AUTOMATED GRAIN BOUNDARY DETECTION

The grain boundary detection software consists of two critical pieces. The first is an automated (or at this point, semi-automated) process that provides an initial guess for the locations of the grain boundaries in a given field of view. The input for the algorithm is a set of images taken at different tilts of the same field of view. Its output is a binary 'skeleton' that consists of a reasonably accurate guess at the grain boundaries. The second important functionality of the software consist of tools that enable a user to easily edit this skeleton by both deleting erroneous boundaries and adding missing ones.

Below are the essential ingredients of the skeleton generating algorithm. Steps (2)$\rightarrow$(5) are almost entirely automatic, however steps (1) and (6) require user interaction.

(1) **Registration**: The user provides the software with multiple images taken at different tilts of the same field of view. These images are aligned and then combined into vectorial data (see figure 1). Registration can be done by selecting reference points in the set that the software will then automatically match, or by simply overlaying and aligning the images by hand.

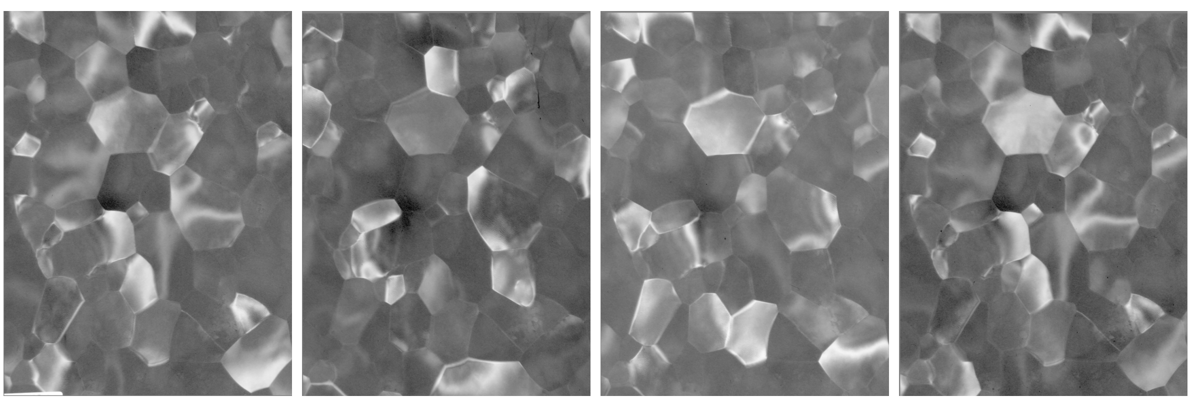

Figure 1. Aligned image set of a single field of view a polycrystalline aluminum sample

(2) **Smoothing**: Smoothing is performed on the vectorial image to remove 'noise' in such a way that homogeneity is encouraged inside the individual grains while contrast is preserved between neighboring grains. The current smoothing process is based on the nonlinear, anisotropic diffusion model

$$\frac{\partial I}{\partial t} = \operatorname{div}\left(D(\nabla I_\sigma)\nabla I\right),$$

where D controls the diffusivity. The solution, I, is the smoothed version of the image. Our diffusivity uses likely edge information to determine the type of anisotropy as well as the speed of the diffusion at each location in the image sets. It also has two key features that are specific to the grain boundary problem. The first is a mechanism for enhancing likely edges while suppressing false ones that arises from e.g. bend contours in the sample. The second is the fact that it incorporates information from the entire vectorial image (that is, information at each tilt) when processing each of its components.

(3) **Edge Detection**: A combination of first and second order edge detectors are then applied to the the smoothed vectorial image to get a fairly accurate description of likely boundaries. Likely edge information from all components of the vectorial image is combined to create a single edge map (see figure 2).

