Please submit your written answers at the start of class on Thursday.

1. Image warping and invertible transformations

Given a digital image, and an invertible transformation \( \tilde{H} \) of the form

\[
\tilde{p}' = \tilde{H}\tilde{p}
\]

we would like to compute the warped image whereby each point \( \tilde{p} \) in the original image is transformed to its new location \( \tilde{p}' \). This type of image warping is exactly what the \texttt{cv2.warpPerspective} function does, for example.

Here is a straightforward (but bad) algorithm for performing this image warp: for each pixel location \( \tilde{p} \) in the input, copy the color found at \( \tilde{p} \) in the source image to the pixel location nearest to \( \tilde{p}' \) in the output.

However, the vastly preferable algorithm is to loop over the destination pixels \( \tilde{p}' \) in the warp image, and use the inverse transformation \( \tilde{H}^{-1} \) to identify the nearest pixel \( \tilde{p} \) in the source image and copy the color from that source pixel to the destination.

What is the difference between the two approaches? Why is the second one preferable? 

*Hint:* consider the scaling transformation

\[
\tilde{H} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]

2. Normalized correlation and template matching.

Download and run the \texttt{findx} example from the course website. Then, read the documentation for the OpenCV functions \texttt{cv2.matchTemplate()}, \texttt{cv2.rectangle()}, and \texttt{cv2.minMaxLoc()}. Finally, use the information you learned to answer these questions.

a. What is the purpose of \texttt{matchTemplate} here? What task are we trying to accomplish?

b. One of the \texttt{rectangle} calls is necessary for the operation of the program, and the other is for display only. Which is which, and why? (*Hint: you can find out by commenting out one or the other...*)

c. What is the \texttt{r} variable controlling? What happens if you set it to 1? To 500?