

In this homework, you will develop a MIPS assembler program that reduces the resolution and color depth of digital images.

A. Load and save images

An image can be stored in the memory as an array by continuously storing the individual lines with image points one behind the other. At the beginning of the image there is also a header that contains the width, height and color depth, so the number of colors.

Your program should be able to read and write image files in the PGM format (binary, P5). For example, the string "P5 2 2 255 \00 \00 \00 \00" describes a gray scale image in binary format with 2×2 black pixels and "\nn" describes a byte with the hexadecimal value 0xnn.

You must calculate the value of the number from a string of digits. You need an ASCII table for this.

1. Implement a load_img routine that loads an image and processes the header.
2. Also implement a store_img routine that obtains the height and width of the image as well as a pointer to the user data to create an image file.

B. Decrease image resolution

In this task the resolution of an image is to be reduced.

Since the resolution in each dimension changes by a factor of two, you can calculate a new pixel from 4 adjacent pixels. For example, the image a0 having a resolution of 2×2 pixels can be calculated from the image a with a resolution of 4×4 pixels.

a =

....
....
....
....

, a0 =

:: ::
:: ::

Remember that the color values must also change with the resolution.

1. Implement an interpolate2 routine that reduces the resolution of an image by a factor of 2. The routine receives as arguments the memory address as well as the height and width of the image.
2. Implement a routine interpolate that reduces the resolution of an image by a factor of 2^n .

C. Decrease the color depth

In this task the color depth of an image is to be reduced. The given images have a color depth of 8 bits. The reduction of the color depth should also be possible in powers of 2 (8 bits, 4 bits, 2 bits, 1 bit). That is, instead of 2^8 e.g. Only 2^4 grades are available.

Implement a quantize routine that takes as arguments the memory address of the image, the width/height of the image, and a quantization factor.