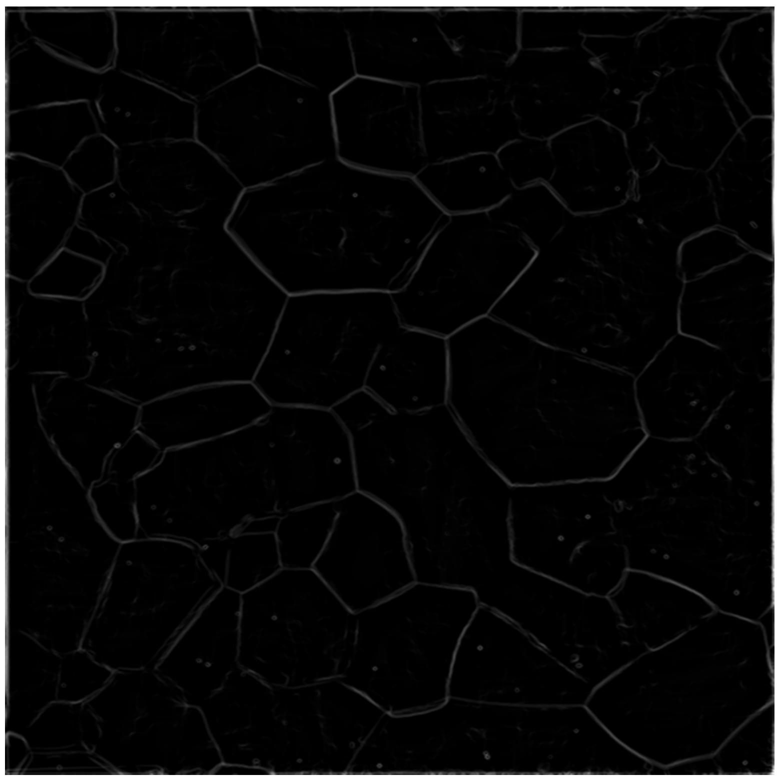

FIGURE 2. edge map: result after smoothing and edge detection

(4) **Morphological Operations**:
   (a) The first binary approximation of the grain boundary network is extracted from the solution of a slightly modified version of the geodesic active contour model from 'Geodesic Active Contours', by Caselles, Kimmel, Sapiro. In particular, the boundary is the zero level set of the solution of the model:

$$\frac{\partial u}{\partial t} = g(I)|\nabla u|\text{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \nabla g(I)\nabla u,$$

   Our modification comes from both inputting a vectorial image (instead of a scalar valued image) I, and using a monotonically decreasing edge detector, $g(I)$, that incorporates both first and second order edge information. Furthermore, we handle triple junctions by allowing a single level set function to 'wrap around' likely edges; a simple thresholding at zero (instead of extracting the zero level set) gives a natural dilation, resulting in a network of boundaries that is several pixels thick (see figure 3).

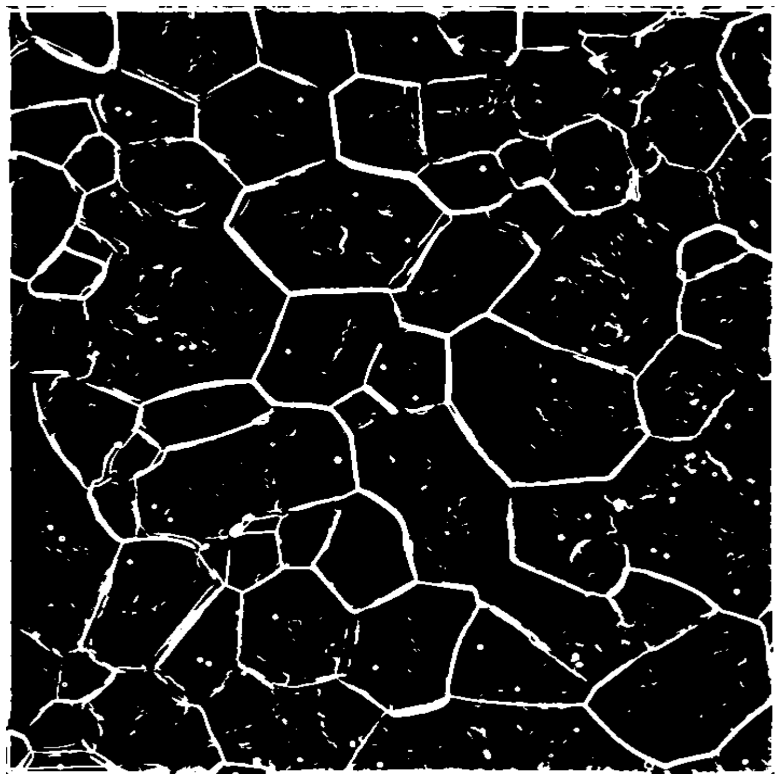

FIGURE 3. result of the active contour model

(b) A series of morphological operations are then used to obtain a skeleton, that is, a one pixel thick description of the grain boundaries. These operations include denoising, dilation, and erosion. The denoising step removes isolated clusters of pixels that fall below a pre-determined size. Dilation then fills in and smoothes out the remaining boundaries. Finally, topology preserving thinning yields a one-pixel thick binary skeleton of the grain boundaries (see figures 4 and 5).

