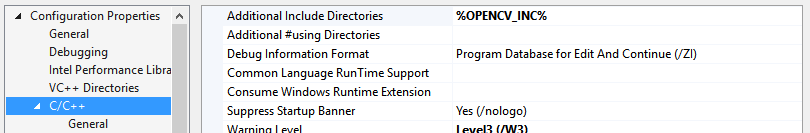
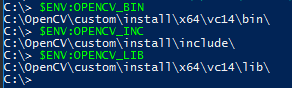
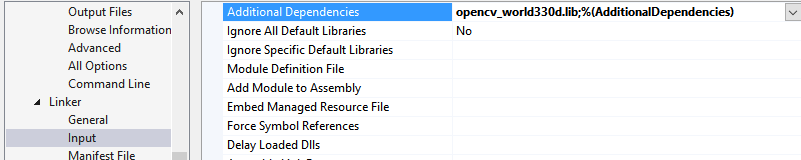
Instructions for **Building** the project included in this ZIP folder:

1. It is assumed the user is familiar with the PDF document “[OpenCV\_Instructions\_Windows\_Fall17.pdf](https://myasucourses.asu.edu/bbcswebdav/pid-19118610-dt-content-rid-135205475_1/xid-135205475_1)”. A copy is provided in the main directory for reference.
2. It is also assumed that the user has compiled from sources, their own OpenCV Libraries including at least the “xfeatures2d” module from opencv\_contrib.
3. To build and compile, start by opening the provided solution “OpenCVSolution.sln”.
4. The project used for this assignment is called “HandOn2”. To get this project ready for build, right click on it and open the properties menu.
5. For convenience, the project parameters use System Variables, but could easily be replaced with full-paths to each directory.  
   1. In C/C++ General Properties, Additional Include Directories contains the **OPENCV\_INC** variable which is an absolute path that should point to your OpenCV “include” directory.  
      
   2. In the Linker properties, Additional Library Directories contains the **OPENCV\_LIB** variable witch is an absolute path which should point to the “lib” directory in your OpenCV deployment.  
      
6. The path variables are easily set in windows like so:  
   
   1. The variables on my system at the time of writing:  
      
   2. However, populating the project settings with appropriate paths should achieve the same goal.
7. Finally, include the **opencv\_world330d.lib** library file from your OpenCV distribution in the **Liner🡪Input🡪Addition Dependencies** project setting.  
   
8. Build and Run the project as normal, I prefer the “**F5**” key for consistency.

Instructions for **Using** the program:

After launching the application, simply watch the output window for details of which image is being updated and when. The output windows will include the original bird image, three transforms for the Laplacian, Sobel, and Canny Edge Detector, and finally the distance transform of each.

Results



Figure The original Bird image



Figure The cv::Laplacian transformed image, with Kernel=5 to improve contrast.



Figure the cv:Sobel transform of the bird image, using dx,dy=1 to identify the edges I care about, kernel=5 found the sharpest edges, and a scale factor of 3 brought he perceived intensity up to par with the Laplacian transform.



Figure The bird image transformed with the Canny Edge Detector using thresholds 50 and 100, giving nice sharp edges.



Figure The distance transformed version of the (value-inverted) Laplacian bird.

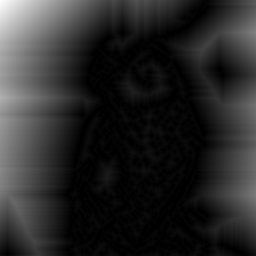


Figure the distance transform of the Sobel transformed bird.



Figure The distance transformed version of the canny bird.