Colorado School of Mines

Image and Multidimensional Signal Processing

Professor William Hoff
Dept of Electrical Engineering & Computer Science
http://inside.mines.edu/~whoff/
Spatial and Intensity Resolution
Spatial Resolution

- Can express as the number of pixels (e.g., #rows x #columns)

For easier comparison, we can zoom the smaller images back to the original size.

**FIGURE 2.19** A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.
• Re-expanding (by replication) to the original size

**FIGURE 2.20** (a) 1024 × 1024, 8-bit image. (b) 512 × 512 image resampled into 1024 × 1024 pixels by row and column duplication. (c) through (f) 256 × 256, 128 × 128, 64 × 64, and 32 × 32 images resampled into 1024 × 1024 pixels.
Gray Level Resolution

- The number of gray levels; in integer images usually a power of 2
- \( L = 2^k \)

\[
\begin{align*}
\text{k = 8, 7, 6, 5} & \quad \text{vs.} \quad \text{k = 4, 3, 2, 1}
\end{align*}
\]
Example using Matlab

- Load gray scale image, reduce gray level resolution, display it

- To reduce resolution from 256 gray levels to 16 gray levels:
  - Can divide every value by 16 (this rounds to integer)
    - \( I_2 = I / 16; \)
    - This reduces range from 0.255 to 0.15
  - Or can do a logical AND of every pixel with the binary number \( 11110000 \)
    - \( I_2 = \text{bitand}(I, 240); \)
    - This zeros out least significant four bits

- How to reduce resolution to 4 gray levels?
Example using Matlab

• Load gray scale image, reduce spatial resolution, re-expand it, display it

• To reduce spatial resolution by a factor of two in each dimension:
  – Sample every other row, every other column
    • \( I_2 = I(1:2:end, 1:2:end); \)
  – Or use Matlab’s resize function
    • \( I_2 = 	ext{imresize}(I, 0.5); \)
Storage

• If an image is NxN pixels
  – Each pixel has up to $2^k$ gray levels
  – Total number of bits stored = $N \times N \times k$ (with no compression)

• To reduce storage, is it better to reduce $N$ or $k$?
  
  96x96x5 = 46080 bits
  107x107x4 = 45796 bits
Live demo using USB camera and OpenCV

• OpenCV
  – A library of C/C++ functions for real time computer vision
  – Free from http://opencv.willowgarage.com/wiki/
  – Documentation online at http://opencv.itseez.com/
  – Some books available

• Not as easy to use as Matlab (but Matlab not good for realtime)
/* Demonstrate effects of spatial resolution and intensity quantization
Bill Hoff
August 2012*/
#include <iostream>

#include <opencv2/opencv.hpp>

#include <opencv2/opencv.hpp>

////////////////////////////////////////////////////////
// These are adjustable parameters, to be set using sliders
double g_bpp = 8.0;       // bits per pixel
int g_bpp_position = 8;
void g_bpp_trackbar(int, void*)
{   g_bpp = double(std::max(0,g_bpp_position));   }

double g_imgscale = 1.0;   // scale size of the image, 0..1
int g_imgscale_position = 100;
void g_imgscale_trackbar(int, void*)
{   g_imgscale = double(std::max(0,g_imgscale_position))/100;   }

////////////////////////////////////////////////////////

int main(int argc, char* argv[])
{
    cv::VideoCapture cap(0); // open the default camera
    if(!cap.isOpened()) { // check if we succeeded
        printf("error - can't open the camera\n");
        system("PAUSE");
        return -1;
    }

double WIDTH = cap.get(CV_CAP_PROP_FRAME_WIDTH);
double HEIGHT = cap.get(CV_CAP_PROP_FRAME_HEIGHT);
printf("Image width=%f, height=%f\n", WIDTH, HEIGHT);

    cv::Mat imgInput;
    cv::Mat imgInputGray( cv::Size(WIDTH,HEIGHT), CV_8UC1);
    cv::Mat imgResize( cv::Size(WIDTH,HEIGHT), CV_8UC1);
    cv::Mat imgQuantize( cv::Size(WIDTH,HEIGHT), CV_8UC1);
    cv::Mat imgTemp;
printf("Hit ESC key to quit ...
");

// Create image windows. Meaning of flags:
// CV_WINDOW_NORMAL enables manual resizing; CV_WINDOW_AUTOSIZE is automatic
// You can "or" the above choice with CV_WINDOW_KEEPRATIO, which keeps aspect ratio
cv::namedWindow("Input image", CV_WINDOW_AUTOSIZE);
cv::namedWindow("Resized", CV_WINDOW_AUTOSIZE);
cv::namedWindow("Quantized", CV_WINDOW_AUTOSIZE);

// Create trackbars
// These are so we can play with the parameters
cv::createTrackbar( "g_bpp", "Quantized", &g_bpp_position, 8, g_bpp_trackbar );
cv::createTrackbar( "g_imgscale", "Resized", &g_imgscale_position, 100, g_imgscale_trackbar );

while (1){
    cap >> imgInput;// get image from camera
    if (!imgInput.data){
        printf("error - image is empty
");
        system("PAUSE");
        return -1;
    }

    // Convert to gray
    cv::cvtColor(imgInput,imgInputGray,CV_BGR2GRAY);
cv::imshow("Input image", imgInputGray);

    // Change bits per pixel
    double scaleFactor = pow(2.0,g_bpp)/256.0;
    imgInputGray.convertTo(imgQuantize, CV_8UC1, scaleFactor); // reduce scale
    imgQuantize.convertTo(imgQuantize, CV_8UC1, 1/scaleFactor); // restore scale
    cv::imshow("Quantized", imgQuantize);
// Change image size
// Options for resizing:
// CV_INTER_NN Nearest neighbor
// CV_INTER_LINEAR Bilinear (default)
// CV_INTER_AREA Pixel area re-sampling
// CV_INTER_CUBIC Bicubic interpolation
resize(imgInputGray, imgTemp, cv::Size(), g_imgscale, g_imgscale, cv::INTER_NEAREST); // reduce size
resize(imgTemp, imgResize, cv::Size(), 1/g_imgscale, 1/g_imgscale, cv::INTER_NEAREST); // restore size
cv::imshow("Resized", imgResize);

// wait key for x ms (0 means wait until keypress)
if (cv::waitKey(1) == 27)
break;// ESC is ascii 27

//system("PAUSE");
return EXIT_SUCCESS;
Summary / Questions

• Spatial resolution can be expressed in terms of the number of pixels in the image.

• Intensity resolution can be expressed as the number of gray levels in the image.

• Can you increase the spatial resolution of an image by just replicating the pixels?