1 Basic image input/operations in Octave

Preparation. First, let's learn how to define functions in Octave. Here's a simple function that returns the double of the input value. When writing functions in an Octave session within TeXmacs, use Shift+Enter to get a new line, and use Enter to complete the definition.

```
octave> function d=dbl(x)
          d=2*x;
        endfunction;
octave>
```

Test that the function works

```
octave> dbl(2)

   4

octave>
```

Next, let's learn how to work with strings. Strings are enclosed in quotes, and can be assigned as values to variables. You can join multiple strings with the `strcat` function.

```
octave> astr="a"; bstr="b"; cstr="c";
octave> strcat("a","b","c")

   abc

octave> strcat(astr,bstr,cstr)

   abc

octave>
octave>
```

You convert numbers to strings using the `num2str` function.

```
octave> x=2.2; strcat("The answer is x=",num2str(x))

   The answer is x=2.2

octave>
```

You can control the format of how a number is converted to a string using C-language formatting directives.

```
octave> num2str(2,"%2.2d")

   02

octave>
```

Using the above, let us define a function to read a face from the yalefaces database

```
octave> function img=readface(n,type)
          fhead = "/home/student/courses/MATH547/lessons/yalefaces/subject";
          fnr = strcat(num2str(n,"%2.2d"),".");
          fname = strcat(fhead,fnr,type);
          [img,map,alpha] = imread(fname);
        endfunction;
```

```
octave>
```

Parse the above to understand how things work. A string is formed for the file name using a header, the number of the image we want, and the type of the image. The image is read from the file and returned. We can find out its size and set the number of height, width pixels (h, w). Look at a portion of the matrix to see that an image is simply an array of gray-value intensities with the value 0 denoting black and the value 255 denoting white. We define a function to reformat an image as a vector with entries between 0 and 1.

```
octave> f01n=readface(1,"normal");

   warning: your version of ImageMagick limits images to 16 bits per pixel

octave> [h,w]=size(f01n);
octave> disp([h w]);

      243    320

octave> f01n(97:103,97:103)

     255   255   186    83   135   153    79
     255   248   111   149   192   104    62
     255   224   164   251   223   102    46
     255   235   202   255   203    93    45
     255   255   255   246   136    58    50
     255   255   255   231   123    52    51
     255   255   255   175    79    50    58

octave> function vec=img2vec(img)
          [h,w]=size(img); m=h*w;
          vec=zeros(m,1);
          vec=double(reshape(img,m,1))/255.;
        endfunction;
octave> v01n=img2vec(f01n);
octave>
octave>
```

Now let us form a matrix of the normal face images $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, with $m = hw = 320$, and $n = 15$ subjects. First create a matrix of the appropriate size with zero entries everywhere.

```
octave> m=h*w; n=15; A=zeros(m,n);
octave> rank(A)

   0

octave>
```

Now use a loop to fill each column, with image gray level rescaled to be between 0 and 1.

```
octave> for j=1:n
          fj=readface(j,"normal");
          A(:,j) = img2vec(fj);
        end;
octave>
```

At present there is no command from within Octave to embed an image directly into TeXmacs. This has to done manually by using the menu option Insert->Image->Insert image ... dialogue. Here are the 15 “normal faces”

Figure 1. The "normal" faces from the Yale face data base.

Since those are 15 rather different people, we should expect the column vectors of the matrix $\boldsymbol{A}$ to be linearly independent as confirmed by computing the rank

```
octave> rank(A)

   15

octave>
```

Let us construct a linear combination of some faces

```
octave> newf=0.5*A(:,1)+0.5*A(:,3);
octave>
```

Define a function that saves the vector to an image file on disk

```
octave> function writeface(f,h,w,name)
          i = reshape(f,h,w);
          fname = strcat("/home/student/courses/MATH547/homework/",name,".png");
          imwrite(i,fname);
        endfunction;
octave> writeface(newf,h,w,"newface");
octave> format short;
octave>
```

The image has been written to a the file newface.png within your homework directory. Here it is (Insert->Small figure followed by Insert->Image->Insert image ...):

Figure 2. Linear combination of subject01 and subject03.

Just from the above you notice there are additional issues that arise:

- image alignment
- feature alignment

These are interesting open research problems. For now start investigating practical application of the concepts learned so far by answering the following.