Todo 8 and 9 find the necessary size of Acc so it fits all the images, and adjusts the transforms (M) to map all images into the common Acc coordinate system.

Todo 10 warps a given image and adds it to the accumulator, using inverse warping: see also Log slides 34-39

for each \([x', y']\) in Acc, compute input image coordinates:

\[
\begin{bmatrix} x \\ y \end{bmatrix} = M^{-1} \begin{bmatrix} x' \\ y' \end{bmatrix}
\]

\(\text{Sue}(x, y)\) is not an integer pixel location, use bilinear interpolation to determine source pixel value, then fill into Acc\([y, x]\).
Start with: just round $\left\lceil \frac{y}{w} \right\rceil$ to integers to do nearest neighbor interpolation and ignore blend width.

For proper blending, pixels need to get added to Acc with associated weights. If a pixel in Acc has multiple images overlapping, the final pano value is

$$\frac{\alpha_1 p_1 + \alpha_2 p_2 + \ldots + \alpha_k p_k}{\sum_i \alpha_i}$$

i.e. the weighted average of pixel values, normalized so total weight = 1.

The weights are determined by blend width:

The accumulator collects total weight in its 4th (alpha) channel.

Todo II simply divides Acc by the total weight in the alpha channel.
Tips:

- Lecture 9 has example code for inverse warping.
  - with I in 4th channel

- Make each input image 4 channels, then scale pixels according to the blending function before interpolation.
  - Then, Acc gets +\( x; p \); in the first 3 channels, and +\( x; i \) in the 4th.

- Start by just overwriting values in Acc; then worry about blending and feathering after the inverse warping works.