0	15	128	240	255
2	0	29	192	254	255
3	0	42	224	255	255
4	0	55	240	255	255
5	0	67	248	255	255

3.2 After a large number of compositing operations, how will the image look different?

The circle will have hard, not soft edges. Specifically, any pixel not specified as solid black in the alpha channel will eventually converge to be solid white.

3.3 If the alpha channel shown above is meant to approximate a sharp edged opaque circle, is this behavior appropriate?

No - if the circle is antialiased in order to approximate a hard edged circle, then many composites will remove just such an antialiasing, and we will be left with the same sort of visual artifacts that antialiasing attempts to remove.