1. **Pre-processing:**
   1. **Resize image with same W/H ratio, optional**
   2. **Convert to gray**
   3. **Gaussian blur to remove high frequency noise**
   4. **Enhance contrast**
2. **Find seismic section:**
   1. **Convert to binary with adaptive or other threshold**
   2. **Find and retain only largest object in binary image**
   3. **Fill holes**
   4. **Apply opening and dilation to remove minutiae (tick marks and labels).**
3. **Define rectification transformation:**
   1. **Detect contour of largest object from (2)**
   2. **Approximate contour with enough tolerance to ensure it has 4 sides only**
   3. **Sort contour corners using angle from centroid**
   4. **Define new rectangular image using length of largest long and largest short sides of initial contour**
   5. **Estimate and output warp transformation from DST to SRC corners**
4. **Warp using transformation. N.B. to be applied to the final, recovered and recolored section**
5. **Optional impainting (to fill polygon holes left by annotations). This also should be done on the final, recovered and recolored image:**
   1. **Start with output of (4)**
   2. **Pre-process (including compressor?) and find canny edges**
   3. **Find contours smaller than input size**
   4. **Sort contours (by shape/position/n edges)**
   5. **Loop over contours:**
      1. **Approximate contour**
      2. **If approximation has 4 points AND**

**(4 semi-diagonals are of same length OR 4 angles are equal):**

* + - * **Fill contour using either skimage.draw.polygon or .\_pnpoly grid\_points\_inside\_polygon**
      * **Add to mask**
  1. **Use mask to remove text inside rectangle in the input and impaint the whole rectangle**

1. **Optional tools to remove arrows and circles/ellipses**
   1. **For arrows: contours from (4) find ones with 7 sizes and low convexity (concave) or alternatively harris corner and count 7 corners, or template matcing**
   2. **For ellipses:**
2. **Optional FFT filters to remove timing lines and vertical line**