(5) **Interpolation**: The skeleton resulting from the morphological operations is a good description of the boundaries, but usually the description is incomplete. To complete this description, an interpolation step is applied (see figure 6). The interpolation step uses information about the structure of the grains in a given metal and only connects corners (boundaries locations with sufficiently high curvature) and endpoints (locations where a boundary suddenly terminates). The main criteria for filling in these missing boundaries are as follows:

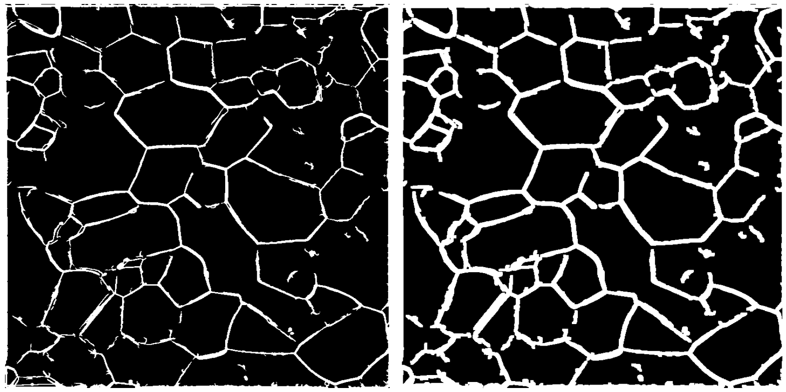

FIGURE 4. from left to right: results after denoising and dilation respectively

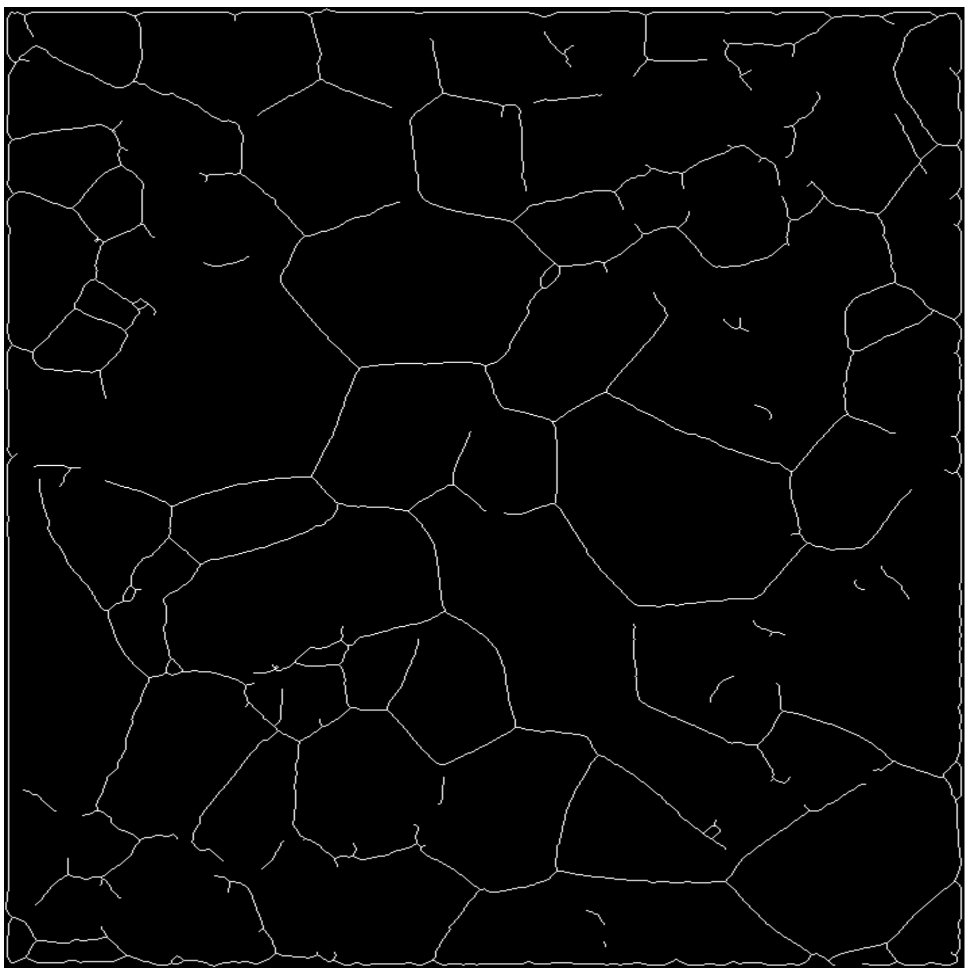

FIGURE 5. skeleton resulting from erosion

- An important property of these planar sections of polycrystalline samples is that every boundary pixel connects either two or three grains (no more, no less). The former comes from a boundary between two grains, and the latter is a triple junction. At boundary locations with high curvature; i.e., 'corners', there is evidence that a triple junction may exist. If a corner is detected, then the algorithm checks for likely boundaries in the direction opposite the bisecting ray of its angle. Here we use the property that the angles comprising a triple junction cannot exceed 180^o. Furthermore, a potential interpolated boundary radiating from a corner must connect to either an endpoint (which we describe below) or another corner, and is not allowed to exceed a pre-determined length.

