Problem set III: Image visualization, analysis and animation using Visual Basic

Please turn in one of these exercises on a floppy disk. Your exercise should be completed by Friday, November 12. Note that you need to sign up for one of the exercises on the distribution sheet circulated in class: no more than 3 can sign up for any one problem. These exercises involve the visualization and analysis of (mostly simplified) biological images.. Please use any objects like picture boxes, text boxes, command buttons, etc. as appropriate for creating an effective graphical user interface for your model. The images xray.bmp, marker1.bmp and marker2.bmp are posted on the website in the file “Problem Set 3 extra data files”. (The other data files have already been posted in the Image Data Files directory). Each of the five exercises to be done in the course, including this one, is worth 25 points, based on the following values: Does the program follow the specifications and description provided in the statement of the problem? (5 points). Does the program run without crashing and provide the expected output? (10 points). Is the Graphical User Interface well designed and easy for the user to understand? (5 points). Are your documentation and comments adequate? (5 points) Was the exercise turned in on time? (5 bonus points) You are expected to work on these assignments independently. The disk you turn in should be labeled clearly with your name, problem set number. Be sure to keep backups of all your work in case your disks are lost or damaged!

1. Write a program to measure the average velocity of movement of the marker moving in a video field, using image files Marker1.bmp and Marker2.bmp. These images are 300 by 300 pixels in size, and were taken 1 second apart.. The calibration constant is 0.257 μm per pixel. Please create a GUI that displays the two images, and a textbox to show the calculated velocity to the nearest tenth of a μm.

2. Write a program to apply pseudocolor to the image file "Imgdata2.bmp" so that each object in the image has a unique color assigned to it. (However, if two objects have the same optical density, they'll have the same pseudocolor assignment.) Design your GUI so that the standard 16 VB pre-defined colors can be assigned to the entire 8 bit intensity spectrum of the original monochrome image, and then tested on the image. Use two picture boxes to display the original and the processed images side by side.

3. Use the mouse to set the screen position and height to display the xray.bmp record provided. (Given the height, the width should be scaled proportionately so as to not disturb the x-y aspect ratio of the image.)

4. Display the xray.bmp record on the left side of the screen; this record is 250 pixels wide by 350 pixels high. Use the mouse to zoom in on a selected small segment of this record which is then blown up to fill a large window occupying the entire right half of the screen.

5. Write a program to count the number of objects contained in the graphics file ImgData1.bmp. Locate and list the center of gravity of each object. All objects are contained in the range of screen pixel x and y coordinates from 1 to 300. All of the objects are black on a white background.

6. Write a program to measure the average image densities of the objects contained in the graphics file ImgData2.bmp. All objects are contained in the range of screen pixel x and y coordinates from 1 to 300. All of the objects are various shades of gray on a white background.

7. Write a program to enhance the contrast of the objects contained in the graphics file ImgData3.bmp. All objects are contained in the range of screen pixel x and y coordinates from 1 to 300. All of the objects are in various shades of gray on a white background.

8. Write a program to measure the perimeter (in pixels) of the single object contained in ImgData4.bmp. Display the result in a text box.

9. Animate a simple drawing of a water molecule (a color-filled circle representing the oxygen atom, two smaller color-filled circles connected by lines between the circles' centers, representing the covalent bonds). Make the molecule move from the lower left quadrant to the upper right quadrant of the screen, with the molecule rotating slowly and smoothly as it moves across the screen.

10. Simulate the random movement of two identical molecules of gas bouncing off the walls of a container. Animate the random motion of these two idealized objects (colored circles against a white background, one red, the other green), remembering that the angle of reflection equals the angle of incidence when each object strikes the wall. Be certain that the molecules collide rather than pass over or through each other if they try to occupy the same space.

11. Analyze "Imgdata6.bmp" to count the number of objects in two classes (circles and squares). Your GUI should display the source image, and two labeled textboxes should report the number found in each class of objects. Both types of objects are black against a background of white.

12. Produce a histogram of the distribution of objects (circles of different sizes) found in "Imgdata7.bmp". The objects are bright white (pixel intensity = 255) against a background of pure black (intensity = 0). The histogram should show the frequency of occurrence of objects of each size, in a range of about 50 square pixel intervals (e.g. "bins" between 0 and 50, 51-100, 100-150, etc.).