

CoE 3SK3 Course Project

Image Interpolation

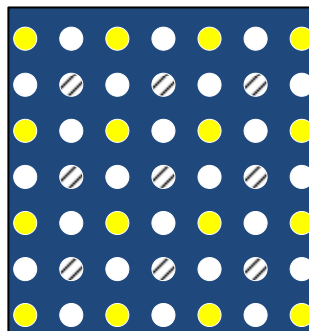
Due date: April 1, 2011.

When displaying low-resolution images on high-resolution monitors, for example, watching standard definition TV programs on high definition TV sets, the viewers or players have to interpolate the missing pixels as illustrated by the following figure. In the figure, the yellow dots are the pixels of the low-resolution image and the white dots are the missing pixels to be interpolated.

- (a) Derive the bicubic interpolation algorithm that is described by the following formula:

$$f(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{i,j} x^i y^j$$

where $a_{i,j}$ are the coefficients of the bi-cubic interpolation function. The two-dimensional function $f(x, y)$ interpolates the low-resolution pixels, and it is used to generate the missing pixels.



- (b) Generalize cubic splines of Section 18.6.3 (p. 515, textbook ed. 6) from one dimension to two dimensions for the application of image interpolation.
- (c) Implement the image interpolation algorithms derived in (a) and (b) (either in C or in Matlab). Your programs need to read an image, perform image interpolation, and then display the interpolated image.
- (d) Image interpolation algorithms are also useful to resample digital images at arbitrary positions (non-integer grid points). Apply the two interpolation algorithms of (a) and (b) to perform image rotation by an arbitrary angle.

- (e) Compare the algorithms of (a) and (b) in interpolation precision. We will give you both the high-resolution (ground truth) and low-resolution versions of test images.
- (f) Make your implementations as efficient as possible. What is the average number of operations needed to interpolate a missing pixel for algorithm (a) and algorithm (b)?
- (g) (option) Make your implementations applicable to arbitrary scaling factors.