- The incomplete description of the grain boundaries also leaves some boundaries only partially delineated, terminating at what we call 'endpoints'. These endpoints are also likely candidates for interpolation. If two endpoints are sufficiently close, and if the angles between these endpoints and a connecting line segment is within a given bound, the line segment is added to the skeleton.

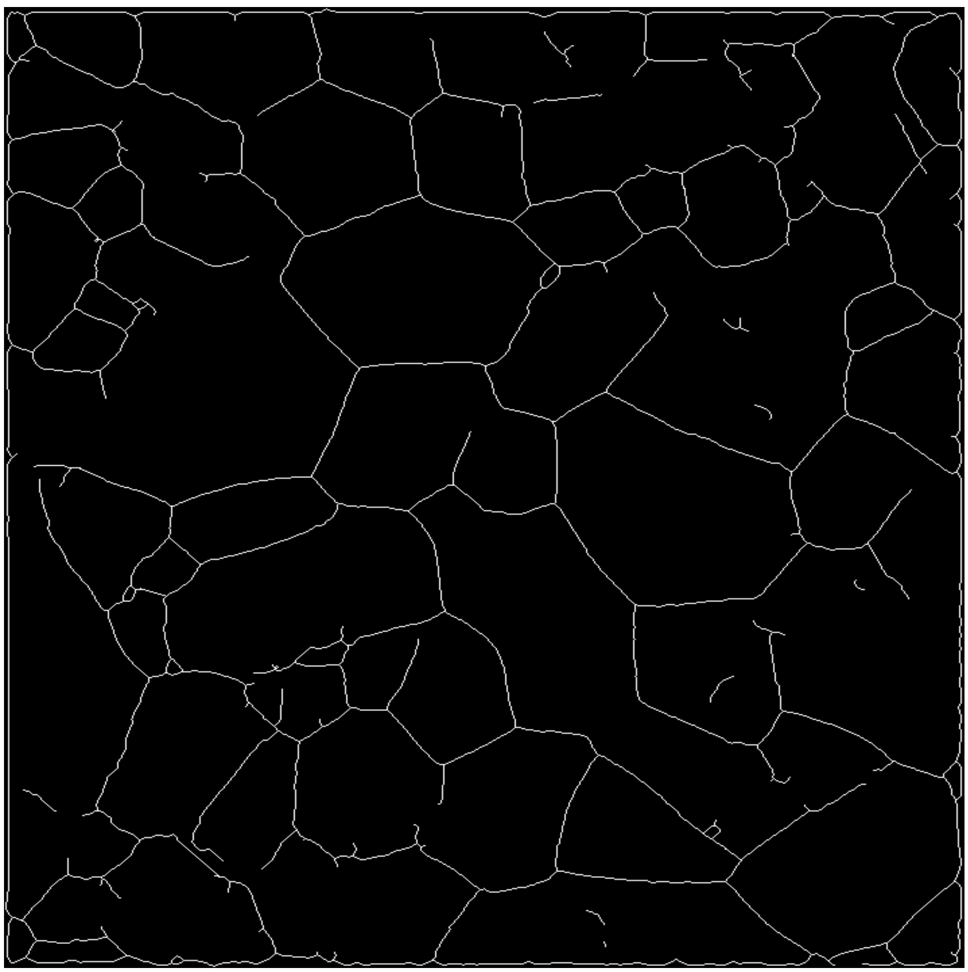

FIGURE 6. result after interpolation

(6) Finally, the software has the capability to allow the user to now edit the skeleton obtained automatically through steps (2) $\rightarrow$(5). The user is able to both delete

erroneous boundaries as well as add missing ones. Triple junctions, corners, and endpoints are tracked by the software, so when removing erroneous information, the user simply has to click anywhere on the segment to be removed, and only that precise segment is deleted from the skeleton. The final user-edited skeleton can be seen in figure 7.

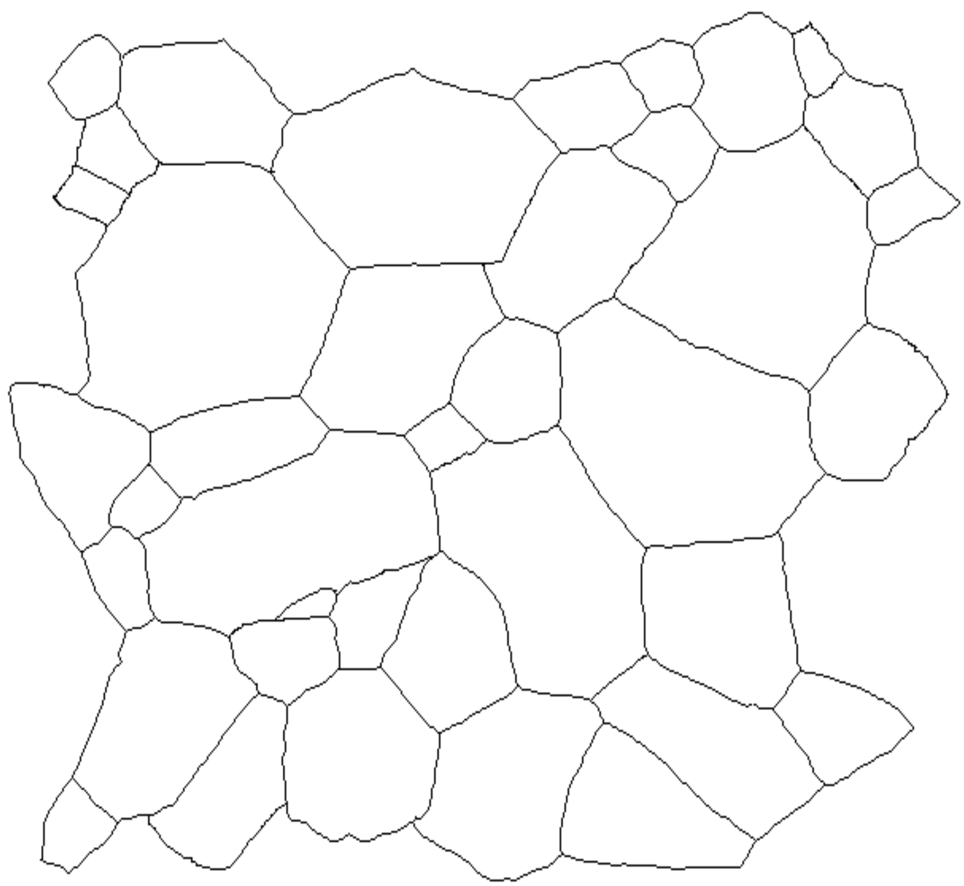

FIGURE 7. Final edited skeleton

A summary of the algorithm can be found in the following flow chart:

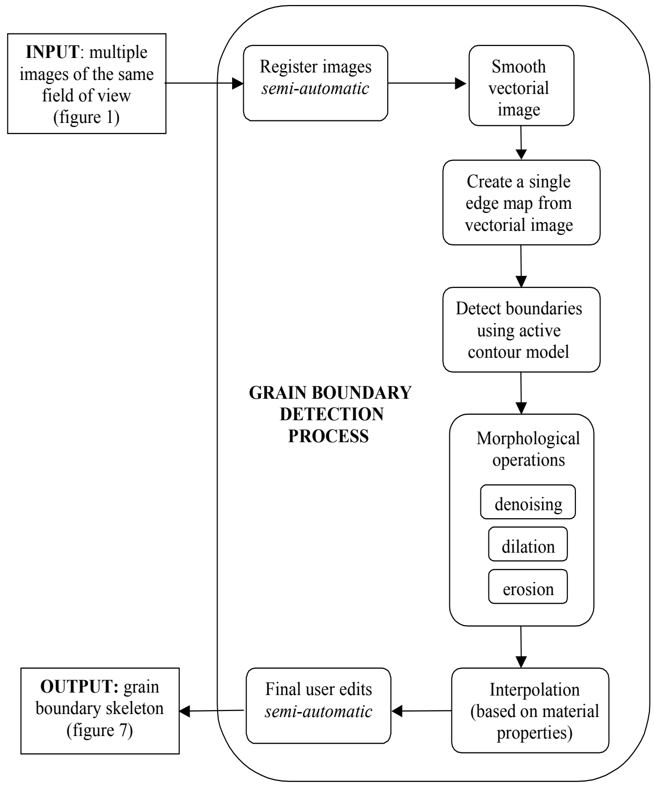

FIGURE 8. Summary of the grain boundary detection process